SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 1–18, February 1996
© 1996 Society for Industrial and Applied Mathematics

A SUPERLINEAR INFEASIBLE-INTERIOR-POINT AFFINE SCALING ALGORITHM FOR LCP*

R. D. C. MONTEIRO† AND S. J. WRIGHT‡

Abstract. We present an infeasible-interior-point algorithm for monotone linear complementarity problems in which the search directions are affine scaling directions and the step lengths are obtained from simple formulae that ensure both global and superlinear convergence. By choosing the value of a parameter in appropriate ways, polynomial complexity and convergence with Q-order up to (but not including) two can be achieved. The only assumption made to obtain the superlinear convergence is the existence of a solution satisfying strict complementarity.

Key words. infeasible-interior-point methods, monotone linear complementarity problems, superlinear convergence

AMS subject classifications. 90C33, 90C05, 65K05

1. Introduction.

The monotone linear complementarity problem (LCP) is to find a vector pair $(x, y) \in \mathsf{R}^n \times \mathsf{R}^n$ that satisfies the following conditions:

$$y = Mx + q, \tag{1.1a}$$

$$x \geq 0, \quad y \geq 0, \tag{1.1b}$$

$$x^T y = 0, \tag{1.1c}$$

where M is a positive semidefinite matrix. We use $\mathcal{S}$ to denote the solution set of (1.1) and $\mathcal{S}_c$ to denote the set of solutions that satisfy strict complementarity, that is,

$$\mathcal{S}_c = \{(x^*, y^*) \in \mathcal{S} \mid x^* + y^* > 0\}.$$

We say that (1.1) is *nondegenerate* if $\mathcal{S}_c \neq 0$.

It is well known that problems in linear programming and convex quadratic programming can be posed in the form (1.1). This fact partly accounts for the extensive effort that has gone into designing interior-point algorithms for this class of problems.

In this paper, we develop a superlinearly convergent infeasible-interior-point algorithm that uses only affine scaling directions. Several infeasible-interior-point algorithms have been developed to solve problem (1.1), or the linear programming (LP) subclass. For LP problems, these include works by Kojima, Megiddo, and Mizuno [1], Mizuno, Kojima, and Todd [3], Mizuno [2], Potra [6], and Zhang and Zhang [11]. For LCP, we mention Wright [7, 8], while Zhang [10] considers a generalized form of the monotone LCP.

In a companion paper [4], we describe a method in which all iterates (x^k, y^k) are *strictly feasible* (that is, $(x^k, y^k) > 0$ and $y^k = Mx^k + q$) and search directions are

* Received by the editors July 12, 1993; accepted for publication (in revised form) August 24, 1994.

† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (monteiro@isye.gatech.edu). The research of this author was supported by National Science Foundation grant DDM-9109404 and Office of Naval Research grant N00014-93-1-0234. This research was done while the author was a faculty member of the Systems and Industrial Engineering Department at the University of Arizona.

‡ Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439 (wright@mcs.anl.gov). The research of this author was supported by the Office of Scientific Computing, U.S. Department of Energy contract W-31-109-Eng-38.

generated by solving the linear system

$$\begin{bmatrix} M & -I \\ Y^k & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = \begin{bmatrix} 0 \\ -X^k Y^k e \end{bmatrix}, \tag{1.2}$$

where $X^k = \operatorname{diag}(x_1^k, x_2^k, \ldots, x_n^k)$, $Y^k = \operatorname{diag}(y_1^k, y_2^k, \ldots, y_n^k)$, and $e = (1, 1, \ldots, 1)^T$. The direction $(\Delta x^k, \Delta y^k)$ is known as the primal-dual affine scaling direction, and one can easily see that it is simply the Newton direction for the nonlinear mapping $F(x, y) = (Mx + q - y, XYe) = (0, 0)$ at the point (x^k, y^k). In [4], we choose a parameter $\delta \in (0, 1/2)$ and define the scalars ϕ_k and χ_k by

$$\phi_k \triangleq \frac{\Delta x^{k^T} \Delta y^k}{x^{k^T} y^k}, \qquad \chi_k \triangleq \max\left(0, \max_{\substack{i=1,2,\ldots,n \\ \Delta x_i^k \Delta y_i^k < 0}} \left(\frac{|\Delta x_i^k \Delta y_i^k|}{x_i^k y_i^k}\right)\right) \tag{1.3}$$

and the step length α_k by

$$\alpha_k \triangleq \frac{\delta}{\delta + \chi_k + (1+\delta)\phi_k}. \tag{1.4}$$

The step from one iterate to the next is defined as

$$(x^{k+1}, y^{k+1}) = (x^k, y^k) + \alpha_k(\Delta x^k, \Delta y^k).$$

The resulting simple algorithm is globally and superlinearly convergent. It even exhibits polynomial complexity for certain choices of the parameter δ and starting point (x^0, y^0).

In this paper, we generalize this algorithm by allowing the iterates to be *infeasible*. Given an initial point $(x^0, y^0) > 0$, the modified algorithm generates iterates (x^k, y^k) in such a way that $x^{k^T} y^k \downarrow 0$ and $\|y^k - Mx^k - q\| \to 0$ as $k \to \infty$. Moreover, the convergence of $\{x^{k^T} y^k\}$ to zero is superlinear. Much of the elegance of the analysis for the feasible algorithm [4] does not carry through to this infeasible case and the results are slightly weaker here, so we feel that a separate report is needed.

Our analysis makes use of the fact that all iterates generated by our algorithm lie in a set of the form

$$\mathcal{N}(\delta, \eta, \rho) = \{(x, y) > 0 \,|\, \rho x^T y \geq \|r\|^{1+\delta},\ x_i y_i \geq \eta (x^T y)^{1+\delta},\ i = 1, 2, \ldots, n\},$$

where $\delta > 0$, $\eta > 0$, and $\rho \geq 0$, and $r = y - Mx - q$. Note that $\mathcal{N}(\delta, \eta, 0)$ corresponds to the neighborhood $\mathcal{N}(\delta, \eta)$ that was used in [4].

We formally specify our algorithm as follows.

ALGORITHM IAS
initially: Choose $\delta \in (0, 1]$ and $(x^0, y^0) > 0$;
for $k = 0, 1, 2, \ldots$
Let $(\Delta x^k, \Delta y^k)$ denote the solution of the linear system

$$M\Delta x - \Delta y = r, \tag{1.5a}$$

$$Y\Delta x + X\Delta y = -YXe, \tag{1.5b}$$

where $r = r^k = y^k - Mx^k - q$, $X = X^k$, and $Y = Y^k$;
Calculate ϕ_k and χ_k as in (1.3) and set

$$\alpha_k = \frac{\delta}{\delta + \chi_k + (1+\delta)\phi_k^+}, \tag{1.6}$$

where $\phi_k^+ = \max(0, \phi_k)$;
Set $(x^{k+1}, y^{k+1}) = (x^k + \alpha_k \Delta x^k, y^k + \alpha_k \Delta y^k)$;
end for

Note that in the case of (x^k, y^k) feasible, we have

$$\Delta x^{k^T} \Delta y^k = \Delta x^{k^T} M \Delta x^k \geq 0.$$

Therefore $\phi_k \geq 0$ for all k, so the definition of α_k in (1.6) reduces to (1.4). Hence, the algorithm in [4] is a special case of Algorithm IAS.

In the remainder of the paper, we prove that Algorithm IAS produces a sequence of iterates $\{(x^k, y^k)\}$ such that the complementarity gap $x^{k^T} y^k$ and the residual $\|r^k\|$ converge to zero. Moreover, we show that this convergence is Q-superlinear provided that $\mathcal{S}_c \neq \emptyset$ and $\delta < 1/7$. An interesting feature of our superlinear convergence result is that we do not assume any condition that would imply the boundedness of the sequence $\{(x^k, y^k)\}$ such as (i) existence of a strictly feasible point or (ii) convergence of $\{x^{k^T} y^k\}$ and $\{\|r^k\|\}$ to 0 at a similar rate. Most superlinear convergence results for infeasible-interior-point algorithms require the first condition. Exceptions are the results of Potra [6] and Wright [8], which require only the second condition; specifically, for some $\gamma > 0$ we have $x^{k^T} y^k \geq \gamma \|r^k\|$ for all $k \geq 0$. In this paper, we show first that $\{x^{k^T} y^k\}$ converges superlinearly to zero then, as a consequence, that the sequence of iterates $\{(x^k, y^k)\}$ converges and hence is bounded. Also, for certain values of δ and for initial points (x^0, y^0) satisfying $(x^0, y^0) \geq (x^*, y^*)$, for some $(x^*, y^*) \in \mathcal{S}$, we show that the number of iterations required to reduce the gap $x^{k^T} y^k$ below a certain $\epsilon > 0$ is polynomially bounded.

The following notational conventions are used in the remainder of the paper. When $\mathcal{S}_c \neq \emptyset$, we can partition the index set $\{1, 2, \ldots, n\}$ into subsets B and N such that

$$i \in B \;\Leftrightarrow\; x_i^* > 0 \quad \forall (x^*, y^*) \in \mathcal{S}_c, \qquad i \in N \;\Leftrightarrow\; y_i^* > 0 \quad \forall (x^*, y^*) \in \mathcal{S}_c.$$

Unless otherwise specified, $\|.\|$ denotes the the Euclidean norm. For a general vector $z \in \mathsf{R}^n$ and index set $L \subseteq \{1, 2, \ldots, n\}$, z_L denotes the vector made up of components z_i for $i \in L$. If $M \in \mathsf{R}^{n \times n}$ and $I, L \subseteq \{1, 2, \ldots, n\}$, then M_{IL} refers to the submatrix of M consisting of the elements M_{ij} for $i \in I$ and $j \in L$. If $D \in \mathsf{R}^{n \times n}$ is diagonal, then D_B denotes the diagonal matrix constructed from D_{ii} for $i \in B$.

2. Global convergence. In this section, we prove the global convergence of Algorithm IAS. We show that for certain values of δ, Algorithm IAS has a polynomial bound on the number of iterations. We also obtain certain bounds on the directions generated by the algorithm, which will be used in the next section to prove superlinear convergence.

For much of the analysis, it is convenient to drop the iteration index k that appears on quantities such as (x^k, y^k), ϕ_k, and α_k. Accordingly, we define some index-free notation. For an arbitrary positive vector pair $(x, y) > 0$, let $(\Delta x, \Delta y)$ be obtained from (1.5), and define

$$(x(\alpha), y(\alpha)) \triangleq (x, y) + \alpha(\Delta x, \Delta y),$$

$$
\begin{aligned}
r(\alpha) &\triangleq y(\alpha) - Mx(\alpha) - q, \\
\phi &\triangleq \frac{\Delta x^T \Delta y}{x^T y}, \\
\chi &\triangleq \max\left(0, \max_{\substack{i=1,2,\ldots,n \\ \Delta x_i \Delta y_i < 0}} \left(\frac{|\Delta x_i \Delta y_i|}{x_i y_i}\right)\right),
\end{aligned} \tag{2.1}
$$

where $\alpha \in [0,1]$.

We start by stating and proving some basic results for quantities in (2.1).

LEMMA 2.1. $\phi \geq -\chi$.

Proof. We have

$$
\phi = \frac{\Delta x^T \Delta y}{x^T y} \geq \sum_{\substack{i=1,2,\ldots,n \\ \Delta x_i \Delta y_i < 0}} \frac{\Delta x_i \Delta y_i}{x^T y} \geq -\frac{\chi}{x^T y} \sum_{\substack{i=1,2,\ldots,n \\ \Delta x_i \Delta y_i < 0}} x_i y_i \geq -\chi. \qquad \square
$$

LEMMA 2.2. *Let $(x,y) > 0$ be given, and let $(\Delta x, \Delta y)$ be the direction determined by (1.5). Then, for all $\alpha \in [0,1]$,*

(a) $x(\alpha)^T y(\alpha) = x^T y(1 - \alpha + \alpha^2 \phi)$;

(b) $x_i(\alpha) y_i(\alpha) \geq x_i y_i (1 - \alpha - \alpha^2 \chi)$;

(c) $r(\alpha) = (1-\alpha) r$;

(d) $\Delta x_i \Delta y_i = 0$ *for all* $i = 1, 2, \ldots, n$ *if and only if* $(x(1), y(1))$ *is a solution of* (1.1) *that satisfies strict complementarity.*

Proof. Statements (a), (b), and (c) follow immediately from (1.5) and (2.1). The proof of (d) is essentially given by Monteiro and Wright [4, Lemma 3.1(d)], if we note in addition that $r(1) = 0$. $\square$

We now show that $\{x^{k^T} y^k\}$ is a decreasing sequence and that the property of membership of $\mathcal{N}(\delta, \eta, \rho)$ is inherited by successive iterates of the algorithm. In the proof of this result, we make use of the inequality

$$
(1-u)^{1+\delta} \leq 1 - (1+\delta)u + \delta u^2 \quad \forall u \leq 1, \quad \forall \delta \in (0,1], \tag{2.2}
$$

which is proved by Monteiro and Wright [4, Lemma 3.2].

LEMMA 2.3. *Suppose that $(x,y) \in \mathcal{N}(\delta, \eta, \rho)$ for some $\delta \in (0,1]$, $\eta > 0$, and $\rho \geq 0$ and that $(\Delta x, \Delta y)$ is determined by (1.5). Assume that $\Delta x_i \Delta y_i \neq 0$ for some $i \in \{1, 2, \ldots, n\}$. Then, for all*

$$
\alpha \in J \triangleq \left[0, \frac{\delta}{\delta + \chi + (1+\delta)\phi^+}\right], \tag{2.3}
$$

where $\phi^+ \equiv \max(0, \phi)$, we have

(a) $x(\alpha)^T y(\alpha) \leq [1 - \alpha/(1+\delta)] x^T y$; *and*

(b) $(x(\alpha), y(\alpha)) \in \mathcal{N}(\delta, \eta, \rho)$.

Proof. To show statement (a), note that for $\alpha \in J$, we have

$$
\alpha \phi^+ \leq \frac{\delta \phi^+}{\delta + \chi + (1+\delta)\phi^+} \leq \frac{\delta}{1+\delta}. \tag{2.4}
$$

Thus in view of Lemma 2.2(a), we obtain

$$
x(\alpha)^T y(\alpha) \leq x^T y (1 - \alpha(1 - \alpha\phi^+)) \leq x^T y \left(1 - \frac{\alpha}{1+\delta}\right).
$$

For statement (b), we start by showing that $x(\alpha)^T y(\alpha) > 0$ for all $\alpha \in J$. Observe first that the condition that $\Delta x_i \Delta y_i \neq 0$ for some $i \in \{1, 2, \ldots, n\}$ implies that either $\chi > 0$ or $\phi > 0$. Hence, $\chi + (1+\delta)\phi^+ > 0$, and therefore

$$\alpha \leq \frac{\delta}{\delta + \chi + (1+\delta)\phi^+} \leq \frac{1}{1 + \chi + (1+\delta)\phi^+} < 1, \qquad \alpha \leq \frac{1}{1+\chi}, \tag{2.5}$$

for all $\alpha \in J$. Observe that if $\phi \geq 0$, then, in view of Lemma 2.2(a) and (2.5), we have $x(\alpha)^T y(\alpha) \geq (1-\alpha) x^T y > 0$ for all $\alpha \in J$. Assume that $\phi < 0$. In this case, we must have $\chi > 0$. This together with (2.5), Lemma 2.1, and Lemma 2.2(a) then imply

$$\begin{aligned} x(\alpha)^T y(\alpha) &= x^T y(1 - \alpha + \alpha^2 \phi) \\ &\geq x^T y(1 - \alpha - \alpha^2 \chi) \\ &> x^T y\,(1 - \alpha(1+\chi)) \geq 0 \end{aligned}$$

for all $\alpha \in J$ with $\alpha > 0$. We have thus shown that $x(\alpha)^T y(\alpha) > 0$ for all $\alpha \in J$. Now, using Lemma 2.2(a), (2.2), and (2.4), we obtain for all $\alpha \in J$ that

$$\begin{aligned} 0 &< \eta[x(\alpha)^T y(\alpha)]^{1+\delta} \\ &\leq \eta(x^T y)^{1+\delta}[1 - \alpha(1 - \alpha\phi^+)]^{1+\delta} \\ &\leq \eta(x^T y)^{1+\delta}[1 - (1+\delta)\alpha(1 - \alpha\phi^+) + \delta\alpha^2(1 - \alpha\phi^+)^2] \\ &\leq \eta(x^T y)^{1+\delta}[1 - (1+\delta)\alpha(1 - \alpha\phi^+) + \delta\alpha^2] \\ &\leq \eta(x^T y)^{1+\delta}[1 - \alpha - \alpha^2\chi], \end{aligned} \tag{2.6}$$

where in the last inequality we used the fact that the interval J is exactly the set of all $\alpha \geq 0$ for which

$$1 - \alpha(1+\delta)(1 - \alpha\phi^+) + \delta\alpha^2 \leq 1 - \alpha - \alpha^2\chi.$$

From (2.6), Lemma 2.2(b), and the fact that $(x, y) \in \mathcal{N}(\delta, \eta, \rho)$, we have

$$\begin{aligned} x_i(\alpha) y_i(\alpha) &\geq x_i y_i (1 - \alpha - \alpha^2 \chi) \\ &\geq \eta(x^T y)^{1+\delta}(1 - \alpha - \alpha^2\chi) \\ &\geq \eta[x(\alpha)^T y(\alpha)]^{1+\delta} > 0. \end{aligned}$$

Thus, the second condition for membership of $\mathcal{N}(\delta, \eta, \rho)$ is satisfied by $(x(\alpha), y(\alpha))$. To show the first condition, we observe that

$$\rho x^T y \geq \|r\|^{1+\delta}. \tag{2.7}$$

From Lemma 2.2(c), we know that $\|r(\alpha)\| = (1-\alpha)\|r\|$. By (2.2) and (2.7), it follows that

$$\|r(\alpha)\|^{1+\delta} = (1-\alpha)^{1+\delta}\|r\|^{1+\delta} \leq [1 - (1+\delta)\alpha + \delta\alpha^2]\rho x^T y. \tag{2.8}$$

For $\alpha \geq 0$, the inequality

$$1 - (1+\delta)\alpha + \delta\alpha^2 \leq 1 - \alpha + \alpha^2\phi \tag{2.9}$$

is equivalent to $\delta \geq \alpha(\delta - \phi)$, which is satisfied by every $\alpha \in [0, \delta/(\delta + \phi^-)]$, where $\phi^- \equiv -\min(0, \phi)$. Now, using Lemma 2.1, we can easily see that $J \subseteq [0, \delta/(\delta + \phi^-)]$.

We then conclude that (2.9) is satisfied by every $\alpha \in J$. Hence, in view of (2.8) and Lemma 2.2(a), we have

$$\|r(\alpha)\|^{1+\delta} \leq (1-\alpha+\alpha^2\phi)\rho x^T y = \rho x(\alpha)^T y(\alpha).$$

Therefore, the first condition for membership of $(x(\alpha), y(\alpha))$ in $\mathcal{N}(\delta,\eta,\rho)$ is also satisfied, and we have the result. □

From now on, given the initial point (x^0, y^0) and $\delta \in (0,1]$, we let

$$\pi_0 = \frac{\min_i(x_i^0 y_i^0)}{x^{0T}y^0/n}, \qquad \rho = \frac{\|r^0\|^{1+\delta}}{x^{0T}y^0}, \qquad \eta = \frac{\min_i(x_i^0 y_i^0)}{(x^{0T}y^0)^{1+\delta}} = \frac{\pi_0}{n(x^{0T}y^0)^{\delta}}. \tag{2.10}$$

Note that π_0 (a measure of the centrality of the initial point) is at most 1 and $(x^0, y^0) \in \mathcal{N}(\delta,\eta,\rho)$.

The following result is an immediate consequence of Lemma 2.2(c) and Lemma 2.3.

COROLLARY 2.4. *The sequence* $\{(x^k, y^k)\}$ *generated by Algorithm* IAS *satisfies*

(a) $x^{k+1^T} y^{k+1} \leq [1-\alpha_k/(1+\delta)] x^{k^T} y^k$ *for all* $k \geq 0$;

(b) $r^k = \nu_k r^0$ *for all* $k \geq 0$, *where* $\nu_k \equiv \Pi_{l=0}^{k-1}(1-\alpha_l)$;

(c) $(x^k, y^k) \in \mathcal{N}(\delta,\eta,\rho)$ *for all* $k \geq 0$, *where* η *and* ρ *are as defined in* (2.10).

Our main goal now is to derive estimates for the quantities χ and ϕ. These estimates will then be used to establish global convergence of Algorithm IAS, as well as polynomial complexity for certain values of δ. The following inequality is exploited in a number of proofs that follow.

LEMMA 2.5. *Let* (x^0, y^0) *and* (x, y) *be points such that* $y - Mx - q = \nu(y^0 - Mx^0 - q)$ *for some* $\nu \in \mathsf{R}$, *and let* $(\bar{x}, \bar{y})$ *be a point such that* $\bar{y} - M\bar{x} - q = 0$. *Then,*

$$\begin{aligned} 0 \leq \nu^2 x^{0T} y^0 + (1-\nu)^2 \bar{x}^T \bar{y} + x^T y + \nu(1-\nu)(x^{0T}\bar{y} + \bar{x}^T y^0) \\ -\nu(x^{0T} y + x^T y^0) - (1-\nu)(\bar{x}^T y + x^T \bar{y}). \end{aligned} \tag{2.11}$$

Proof. See, for example, Wright [8, Lemma 3.2]. □

As an immediate consequence of this inequality, we obtain the following result.

LEMMA 2.6. *Let* $(x^0, y^0) > 0$ *be given. Then there exists a constant* $C_0 \geq 0$ *satisfying the following property: If* $(x, y) \geq 0$ *is a point such that* $y - Mx - q = \nu(y^0 - Mx^0 - q)$ *for some* $\nu \in [0,1]$, *then*

$$\max\{\nu\|x\|_\infty, \nu\|y\|_\infty\} \leq C_0 \left[x^T y + \nu\right]. \tag{2.12}$$

Proof. Assume that $(\bar{x}, \bar{y})$ is a solution of (1.1). Using inequality (2.11) and the facts that $\nu \in [0,1]$, $(x^0, y^0) > 0$, $(x, y) \geq 0$, $(\bar{x}, \bar{y}) \geq 0$, and $\bar{x}^T \bar{y} = 0$, we obtain

$$\begin{aligned} x^{0T}(\nu y) + y^{0T}(\nu x) &\leq \nu x^{0T} y^0 + x^T y + \nu(x^{0T}\bar{y} + y^{0T}\bar{x}) \\ &\leq \max\left\{1,\, x^{0T} y^0 + x^{0T}\bar{y} + y^{0T}\bar{x}\right\}(x^T y + \nu). \end{aligned}$$

Relation (2.12) now follows by letting

$$C_0 \triangleq \frac{\max\left\{1, x^{0T} y^0 + x^{0T}\bar{y} + y^{0T}\bar{x}\right\}}{\min_{i=1,2,\ldots,n}\{\min(x_i^0, y_i^0)\}}. \qquad \square$$

LEMMA 2.7. *Let the vectors $u^0 \in \mathsf{R}^n$ and $v^0 \in \mathsf{R}^n$ and the parameters $\eta > 0$ and $\delta \geq 0$ be given. Then, for every positive vector $(x, y) \in \mathsf{R}^{2n}$ and scalar $\nu \geq 0$ such that*

$$y - Mx - q = \nu(v^0 - Mu^0), \tag{2.13}$$

$$\min_{i=1,2,\ldots,n} x_i y_i \geq \eta(x^T y)^{1+\delta}, \tag{2.14}$$

the corresponding direction $(\Delta x, \Delta y)$ determined by (1.5) satisfies

$$\max\left(\|D^{-1}\Delta x\|, \|D\Delta y\|\right) \leq (x^T y)^{1/2} + \frac{2\nu}{\eta^{1/2}(x^T y)^{(1+\delta)/2}}\left[\|Yu^0\| + \|Xv^0\|\right], \tag{2.15}$$

where $D \equiv X^{1/2}Y^{-1/2}$.

Proof. We will first show that

$$\max\left(\|D^{-1}\Delta x\|, \|D\Delta y\|\right) \leq (x^T y)^{1/2} + 2\nu\left[\|D^{-1}u^0\| + \|Dv^0\|\right]. \tag{2.16}$$

Indeed, it follows from (1.5a) and (2.13) that

$$M\Delta x - \Delta y = r = y - Mx - q = \nu(v^0 - Mu^0).$$

Therefore,

$$M(\Delta x + \nu u^0) = \Delta y + \nu v^0,$$

which by monotonicity of (1.1) implies that

$$(\Delta x + \nu u^0)^T(\Delta y + \nu v^0) \geq 0.$$

Hence, we obtain

$$\Delta x^T \Delta y \geq -\nu u^{0T}\Delta y - \nu v^{0T}\Delta x - \nu^2 u^{0T} v^0. \tag{2.17}$$

If we multiply (1.5b) by $(XY)^{-1/2}$ and square both sides, we obtain

$$\|D^{-1}\Delta x\|^2 + \|D\Delta y\|^2 + 2\Delta x^T \Delta y = x^T y,$$

which, in view of (2.17), implies

$$\|D^{-1}\Delta x - \nu D v^0\|^2 + \|D\Delta y - \nu D^{-1}u^0\|^2 - \nu^2\|Dv^0 + D^{-1}u^0\|^2 \leq x^T y.$$

Therefore, using the inequality $(\beta^2 + \gamma^2)^{1/2} \leq |\beta| + |\gamma|$ and the triangle inequality, we have

$$\begin{aligned}
\|D^{-1}\Delta x\| &\leq \left[x^T y + \nu^2\|Dv^0 + D^{-1}u^0\|^2\right]^{1/2} + \nu\|Dv^0\| \\
&\leq (x^T y)^{1/2} + 2\nu\|Dv^0\| + \nu\|D^{-1}u^0\| \\
&\leq (x^T y)^{1/2} + 2\nu\left[\|Dv^0\| + \|D^{-1}u^0\|\right]. \qquad (2.18)
\end{aligned}$$

A similar bound can be derived for $\|D\Delta y\|$, and hence relation (2.16) follows.

We now derive (2.15) from (2.16) and (2.14). From (2.14) we obtain

$$\|D^{-1}u^0\|^2 = \sum_{i=1}^n \frac{y_i(u_i^0)^2}{x_i} = \sum_{i=1}^n \frac{(y_i u_i^0)^2}{x_i y_i} \leq \frac{\|Yu^0\|^2}{\eta(x^T y)^{1+\delta}}. \tag{2.19}$$

Similarly, we also have

$$\|Dv^0\|^2 \leq \frac{\|Xv^0\|^2}{\eta(x^Ty)^{1+\delta}}. \tag{2.20}$$

Relation (2.15) now follows by substituting (2.19) and (2.20) into (2.16). □

We are now ready to show global convergence of Algorithm IAS without imposing any condition on the initial point (x^0, y^0).

THEOREM 2.8. *For any initial point $(x^0, y^0) > 0$, the sequence $\{(x^k, y^k)\}$ generated by Algorithm* IAS *satisfies*

(a) $x^{k^T} y^k \to 0$ *and*

(b) $r^k \to 0$.

Proof. First note that (b) follows as a consequence of (a) from Corollary 2.4(c). To show (a), we assume for contradiction that $x^{k^T} y^k$ does not converge to 0. Thus, in view of Corollary 2.4(a), it follows that there exists $\epsilon > 0$ such that

$$x^{k^T} y^k \geq \epsilon \quad \forall k \geq 0. \tag{2.21}$$

Hence, in view of Corollary 2.4(c), we have

$$\min_{i=1,2,\ldots,n} \{x_i^k y_i^k\} \geq \eta(x^{k^T} y^k)^{1+\delta} \geq \eta\epsilon^{1+\delta} \quad \forall k \geq 0. \tag{2.22}$$

Since the point (x^k, y^k) satisfies the assumptions of Lemma 2.6 from Corollary 2.4(b), we obtain

$$\max\{\nu^k\|x^k\|_\infty, \nu^k\|y^k\|_\infty\} \leq C_0\left(x^{k^T} y^k + \nu^k\right) \leq C_0\left(x^{0^T} y^0 + 1\right) \quad \forall k \geq 0. \tag{2.23}$$

Let $(u^0, v^0) \in \mathsf{R}^{2n}$ be such that $v^0 - Mu^0 = r^0$ (e.g., $u^0 = 0$ and $v^0 = r^0$). From Corollary 2.4, (x^k, y^k) and ν_k satisfy the assumptions of Lemma 2.7 with η given by (2.10). Hence, inequality (2.15) gives

$$\begin{aligned}
&\max\left(\|(D^k)^{-1}\Delta x^k\|, \|D^k\Delta y^k\|\right) \\
&\leq \left(x^{k^T} y^k\right)^{1/2} + \frac{2\nu^k}{\eta^{1/2}(x^{k^T} y^k)^{(1+\delta)/2}}\left[\|Y^k u^0\| + \|X^k v^0\|\right] \\
&\leq \left(x^{k^T} y^k\right)^{1/2} + \frac{2}{\eta^{1/2}\epsilon^{(1+\delta)/2}}\max\left(\|\nu^k y^k\|_\infty, \|\nu^k x^k\|_\infty\right)\left(\|u^0\| + \|v^0\|\right) \\
&\leq \left(x^{0^T} y^0\right)^{1/2} + \frac{2C_0\left(x^{0^T} y^0 + 1\right)}{\eta^{1/2}\epsilon^{(1+\delta)/2}}\left(\|u^0\| + \|v^0\|\right) \triangleq \hat{C}_0. \qquad (2.24)
\end{aligned}$$

Using (2.22) and (2.24), we obtain for all $k \geq 0$ that

$$\begin{aligned}
\max\{\phi_k^+, \chi_k\} &\leq \max_i\left\{\frac{|\Delta x_i^k \Delta y_i^k|}{x_i^k y_i^k}\right\} \\
&\leq \frac{\|\mathrm{diag}(\Delta x^k)\Delta y^k\|}{\min_i\{x_i^k y_i^k\}} \\
&\leq \frac{\|(D^k)^{-1}\Delta x^k\|\|D^k\Delta y^k\|}{\eta\epsilon^{1+\delta}} \\
&\leq \frac{\hat{C}_0^{\,2}}{\eta\epsilon^{1+\delta}},
\end{aligned}$$

where the first inequality follows by using a similar argument as in the proof of Lemma 2.1. Hence, from (1.6) we have $\inf_k \alpha_k > 0$, which implies ${x^k}^T y^k \to 0$ from Corollary 2.4(a). We have thus obtained a contradiction, and statement (a) follows. □

We note that Theorem 2.8 does not imply that the sequence $\{(x^k, y^k)\}$ is convergent or bounded. In §3, we will show that $\{(x^k, y^k)\}$ is convergent under the assumptions that there is a solution of (1.1) satisfying strict complementarity and that $\delta < 1/7$.

Our next aim is to derive an upper bound on the number of iterations required by Algorithm IAS to reduce the duality gap below a given tolerance $\epsilon > 0$. We will see that for certain values of δ, Algorithm IAS has a polynomial bound on this number of iterations. We restrict our analysis to certain choices of the initial point (x^0, y^0).

Given some conditions on the initial point (x^0, y^0), the next result gives bounds on the left-hand side of (2.15) in terms of the initial gap ${x^0}^T y^0$, the current gap $x^T y$, and the measure of initial centrality π_0.

LEMMA 2.9. *Suppose that the initial point $(x^0, y^0) > 0$ is chosen so that $(x^0, y^0) \geq (x^*, y^*)$ for some $(x^*, y^*) \in \mathcal{S}$ and that π_0, η, and ρ are defined by (2.10). Let (x, y) be any vector pair in $\mathcal{N}(\delta, \eta, \rho)$ with $x^T y \leq {x^0}^T y^0$ and $y - Mx - q = \nu(y^0 - Mx^0 - q)$ for some $\nu \in (0, 1]$. Let $(\Delta x, \Delta y)$ be the direction determined by (1.5). Then, we have*

$$\max\left(\|D^{-1}\Delta x\|, \|D\Delta y\|\right) \leq 9\left(\frac{n}{\pi_0}\right)^{1/2}\left(\frac{{x^0}^T y^0}{x^T y}\right)^{3\delta/2}(x^T y)^{1/2}. \tag{2.25}$$

Proof. We start by defining a vector pair (u^0, v^0) by

$$(u^0, v^0) = (x^0, y^0) - (x^*, y^*) \geq 0, \tag{2.26}$$

for $(x^*, y^*) \in \mathcal{S}$. It is easy to check that

$$\nu(v^0 - Mu^0) = \nu(y^0 - Mx^0 - q) = y - Mx - q.$$

Therefore Lemma 2.7 applies and, as a consequence, the inequality (2.15) holds. Now, using (2.26), we obtain

$$\|Yu^0\| + \|Xv^0\| \leq y^T u^0 + x^T v^0 = y^T(x^0 - x^*) + x^T(y^0 - y^*) \leq y^T x^0 + x^T y^0. \tag{2.27}$$

By substituting this last relation into (2.15), we obtain

$$\max\left(\|D^{-1}\Delta x\|, \|D\Delta y\|\right) \leq (x^T y)^{1/2} + 2\nu\eta^{-1/2}(x^T y)^{-(1+\delta)/2}[y^T x^0 + x^T y^0]. \tag{2.28}$$

We now invoke the inequality (2.11) with $(\bar{x}, \bar{y}) = (x^*, y^*)$, noting that $\nu \in (0, 1]$, $(x^*, y^*) \geq 0$, ${x^*}^T y^* = 0$, and $(x, y) > 0$, to deduce that

$$\nu({x^0}^T y + x^T y^0) \leq \nu^2 {x^0}^T y^0 + x^T y + \nu({x^0}^T y^* + {x^*}^T y^0). \tag{2.29}$$

Now, by (2.26), we have

$$\frac{{x^0}^T y^* + {x^*}^T y^0}{{x^0}^T y^0} = \frac{{x^0}^T(y^0 - v^0) + (x^0 - u^0)^T y^0}{{x^0}^T y^0} = 2 - \frac{{x^0}^T v^0 + {u^0}^T y^0}{{x^0}^T y^0} \leq 2. \tag{2.30}$$

In addition, by (2.10) and the definition of $\mathcal{N}(\delta, \eta, \rho)$, we have

$$\nu = \frac{\|r\|}{\|r^0\|} \leq \frac{[\rho x^T y]^{1/(1+\delta)}}{\|r^0\|} = \left(\frac{x^T y}{{x^0}^T y^0}\right)^{1/(1+\delta)}. \tag{2.31}$$

Combining (2.29), (2.30), and (2.31), we get

$$\begin{aligned}\nu(x^{0T}y + x^T y^0) &\le x^{0T}y^0\left[\nu^2 + \left(\frac{x^Ty}{x^{0T}y^0}\right) + 2\nu\right] \\ &\le x^{0T}y^0\left(\frac{x^Ty}{x^{0T}y^0}\right)^{1/(1+\delta)}\left[\nu + \left(\frac{x^Ty}{x^{0T}y^0}\right)^{\delta/(1+\delta)} + 2\right] \\ &\le 4x^{0T}y^0\left(\frac{x^Ty}{x^{0T}y^0}\right)^{1/(1+\delta)}. \qquad (2.32)\end{aligned}$$

If we substitute (2.32) into (2.28) and replace η by its definition (2.10), we obtain

$$\begin{aligned}\max\left(\|D^{-1}\Delta x\|, \|D\Delta y\|\right) &\le (x^Ty)^{1/2} + 8\eta^{-1/2}(x^Ty)^{-(1+\delta)/2}(x^{0T}y^0)^{\delta/(1+\delta)}(x^Ty)^{1/(1+\delta)} \\ &= (x^Ty)^{1/2}\left[1 + 8\left(\frac{n}{\pi_0}\right)^{1/2}\left(\frac{x^{0T}y^0}{x^Ty}\right)^{(\delta/2)(3+\delta)/(1+\delta)}\right] \\ &\le 9\left(\frac{n}{\pi_0}\right)^{1/2}\left(\frac{x^{0T}y^0}{x^Ty}\right)^{3\delta/2}(x^Ty)^{1/2},\end{aligned}$$

where the last inequality follows from

$$\frac{x^{0T}y^0}{x^Ty} \ge 1, \qquad \frac{3+\delta}{1+\delta} < 3, \qquad \pi_0 \le 1. \qquad \square$$

We next derive uniform bounds for the quantities $|\phi_k|$ and χ_k defined by (1.3).

LEMMA 2.10. *Suppose that the initial point $(x^0, y^0) > 0$ satisfies $(x^0, y^0) \ge (x^*, y^*)$ for some $(x^*, y^*) \in \mathcal{S}$ and that η and ρ are defined by* (2.10). *Then for all iterates (x^k, y^k) generated by Algorithm* IAS, *we have*

$$|\phi_k| \le 81\left(\frac{n}{\pi_0}\right)\left(\frac{x^{0T}y^0}{x^{kT}y^k}\right)^{3\delta}, \qquad (2.33a)$$

$$0 \le \chi_k \le 81\left(\frac{n}{\pi_0}\right)^2\left(\frac{x^{0T}y^0}{x^{kT}y^k}\right)^{4\delta}. \qquad (2.33b)$$

Proof. By Corollary 2.4, it follows that (x^k, y^k) satisfies the assumptions of Lemma 2.9 for all $k \ge 0$. Hence, it follows from (2.25) that

$$|\phi_k| = \frac{|\Delta x^{kT}\Delta y^k|}{x^{kT}y^k} \le \frac{\|D^k\Delta x^k\|\|(D^k)^{-1}\Delta y^k\|}{x^{kT}y^k} \le 81\left(\frac{n}{\pi_0}\right)\left(\frac{x^{0T}y^0}{x^{kT}y^k}\right)^{3\delta}.$$

Similarly, by (2.10), (2.25), and the definition of $\mathcal{N}(\delta, \eta, \rho)$, we have

$$\chi_k \le \max_i \frac{|\Delta x_i^k \Delta y_i^k|}{x_i^k y_i^k} \le \frac{\|D^k\Delta x^k\|\|(D^k)^{-1}\Delta y^k\|}{\eta(x^{kT}y^k)^{1+\delta}}$$

$$\leq 81\left(\frac{n}{\pi_0}\right)\left(\frac{x^{0T}y^0}{x^{kT}y^k}\right)^{3\delta}(x^{kT}y^k)^{-\delta}\eta^{-1}$$
$$= 81\left(\frac{n}{\pi_0}\right)^2\left(\frac{x^{0T}y^0}{x^{kT}y^k}\right)^{4\delta},$$

as required. □

We can now state a global convergence result.

THEOREM 2.11. *Suppose that the initial point* $(x^0, y^0) > 0$ *satisfies* $(x^0, y^0) \geq (x^*, y^*)$ *for some* $(x^*, y^*) \in \mathcal{S}$. *Let* $\delta \in (0, 1]$ *be given, and let* η *and* ρ *be defined by* (2.10). *Then, given* $\epsilon \in (0, x^{0T}y^0)$, *Algorithm* IAS *will generate an iterate* (x^k, y^k) *with* $x^{kT}y^k \leq \epsilon$ *in at most*

$$O\left(\frac{1}{\delta}\left(\frac{n}{\pi_0}\right)^2\left(\frac{x^{0T}y^0}{\epsilon}\right)^{4\delta}\log(x^{0T}y^0/\epsilon)\right) \tag{2.34}$$

iterations, and the sequence $\{x^{kT}y^k\}$ *will be strictly monotonically decreasing.*

Proof. To simplify notation, let $\tau = (x^{0T}y^0/\epsilon)$. If the iterate (x^k, y^k) satisfies $x^{kT}y^k \geq \epsilon$ then, in view of (2.33), we have

$$|\phi_k| \leq 81(n/\pi_0)\tau^{3\delta}, \qquad \chi_k \leq 81(n/\pi_0)^2\tau^{4\delta},$$

and hence

$$\begin{aligned}\alpha_k &= \frac{\delta}{\delta + \chi_k + (1+\delta)\phi_k^+} \\ &\geq \frac{\delta}{\delta + 81(n/\pi_0)^2\tau^{4\delta} + (1+\delta)81(n/\pi_0)\tau^{3\delta}} \\ &\geq \frac{\delta}{\delta + 243(n/\pi_0)^2\tau^{4\delta}} \\ &\geq \frac{\delta}{244}\left(\frac{\pi_0}{n}\right)^2\tau^{-4\delta} \\ &\triangleq \bar{\alpha}. \end{aligned} \tag{2.35}$$

Strictly monotonic decrease of the sequence $\{x^{kT}y^k\}$ now follows from Corollary 2.4(a) and the fact that the inclusion $(x^k, y^k) \in \mathcal{N}(\delta, \eta, \rho)$ implies that $x^{kT}y^k > 0$. Now let K be the smallest nonnegative integer for which $x^{K+1T}y^{K+1} \leq \epsilon$. Since (2.35) holds for every $0 \leq k \leq K$, we obtain from Lemma 2.3(a) that

$$x^{KT}y^K \leq [1 - \bar{\alpha}/(1+\delta)]^K x^{0T}y^0. \tag{2.36}$$

Assume for contradiction that

$$K \geq \frac{(1+\delta)\log\tau}{\bar{\alpha}}.$$

Using the inequality $\log(1-\beta) \leq -\beta$ for $\beta < 1$ and the previous relation, we can easily show that

$$[1 - \bar{\alpha}/(1+\delta)]^K \leq \tau^{-1},$$

which, in view of the definition of τ and relation (2.36), implies

$$x^{K^T} y^K \leq \tau^{-1} x^{0^T} y^0 = \epsilon.$$

This relation contradicts our definition of K. Therefore, using (2.35) and the definition of τ, we have

$$K < \frac{(1+\delta)\log\tau}{\bar{\alpha}} = O\left(\frac{1}{\delta}\left(\frac{n}{\pi_0}\right)^2 \left(\frac{x^{0^T} y^0}{\epsilon}\right)^{4\delta} \log(x^{0^T} y^0/\epsilon)\right). \qquad \square$$

Note in particular that if

$$\epsilon = e^{-O(L)}, \qquad x^{0^T} y^0 = e^{O(L)}, \qquad \pi_0 = O(1), \qquad \delta = 1/O(L),$$

then the bound in (2.34) is $O(n^2 L^2)$, that is, there is a polynomial bound on the number of iterations of Algorithm IAS.

3. Local convergence. Under the conditions that problem (1.1) is nondegenerate and $\delta \in (0, 1/7)$, we prove in this section that Algorithm IAS started from any initial point $(x^0, y^0) > 0$ generates a sequence of iterates $\{(x^k, y^k)\}$ such that $\{x^{k^T} y^k\}$ converges superlinearly to zero. Moreover, under the same conditions, we also show that the sequence $\{(x^k, y^k)\}$ converges to a solution of (1.1). We emphasize that the convergence of $\{(x^k, y^k)\}$ is obtained without assuming the existence of a strictly feasible point for (1.1). (This assumption was used by the local convergence analysis in Wright [7], for example.)

The following assumption is imposed throughout this section.

ASSUMPTION 1 (Nondegeneracy). $\mathcal{S}_c \neq \emptyset$.

We show in [5] that Assumption 1 is in fact necessary for superlinear convergence of Newton-based primal-dual algorithms.

An important feature of this section is that we are able to prove boundedness of the steps $(\Delta x^k, \Delta y^k)$ in terms of $x^{k^T} y^k$ without an intermediate result that the sequence (x^k, y^k) is bounded. The key result — that $\|\Delta x_B^k\|$ and $\|\Delta y_N^k\|$ are both $O(x^{k^T} y^k)$ — is obtained by combining Lemmas 3.3 and 3.4 below. In [7], the same result was obtained by combining Lemma 3.4 with a more elementary result in which boundedness of (x^k, y^k) was an essential ingredient. In fact, we can use the techniques of this section to drop the strict feasibility assumption from the analysis of [7] without affecting the conclusions of the paper.

After finding the bounds on the steps $(\Delta x^k, \Delta y^k)$ in terms of the complementarity gap $x^{k^T} y^k$, we bound the scalars $|\phi_k|$ and χ_k. Subsequently, we show that $\alpha_k \to 1$, from which Q-superlinear convergence follows. We also find a lower bound on the Q-order of convergence in terms of δ.

The following result yields bounds on the nonbasic components of x and y.

LEMMA 3.1. *Suppose that Assumption* 1 *holds. Assume that a point* $(x^0, y^0) > 0$ *and that scalars* $\delta \geq 0$, $\eta > 0$, *and* $\rho \geq 0$ *are given. Then, there exists a constant* $C_1 \geq 0$ *with the following property: For every vector* $(x, y) \in \mathcal{N}(\delta, \eta, \rho)$ *such that*

$$x^T y \leq x^{0^T} y^0, \tag{3.1}$$

$$y - Mx - q = \nu(y^0 - Mx^0 - q) \text{ for some } \nu \in [0, 1/2], \tag{3.2}$$

we have

$$x_i \le C_1 (x^T y)^{1/(1+\delta)} \quad \forall i \in N, \tag{3.3a}$$

$$y_i \le C_1 (x^T y)^{1/(1+\delta)} \quad \forall i \in B. \tag{3.3b}$$

Proof. Consider first the infeasible case in which $r^0 = y^0 - Mx^0 - q \neq 0$. Setting $(\bar{x}, \bar{y}) = (x^*, y^*)$ for some $(x^*, y^*) \in \mathcal{S}_c$ in inequality (2.11) and using the fact that $\nu \in [0, 1/2]$, $(x, y) > 0$, $(x^*, y^*) \ge 0$ and $x^{*T} y^* = 0$, we obtain

$$x^{*T} y + x^T y^* \le \frac{\nu}{1-\nu} x^{0T} y^0 + \frac{1}{1-\nu} x^T y + \nu (x^{0T} y^* + x^{*T} y^0). \tag{3.4}$$

Using (3.2) and the fact that $(x, y) \in \mathcal{N}(\delta, \eta, \rho)$, we get

$$\nu = \frac{\|r\|}{\|r^0\|} \le \frac{(\rho x^T y)^{1/(1+\delta)}}{\|r^0\|}, \tag{3.5}$$

where, as usual, $r = y - Mx - q$. From (3.1), (3.5), (3.4), and the fact that $1 - \nu \ge 1/2$, we get

$$\begin{aligned}
&x^{*T} y + x^T y^* \\
&\le \frac{2 x^{0T} y^0}{\|r^0\|} \rho^{1/(1+\delta)} (x^T y)^{1/(1+\delta)} + 2 x^T y + \frac{1}{\|r^0\|} \rho^{1/(1+\delta)} (x^{0T} y^* + x^{*T} y^0) (x^T y)^{1/(1+\delta)} \\
&\le (x^T y)^{1/(1+\delta)} \left[\frac{2 x^{0T} y^0}{\|r^0\|} \rho^{1/(1+\delta)} + 2 (x^{0T} y^0)^{\delta/(1+\delta)} + \frac{1}{\|r^0\|} \rho^{1/(1+\delta)} (x^{0T} y^* + x^{*T} y^0) \right] \\
&= \bar{C}_1 (x^T y)^{1/(1+\delta)},
\end{aligned}$$

where $\bar{C}_1$ is defined in an obvious way. This last relation immediately implies (3.3) when we define

$$C_1 \triangleq \bar{C}_1 \max \left\{ \max_{i \in N} \frac{1}{y_i^*}, \max_{i \in B} \frac{1}{x_i^*} \right\}.$$

For the feasible case $r^0 = 0$, note from the condition $y - Mx - q = \nu (y^0 - Mx^0 - q)$ that $r = y - Mx - q = 0$ for all (x, y) satisfying the assumptions of the theorem. Hence we can take $\nu = 0$ in (3.4) and obtain

$$x^{*T} y + x^T y^* \le x^T y.$$

A simple argument produces the required bound with C_1 defined by

$$C_1 = (x^{0T} y^0)^{\delta/(1+\delta)} \max \left\{ \max_{i \in N} \frac{1}{y_i^*}, \max_{i \in B} \frac{1}{x_i^*} \right\}. \qquad \square$$

We now use this lemma to find bounds on the nonbasic components of $(\Delta x, \Delta y)$.

LEMMA 3.2. *Suppose that Assumption 1 holds and that $(x^0, y^0) > 0$ is given. Assume that scalars $\delta \in [0, 1/3]$, $\eta > 0$, and $\rho \ge 0$ are given. Then, there exists a constant $C_2 \ge 0$ with the following property: For any vector pair $(x, y) \in \mathcal{N}(\delta, \eta, \rho)$ such that*

$$x^T y \le \min(1, x^{0T} y^0), \tag{3.6}$$

$$y - Mx - q = \nu (y^0 - Mx^0 - q) \text{ for some } \nu \in [0, 1/2], \tag{3.7}$$

we have

$$|\Delta x_i| \le C_2(x^T y)^{1-3\delta} \quad \forall i \in N, \tag{3.8a}$$

$$|\Delta y_i| \le C_2(x^T y)^{1-3\delta} \quad \forall i \in B. \tag{3.8b}$$

Proof. We first show that there exists a constant $\bar{C}_2$ such that

$$\max\{\|D^{-1}\Delta x\|, \|D\Delta y\|\} \le \bar{C}_2(x^T y)^{(1-3\delta)/2}. \tag{3.9}$$

We give the proof only for the case in which $r^0 \equiv y^0 - Mx^0 - q \neq 0$, since the proof for $r^0 = 0$ is trivial. The assumptions of the lemma imply that (2.12) and (3.5) hold. These two relations together with (3.6) then imply

$$\begin{aligned}
\max\{\nu\|x\|_\infty, \nu\|y\|_\infty\} &\le C_0\left[x^T y + \nu\right] \\
&\le C_0\left[1 + \frac{\rho^{1/(1+\delta)}}{\|r^0\|}\right](x^T y)^{1/(1+\delta)} \\
&= \hat{C}_2(x^T y)^{1/(1+\delta)} \\
&\le \hat{C}_2(x^T y)^{1-\delta}, \qquad (3.10)
\end{aligned}$$

where $\hat{C}_2$ is defined in an obvious way. Since the assumptions of Lemma 2.7 are satisfied, inequality (2.15) together with (3.10) and (3.6) then yields

$$\begin{aligned}
&\max\left(\|D^{-1}\Delta x\|, \|D\Delta y\|\right) \\
&\le (x^T y)^{1/2} + \frac{2\nu}{\eta^{1/2}(x^T y)^{(1+\delta)/2}}\left[\|Yu^0\| + \|Xv^0\|\right] \\
&\le (x^T y)^{1/2} + \frac{2}{\eta^{1/2}(x^T y)^{(1+\delta)/2}}\left(\|u^0\| + \|v^0\|\right)\max\left(\|\nu y\|_\infty, \|\nu x\|_\infty\right) \\
&\le (x^T y)^{1/2} + \frac{2\hat{C}_2}{\eta^{1/2}(x^T y)^{(1+\delta)/2}}\left(\|u^0\| + \|v^0\|\right)(x^T y)^{1-\delta} \\
&\le \left[1 + \frac{2\hat{C}_2}{\eta^{1/2}}\left(\|u^0\| + \|v^0\|\right)\right](x^T y)^{(1-3\delta)/2} \\
&= \bar{C}_2(x^T y)^{(1-3\delta)/2},
\end{aligned}$$

where $\bar{C}_2$ is defined in an obvious way. We have thus shown that (3.9) holds.

Considering $i \in N$, we have from the definition $D = X^{1/2}Y^{-1/2}$, the formulae (3.9) and (3.3a), and the inclusion $(x, y) \in \mathcal{N}(\delta, \eta, \rho)$ that

$$\begin{aligned}
|\Delta x_i| &\le \left(\frac{x_i}{y_i}\right)^{1/2} \bar{C}_2(x^T y)^{(1-3\delta)/2} \\
&= \frac{x_i}{(x_i y_i)^{1/2}}\bar{C}_2(x^T y)^{(1-3\delta)/2} \\
&\le \bar{C}_2\frac{C_1(x^T y)^{1/(1+\delta)}}{\eta^{1/2}(x^T y)^{(1+\delta)/2}}(x^T y)^{(1-3\delta)/2} \\
&= (\bar{C}_2 C_1 \eta^{-1/2})(x^T y)^{(1-2\delta-2\delta^2)/(1+\delta)}. \qquad (3.11)
\end{aligned}$$

Since $\delta \in [0, 1/3]$, we have

$$\frac{1 - 2\delta - 2\delta^2}{1 + \delta} \geq (1 - 2\delta - 2\delta^2)(1 - \delta) = 1 - 3\delta + 2\delta^3 \geq 1 - 3\delta \geq 0,$$

and since $x^T y \leq 1$, we conclude from (3.11) that

$$|\Delta x_i| \leq (\bar{C}_2 C_1 \eta^{-1/2})(x^T y)^{1-3\delta} = C_2 (x^T y)^{1-3\delta},$$

for C_2 defined in an obvious way. Hence, we have proved (3.8a). The proof of (3.8b) is identical. □

To find bounds on the remaining components of $(\Delta x, \Delta y)$, we cite the following two results, which have been proved in earlier reports.

LEMMA 3.3 (see [4, Lemma 2.2]). *Let $f \in \mathsf{R}^q$ and $H \in \mathsf{R}^{p \times q}$ be given. Then there exists a constant $L = L(f, H)$ with the property that for any positive definite diagonal matrix D and any vector $h \in \text{Range}(H)$, the (unique) optimal solution $\bar{w} = \bar{w}(D, h)$ of*

$$\min_w \; f^T w + \frac{1}{2} \|Dw\|^2, \qquad \text{subject to } Hw = h, \tag{3.12}$$

satisfies

$$\|\bar{w}\|_\infty \leq L \left\{ |f^T \bar{w}| + \|h\|_\infty \right\}.$$

LEMMA 3.4 (see [7, Lemma 5.2]). *Let $(x, y) > 0$ be given, and consider the direction $(\Delta x, \Delta y)$ determined by (1.5). Then $(\Delta x_B, \Delta y_N)$ solves the convex quadratic programming problem*

$$\min_{(w,z)} \; \tfrac{1}{2}\|D_B w\|^2 + \tfrac{1}{2}\|D_N^{-1} z\|^2$$

subject to

$$\begin{aligned} M_{BB} w &= r_B - M_{BN} \Delta x_N + \Delta y_B, \\ M_{NB} w - z &= r_N - M_{NN} \Delta x_N. \end{aligned}$$

Lemma 3.4 is an extension to the infeasible case of a result due to Ye and Anstreicher [9, Lemma 3.5].

As a consequence of Lemmas 3.3 and 3.4, we can find bounds on Δx_B and Δy_N.

LEMMA 3.5. *Suppose the assumptions of Lemma 3.2 hold. Then there exists a constant $C_3 \geq 0$ such that*

$$|\Delta x_i| \leq C_3 (x^T y)^{1-3\delta} \quad \forall i \in B, \tag{3.13a}$$

$$|\Delta y_i| \leq C_3 (x^T y)^{1-3\delta} \quad \forall i \in N. \tag{3.13b}$$

Proof. From Lemmas 3.2, 3.3, and 3.4, we have

$$\begin{aligned} \|(\Delta x_B, \Delta y_N)\|_\infty &\leq L \left[\|r\|_\infty + \|M\|_\infty \|\Delta x_N\|_\infty + \|\Delta y_B\|_\infty \right] \\ &\leq L \left[(\rho x^T y)^{1/(1+\delta)} + (\|M\|_\infty + 1) C_2 (x^T y)^{1-3\delta} \right]. \end{aligned}$$

Since $1/(1+\delta) \geq 1-3\delta$ when $\delta \in [0,1/3]$, we obtain the desired result by setting

$$C_3 = L\left[\rho^{1/(1+\delta)} + (\|M\|_\infty + 1)C_2\right]. \qquad \square$$

As an immediate consequence of Lemmas 3.2 and 3.5, we obtain the following result.

THEOREM 3.6. *Suppose that Assumption* 1 *holds and that* $\delta \in [0,1/3]$ *is given. Then, there exists a constant* $C_4 \geq 0$ *such that for all* k *sufficiently large, we have*

$$|\Delta x_i^k| \leq C_4 (x^{k^T} y^k)^{1-3\delta} \quad \forall i = 1,2,\ldots,n, \tag{3.14a}$$

$$|\Delta y_i^k| \leq C_4 (x^{k^T} y^k)^{1-3\delta} \quad \forall i = 1,2,\ldots,n. \tag{3.14b}$$

Proof. Since the assumptions of Theorem 2.8 are satisfied, we know that $x^{k^T} y^k \downarrow 0$ and $\|r^k\| \to 0$. Hence, in view of Corollary 2.4(b), we have that $\nu_k \to 0$. Thus, there exists an index $K \geq 0$ such that $\nu_k \leq 1/2$ and $x^{k^T} y^k \leq 1$ for all $k \geq K$. The above observations together with Corollary 2.4 imply that the assumptions of Lemmas 3.2 and 3.5 are satisfied by all points (x^k, y^k) with $k \geq K$. Thus, we obtain (3.14) by choosing $C_4 = \max(C_2, C_3)$. $\square$

COROLLARY 3.7. *Suppose the assumptions of Theorem* 3.6 *are satisfied. Then there is a constant* $C_5 \geq 0$ *such that for all* k *sufficiently large, we have*

$$|\phi_k| \leq C_5 (x^{k^T} y^k)^{1-6\delta}, \qquad 0 \leq \chi_k \leq C_5 (x^{k^T} y^k)^{1-7\delta}. \tag{3.15}$$

Proof. For sufficiently large k, we have from (3.14) that

$$|\phi_k| = \frac{|\Delta x^{k^T} \Delta y^k|}{x^{k^T} y^k} \leq n C_4^2 (x^{k^T} y^k)^{1-6\delta},$$

as required. Since $(x^k, y^k) \in \mathcal{N}(\delta, \eta, \rho)$, we also have that

$$\chi_k \leq \max_{i=1,2,\ldots,n} \frac{|\Delta x_i^k \Delta y_i^k|}{x_i^k y_i^k} \leq \frac{C_4^2 (x^{k^T} y^k)^{2-6\delta}}{\eta (x^{k^T} y^k)^{1+\delta}} = (C_4^2/\eta)(x^{k^T} y^k)^{1-7\delta}.$$

The result follows if we set $C_5 = \max(nC_4^2, C_4^2/\eta)$. $\square$

We are now ready to prove the main local convergence result.

THEOREM 3.8. *Suppose that Assumption* 1 *holds and that* $\delta \in (0,1/7)$ *is given. Then the sequence* $\{x^{k^T} y^k\}$ *converges Q-superlinearly to zero with Q-order at least* $2-7\delta$.

Proof. Suppose the index K is chosen sufficiently large so that the relation $x^{k^T} y^k \leq 1$ and the bounds in (3.15) hold for all $k \geq K$. Then we have

$$\begin{aligned}
1 - \alpha_k &= \frac{\chi_k + (1+\delta)\phi_k^+}{\delta + \chi_k + (1+\delta)\phi_k^+} \\
&\leq \frac{\chi_k + (1+\delta)\phi_k^+}{\delta} \\
&\leq \frac{C_5 [1 + (1+\delta)(x^{k^T} y^k)^\delta] (x^{k^T} y^k)^{1-7\delta}}{\delta} \\
&\leq \frac{3C_5}{\delta} (x^{k^T} y^k)^{1-7\delta}.
\end{aligned}$$

Hence, from Lemma 2.2(a), we have for all $k \geq K$ that

$$\begin{aligned} {x^{k+1}}^T y^{k+1} &\leq {x^k}^T y^k (1 - \alpha_k + \alpha_k^2 \phi_k) \\ &\leq {x^k}^T y^k (1 - \alpha_k + |\phi_k|) \\ &\leq {x^k}^T y^k \left(\frac{3C_5}{\delta} ({x^k}^T y^k)^{1-7\delta} + C_5 ({x^k}^T y^k)^{1-6\delta} \right) \\ &\leq \frac{4C_5}{\delta} ({x^k}^T y^k)^{2-7\delta}, \end{aligned}$$

as required. □

Convergence (and therefore boundedness) of the sequence $\{(x^k, y^k)\}$ follows as a corollary of Theorem 3.8.

COROLLARY 3.9. *Suppose the assumptions of Theorem* 3.8 *hold. Then the sequence* $\{(x^k, y^k)\}$ *is convergent.*

Proof. It suffices to show that the sequence is Cauchy. From Theorem 3.8, there exists an index $\bar{K}$ such that ${x^{k+1}}^T y^{k+1} \leq (1/2)^{1/(1-3\delta)} {x^k}^T y^k$ for all $k \geq \bar{K}$. Hence, from (3.14) we have for k_1, k_2 with $\bar{K} \leq k_1 < k_2$ that

$$\begin{aligned} \|(x^{k_2}, y^{k_2}) - (x^{k_1}, y^{k_1})\|_\infty &\leq \sum_{k=k_1}^{k_2-1} \|(\Delta x^k, \Delta y^k)\|_\infty \\ &\leq C_4 \sum_{k=k_1}^{k_2-1} \left({x^k}^T y^k \right)^{1-3\delta} \\ &\leq C_4 \left({x^{k_1}}^T y^{k_1} \right)^{1-3\delta} \sum_{k=k_1}^{k_2-1} \frac{1}{2^{k-k_1}} \\ &\leq 2C_4 \left({x^{k_1}}^T y^{k_1} \right)^{1-3\delta}. \end{aligned}$$

Since ${x^k}^T y^k \downarrow 0$, the above relation shows that, given any $\epsilon > 0$, we can choose $\hat{K} \geq \bar{K}$ such that

$$\|(x^{k_2}, y^{k_2}) - (x^{k_1}, y^{k_1})\|_\infty \leq \epsilon \quad \forall k_2 > k_1 \geq \hat{K},$$

and so the sequence $\{(x^k, y^k)\}$ is Cauchy. □

REFERENCES

[1] M. KOJIMA, N. MEGIDDO, AND S. MIZUNO, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Programming, 61 (1993), pp. 263–280.

[2] S. MIZUNO, *Polynomiality of infeasible-interior-point algorithms for linear programming*, Math. Programming, 67 (1994), pp. 109–119.

[3] S. MIZUNO, M. KOJIMA, AND M. TODD, *Infeasible interior-point primal-dual potential-reduction algorithms for linear programming*, SIAM J. Optim., 5 (1995), pp. 52–67.

[4] R. D. C. MONTEIRO AND S. WRIGHT, *Superlinear primal-dual affine scaling algorithms for* LCP, Math. Programming, 69 (1995), pp. 311–333.

[5] ———, *Local convergence of interior-point algorithms for degenerate monotone* LCPs, Comput. Optim. Appl., 3 (1994), pp. 131–155.

[6] F. A. POTRA, *A quadratically convergent predictor-corrector method for solving linear programs from infeasible starting points*, Math. Programming, 67 (1994), pp. 383–406.

[7] S. WRIGHT, *An infeasible interior point algorithm for linear complementarity problems*, Math. Programming, 67 (1994), pp. 29–51.

[8] S. WRIGHT, *A path-following infeasible-interior-point algorithm for linear complementarity problems*, Optimization Methods Software, 2 (1993), pp. 79–106.

[9] Y. YE AND K. ANSTREICHER, *On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for* LCP, Math. Programming, 62 (1993), pp. 537–551.

[10] Y. ZHANG, *On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

[11] Y. ZHANG AND D. ZHANG, *Superlinear convergence of infeasible-interior-point methods for linear programming*, Math. Programming, 66 (1994), pp. 361–377.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 19–32, February 1996
© 1996 Society for Industrial and Applied Mathematics
002

AN INFEASIBLE-INTERIOR-POINT PREDICTOR-CORRECTOR ALGORITHM FOR LINEAR PROGRAMMING*

FLORIAN A. POTRA†

Abstract. A predictor-corrector method is proposed for solving standard form linear programming problems starting from initial points that strictly satisfy the positivity constraints, but do not necessarily satisfy the equality constraints. The algorithm is globally convergent under the assumption that the linear program has an optimal solution. Under some additional assumptions on the starting point we prove that ϵ-feasibility and ϵ-complementarity can be obtained in $O(n\ln(\frac{1}{\epsilon}))$ iterations.

Key words. linear program, interior-point methods, infeasible starting point, polynomial complexity

AMS subject classifications. 90C05, 49D35

1. Introduction.

Following the publication of the celebrated article of Karmarkar [1] an impressive number of papers have been written on the convergence of different classes of interior-point methods for solving the linear programming problem in standard form. The vast majority of those papers assume that a starting point satisfying exactly the linear constraints and lying in the interior of the region defined by the inequality constraints is given. However, in practice it is sometimes very difficult to obtain such a starting point, and many efficient implementations of interior-point methods, like OB1 [3], use starting points that are in the interior of the region defined by the inequality constraints but do not necessarily satisfy the equality constraints. All points subsequently produced by the algorithm will have the same properties. Hence, as suggested in the recent work of Zhang [6], such a method is called an infeasible-interior-point method. Although many infeasible-interior-point methods have shown remarkable practical performance, no computational complexity results for any method in this class have been proved until very recently, when Zhang [6] showed that some variant of a primal-dual centered and damped Newton method can achieve ϵ-feasibility and ϵ-complementarity in $O(n^2\ln(\frac{1}{\epsilon}))$ iterations. We mention that the global convergence of an infeasible-interior-point algorithm had already been proved in [2], under the only assumption that an optimal primal-dual pair exists. The polynomiality is proved under the additional assumption that the starting point $x^0 = s^0 = \xi e$, where $e = (1, 1, \ldots, 1)$ is "large enough" in the sense that $x^* \leq \nu^* x^0$ and $s^* \leq \nu^* s^0$ for some optimal pair (x^*, s^*) and some ν^* independent of n.

Shortly after that, Mizuno [4] proved that a modification of the method of [2] has also $O(n^2\ln(\frac{1}{\epsilon}))$ iteration complexity. In a revised version of [4], Mizuno also proposed an infeasible-interior-point method that has $O(n\ln(1/\epsilon))$ iteration complexity under the same assumptions. A step of this algorithm consists of a predictor in the direction given by the Kojima–Megiddo–Mizuno [2] method followed by a predictor as in Mizuno–Todd–Ye [5]. In each step the "feasibility" is improved at a slightly faster rate than "optimality."

In this paper, we propose a new infeasible-interior-point predictor- corrector method so that both "feasibility" and optimality are improved at the same rate. We prove that the algorithm is globally convergent under the assumption that the prob-

* Received by the editors June 29, 1992; accepted for publication (in revised form) September 14, 1994. This paper was written while the author was visiting the University of Karlsruhe.

† Department of Mathematics, University of Iowa, Iowa City, IA 52240 (potra@math.uiowa.edu).

lem has an optimal solution. Our algorithm attains ϵ-feasibility and ϵ-complexity in $O(n\ln(1/\epsilon))$ iterations under the same assumptions as those considered in [4]. We note that in the first version of this paper, which was written independently of the first version of [4] and appeared immediately after the release of that paper, we only proved $O(n^{1.5}\ln(1/\epsilon))$ iteration complexity. After the release of the second version of [4], which contained the $O(n\ln(1/\epsilon))$-iteration predictor-corrector variant of the Kojima–Megiddo–Mizuno algorithm, we realized that with a minor modification of the proof we can also obtain $O(n\ln(1/\epsilon))$-iteration complexity.

A step of our method consists of three substeps. The first substep is taken in the direction of the Mizuno–Todd–Ye predictor, which results in an improved "optimality" while the feasibility is maintained constant. The second substep is also a predictor step and its role is to improve "feasibility" exactly at the same rate at which "optimality" has been improved in substep one. The last substep corresponds to the Mizuno–Todd–Ye corrector and it brings the points back to a certain neighborhood of the central path while maintaining "feasibility" and "optimality" constant.

We mention that the computation of the two predictor directions is done by using only one matrix factorization, so that we use only two matrix factorizations per step, the same as in the original Mizuno–Todd–Ye algorithm [5]. Our algorithm uses one more back-solve than the predictor-corrector variant of the Kojima–Megiddo–Mizuno algorithm of [4] but has the advantage of improving both "optimality" and "feasibility" at the same rate. Moreover, our first predictor direction coincides with the direction of the Mizuno–Todd–Ye predictor on which "optimality" is improved faster than on the direction of the predictor of [4].

The paper is organized as follows. In §2 we describe the algorithm and derive some of its basic properties. In §3 we prove the convergence of the algorithm under the assumption that an optimal solution exists. Finally, in §4 we show that if the parameters defining the starting point are large enough (big M initialization) then the algorithm has polynomial complexity.

The notation used throughout the paper is rather standard: capital letters denote matrices, lower case letters denote vectors, and greek letters denote scalars.

All vectors are considered to be column vectors. The components of a vector $u \in \mathbf{R}^n$ will be denoted by $[u]_i$, $i = 1,\dots,n$. The relation $u > 0$ is equivalent to $[u]_i > 0$, $i = 1,\dots,n$, while $u \geq 0$ means $[u]_i \geq 0$, $i = 1,\dots,n$. If $u \in \mathbf{R}^n$, $w \in \mathbf{R}^m$ then (u,w) denotes the column vector formed by the components of u and w, i.e., $(u,w) \in \mathbf{R}^{n+m}$, $[(u,w)]_i = [u]_i$ for $1 \leq i \leq n$ and $[(u,w)]_{n+j} = [w]_j$ for $1 \leq j \leq m$. If $u \in \mathbf{R}^n$ then $U = \mathrm{Diag}(u)$ denotes the diagonal matrix having the components of u as diagonal entries. The most used norm is the l_2-norm so that we write $\|\cdot\|$ instead of $\|\cdot\|_2$. Whenever we need other norms like $\|\cdot\|_1$ or $\|\cdot\|_\infty$ we use the corresponding symbol. If u and v are two n-dimensional vectors, then uv denotes their componentwise product, i.e., $u,v \in \mathbf{R}^n \Rightarrow uv \in \mathbf{R}^n$, $[uv]_i = [u]_i[v]_i$, $i = 1,\dots,n$. If $u \in \mathbf{R}^n$, $u > 0$ and $\kappa \in \mathbf{R}$ then $u^\kappa \in \mathbf{R}^n$, $[u^\kappa]_i = [u_i]^\kappa$, $i = 1,\dots,n$. For example $u^{1/2} = \left[[u_1]^{1/2},\dots,[u_n]^{1/2}\right]$. Superscripts are also used to denote elements of a sequence of vectors, but the difference should be obvious from the context.

If A is a matrix with m rows and n columns then $A^\dagger$ denotes its Moore-Penrose pseudoinverse. In this paper we always assume that $\mathrm{rank}(A) = m$ so that in this case $A^\dagger = A^T(AA^T)^{-1}$, where T denotes the transpose operation.

2. The predictor-corrector algorithm. In this paper we consider the standard form linear programming problem

$$\text{(LP)} \qquad \min\{c^T x : x \in \mathbf{R}^n, Ax = b, x \geq 0\},$$

together with its dual

$$\text{(DP)} \quad \max\{b^T y : y \in \mathbf{R}^m, A^T y \leq c\}.$$

It is well known that if the nonlinear system of equations

$$F(z) = 0,$$

where

$$z = (x, s, y),\ F(z) = (xs, Ax - b, s + A^T y - c)$$

has a solution $z^* = (x^*, s^*, y^*)$ such that $x^* \geq 0, s^* \geq 0$, then x^* is a solution of (LP) and y^* is a solution of (DP). We convene to call (x^*, s^*) an optimal primal-dual pair. In what follows, we assume that the solution set

$$\mathcal{F}^* = \{z = (x, s, y) : F(z) = 0, x \geq 0, s \geq 0\}$$

is nonempty, i.e., optimal primal-dual pairs exist. We describe an iterative algorithm that produces, for any given $\epsilon > 0$, a point in the ϵ-approximate solution set

$$\begin{aligned}\mathcal{F}_\epsilon = \{z = (x, s, y) : x > 0, s > 0, \|xs\|_1 = x^T s \leq \epsilon,\\ \|Ax - b\| \leq \epsilon, \|s + A^T y - c\| \leq \epsilon\}.\end{aligned}$$

The reason for taking the l_1-norm for the first component of $F(z)$ is due to the fact that in the feasible case the quantity $x^T s$ represents the primal-dual gap which is a good measure of optimality.

Our algorithm is an extension to the infeasible case of the predictor-corrector method of Mizuno, Todd, and Ye [5]. We analyze the convergence of our method for infeasible starting points of the form

$$z^0 = (x^0, s^0, y^0),\ x^0 = \xi e,\ s^0 = \sigma e,\ y^0 = 0, \tag{2.1}$$

where ξ and σ are positive numbers such that

$$\xi \geq \|A^\dagger b\|_\infty,\ \sigma \geq \|c\|_\infty. \tag{2.2}$$

With the above choice we have the initial residuals

$$\begin{aligned}\mu_0 &= \frac{1}{n}(x^0)^T s^0 = \xi\sigma,\\ p^0 &= b - Ax^0 = b - \xi Ae,\\ q^0 &= c - s^0 - A^T y^0 = c - \sigma e.\end{aligned}$$

We note that our starting pair (x^0, s^0) is perfectly centered in the sense that $X^0 s^0 = \mu_0 e$. Our algorithm will produce pairs belonging to the set of approximately centered pairs defined by

$$\mathcal{N}_\alpha = \{(x, s) : x > 0, s > 0, \|Xs - \mu e\| \leq \alpha\mu\},$$

where $\alpha \in (0, 1)$ is a given parameter (we take $\alpha = 0.25$) and where $\mu = \frac{1}{n} x^T s$.

Suppose now that we are given a $z = (x, s, y)$ such that $(x, s) \in \mathcal{N}_\alpha$, and we denote

$$\mu = \frac{1}{n} x^T s,\ p = b - Ax,\ q = c - s - A^T y. \tag{2.3}$$

We want to produce a new point $z^+ = (x^+, s^+, y^+)$ such that $(x^+, s^+) \in \mathcal{N}_\alpha$ and the new residuals satisfy

$$(\mu_+, p^+, q^+) = (1 - \hat{\varphi})(\mu, p, q), \tag{2.4}$$

where $\hat{\varphi} \in (0, 1]$ is a number to be defined later. The new point z^+ will be obtained in several substeps that can be interpreted as modified Newton steps for the equation $F(z) = 0$. We note that the Jacobian of $F(z)$ is of the form

$$F'(z) = \begin{bmatrix} S & X & 0 \\ A & 0 & 0 \\ 0 & I & A^T \end{bmatrix}. \tag{2.5}$$

The following result concerning the nonsingularity of this matrix is well known.

LEMMA 2.1. *If $(x, s) > 0$ and A has full rank, then the matrix* (2.5) *is nonsingular.*

We use the matrix (2.5) to compute a direction $r = (u, v, w)$, along which we can decrease μ, by solving the linear system

$$F'(z)r = (-Xs, 0, 0). \tag{2.6}$$

Then we compute a direction $\bar{r} = (\bar{u}, \bar{v}, \bar{w})$, along which we can decrease the residuals p and q, by solving

$$F'(z)\bar{r} = (0, p, q). \tag{2.7}$$

Using these two directions we define an intermediate point

$$\hat{z} = (\hat{x}, \hat{s}, \hat{y}) = z + \hat{\theta}r + \hat{\varphi}\bar{r} \tag{2.8}$$

for some parameters $\hat{\theta}, \hat{\varphi}$ with $\hat{\varphi} \in (0, 1]$. We will give a precise way of choosing these parameters such that the intermediate point satisfies

$$(\hat{x}, \hat{s}) \in \mathcal{N}_\beta, \tag{2.9}$$

where β is a parameter such that

$$0 < \alpha < \beta \leq 2\alpha \leq \frac{\sqrt{2}}{1 + \sqrt{2}}. \tag{2.10}$$

Finally a direction $\hat{r}$ is computed, along which the amount of centering can be increased, by solving the system

$$F'(\hat{z})\hat{r} = (\hat{\mu}e - \hat{x}\hat{s}, 0, 0). \tag{2.11}$$

The new point is then obtained by setting

$$z^+ = (x^+, s^+, y^+) = \hat{z} + \hat{r}. \tag{2.12}$$

We will prove that, for properly chosen $\hat{\theta}$ and $\hat{\varphi}$, the new point z^+ satisfies $(x^+, s^+) \in \mathcal{N}_\alpha$ so that the algorithm described above can be repeated with z^+ instead of z. First, let us show how to choose $\hat{\theta}$ and $\hat{\varphi}$ such that both (2.4) and (2.9) are satisfied. For arbitrary θ and φ we define

$$\check{z}(\theta, \varphi) = (\check{x}, \check{s}, \check{y}) = z + \theta r + \varphi\bar{r}, \tag{2.13}$$

where r and $\bar{r}$ are the solutions of (2.6) and (2.7), respectively. We have

$$Su + Xv = -Xs,\ Au = 0,\ v + A^T w = 0 \tag{2.14}$$

and

$$S\bar{u} + X\bar{v} = 0,\ A\bar{u} = p,\ \bar{v} + A^T\bar{w} = q. \tag{2.15}$$

Using the above equations we can immediately check that

$$\check{p} = b - A\check{x} = (1-\varphi)p,\ \check{q} = c - \check{s} - A^T\check{y} = (1-\varphi)q.$$

Also, we have

$$\begin{aligned}\check{\mu} &= \frac{1}{n}(x + \theta u + \varphi\bar{u})^T(s + \theta v + \varphi\bar{v}) \\ &= \frac{1}{n}[x^T s + \theta(s^T u + x^T v) + \theta^2 u^T v \\ &\quad + \theta\varphi(u^T\bar{v} + v^T\bar{u}) + \varphi^2\bar{u}^T\bar{v}],\end{aligned}$$

so that by using $u^T v = 0$ we can write

$$\check{\mu} = [(1-\theta) + \gamma\theta\varphi + \tau\varphi^2]\mu,$$

with

$$\gamma = \frac{u^T\bar{v} + v^T\bar{u}}{n\mu},\ \tau = \frac{\bar{u}^T\bar{v}}{n\mu}. \tag{2.16}$$

It follows that

$$\check{\mu} = (1-\varphi)\mu \tag{2.17}$$

if and only if

$$\theta = \chi(\varphi) = \frac{1 + \tau\varphi}{1 - \gamma\varphi}\varphi. \tag{2.18}$$

From (2.13)–(2.15) we obtain

$$\check{X}\check{s} = (1-\theta)Xs + \theta^2 h + \theta\varphi g + \varphi^2 t,$$

where

$$h = Uv,\ g = U\bar{v} + V\bar{u},\ t = \bar{U}\bar{v}.$$

Hence

$$\check{X}\check{s} - \check{\mu}e = (1-\theta)f + \theta^2 h + \varphi\theta\bar{g} + \varphi^2\bar{t},$$

with

$$f = Xs - \mu e,\ \bar{g} = g - \frac{e^T g}{n}g,\ \bar{t} = t - \frac{e^T t}{n}t. \tag{2.19}$$

The condition

$$\|\check{X}\check{s} - \check{\mu}e\| = \beta\check{\mu}$$

is equivalent to the equation

$$\begin{aligned}
&(1-\theta)^2\|f\|^2 + \theta^4\|h\|^2 + \|\bar{g}\|^2\varphi^2\theta^2 + \|\bar{t}\|^2\varphi^4 \\
&+2\theta^2(1-\theta)f^Th + 2(1-\theta)\varphi\theta f^T\bar{g} + 2(1-\theta)\varphi^2 f^T\bar{t} \\
&+2\theta^3\varphi h^T\bar{g} + 2\theta^2\varphi^2 h^T\bar{t} + 2\theta\varphi^3\bar{g}^T\bar{t} = \beta^2\check{\mu}^2.
\end{aligned}$$

Now by substituting $\check{\mu}$ from (2.17) and θ from (2.18), and by multiplying both sides by $(1-\gamma\varphi)^4$ we obtain an equation of the form

$$\varphi R(\varphi) = \beta^2\mu^2 - \|f\|^2, \tag{2.20}$$

where $R(\varphi)$ is the polynomial of order seven in φ:

$$\begin{aligned}
R(\varphi) = {} & (1+4\gamma) - 2\gamma(2+3\gamma)\varphi + 2\gamma^2(3+2\gamma)\varphi^2 \\
& - \gamma^3(4+\gamma)\varphi^3 + \gamma^4\varphi^4 \\
& + (1+\tau\varphi)(\tau\varphi^2 + (2\gamma+1)\varphi - 2)(1-\gamma\varphi)^2\|f\|^2 \\
& + (1+\tau\varphi)^4\varphi^3\|h\|^2 + (1-\gamma\varphi)^4\varphi^3\|t\|^2 \\
& + (1+\tau\varphi)^2(1-\gamma\varphi)^2\varphi^3\|\bar{g}\|^2 \\
& + 2(1+\tau\varphi)^2(1-\varphi-\tau\varphi^2)(1-\gamma\varphi)\varphi f^Th \\
& + 2(1+\tau\varphi)(1-\varphi-\tau\varphi^2)(1-\gamma\varphi)^2\varphi f^T\bar{g} \\
& + 2(1-\varphi-\tau\varphi^2)(1-\gamma\varphi)^3\varphi f^T\bar{t} \\
& + 2(1+\tau\varphi)^3(1-\gamma\varphi)\varphi^3 h^T\bar{g} \\
& + 2(1+\tau\varphi)^2(1-\gamma\varphi)^2\varphi^3 h^T\bar{t}.
\end{aligned}$$

Taking into account the inequality

$$\beta^2\mu^2 - \|f\|^2 \geq (\beta^2 - \alpha^2)\mu^2 > 0,$$

and the fact that the left-hand side of (2.20) vanishes when $\varphi = 0$, it follows that there is $\varphi_0 > 0$ such that

$$\varphi R(\varphi) \leq \beta^2\mu^2 - \|f\|^2 \tag{2.21}$$

for all $0 \leq \varphi \leq \varphi_0$. If (2.20) has no roots in the interval $(0,1)$, then (2.21) holds for all $\varphi \in [0,1]$. Therefore let us define

$$\hat{\varphi} = \min\left(\{1\} \cup \{\varphi \in (0,1) : \varphi R(\varphi) = \beta^2\mu^2 - \|f\|^2\}\right), \tag{2.22}$$

where the last set in (2.22) may be empty. It follows that (2.21) holds for all $\varphi \in [0, \hat{\varphi}]$. Let us denote $\mu(\varphi) = (1-\varphi)\mu$ and

$$z(\varphi) = (x(\varphi), s(\varphi), y(\varphi)) = \check{z}(\chi(\varphi), \varphi),$$

where $\check{z}(\theta, \varphi)$ is given by (2.13) and $\chi(\varphi)$ is defined in (2.18). We want to show that

$$(x(\varphi), s(\varphi)) \in \mathcal{N}_\beta,\ 0 \leq \varphi \leq \hat{\varphi}.$$

As inequality (2.21) is equivalent to

$$\|X(\varphi)s(\varphi) - \mu(\varphi)e\| \leq \beta\mu(\varphi), \tag{2.23}$$

we only have to prove that

$$x(\varphi) > 0,\ s(\varphi) > 0,\ 0 \leq \varphi \leq \hat{\varphi}. \tag{2.24}$$

We have obviously $x(0) = x > 0$ and $s(0) = s > 0$, so that if (2.24) fails to be satisfied, then there must exist a $\varphi \in [0, \hat{\varphi}]$ and an index i such that

$$[x(\varphi)]_i[s(\varphi)]_i = 0.$$

But in this case we have

$$\begin{aligned}\|X(\varphi)s(\varphi) - \mu(\varphi)e\| &\geq \|X(\varphi)s(\varphi) - \mu(\varphi)e\|_\infty \\ &\geq |[x(\varphi)]_i[s(\varphi)]_i - \mu(\varphi)| = \mu(\varphi)\end{aligned}$$

which contradicts (2.23). In particular $z(\hat{\varphi}) = \mathcal{N}_\beta$, and we define

$$\hat{z} = z(\hat{\varphi}),$$

i.e., $\hat{z}$ is given by (2.8) with $\hat{\varphi}$ defined by (2.22) and $\hat{\theta} = \chi(\hat{\varphi})$. Let us note that if $\hat{\varphi} = 1$ then $\hat{z} \in \mathcal{F}^*$ and our algorithm terminates because an optimal solution is found.

If $\hat{\varphi} < 1$ then a new point z^+ is obtained as in (2.11)–(2.12). The fact that $(x^+, s^+) \in \mathcal{N}_\alpha$ follows from the following proposition which is proved in [5] for $\alpha = \frac{1}{4}$ and $\beta = \frac{1}{2}$. The proof holds for any α and β satisfying (2.10).

PROPOSITION 2.2. *Suppose* $(\hat{x}, \hat{s}) \in \mathcal{N}_\beta$ *where* β *satisfies* (2.10) *and let* z^+ *be defined by* (2.11)–(2.12). *Then* $(x^+, s^+) \in \mathcal{N}_\alpha$.

We now have all necessary ingredients for formally describing our algorithm.

ALGORITHM 2.3.

A1 (Initialization)

A1.1 Choose numbers ξ and σ such that (2.2) is satisfied and define the starting point z^0 as in (2.1).

A1.2 Choose tolerance $\epsilon \geq 0$.

A2 (Iteration)

A2.1 Set $x = x^k$, $s = s^k$ and define μ, p, q as in (2.3).

A2.2 If

$$x^T s \leq \epsilon,\ \|p\| \leq \epsilon,\ \|q\| \leq \epsilon$$

then report $z \in \mathcal{F}_\epsilon$ and terminate.

A2.3 Find the solution $r = (u, v, w)$ of the linear system (2.6).

A2.4 Find the solution $\bar{r} = (\bar{u}, \bar{v}, \bar{w})$ of the linear system (2.7).

A2.5 Compute the quantity $\hat{\varphi}$ defined by (2.22) and set $\hat{\theta} = \chi(\hat{\varphi})$ with $\chi(\varphi)$ as in (2.18).

A2.6 Compute $\hat{z}$ as in (2.8).

A2.7 If $\hat{\varphi} = 1$ then report $\hat{z} \in \mathcal{F}^*$ and terminate.

A2.8 Compute z^+ as in (2.11)–(2.12).

A2.9 Set $z^{k+1} = z^+$, $\hat{\varphi}_k = \hat{\varphi}$.

In practice we only need to find an appropriate lower bound of $\hat{\varphi}$ defined at (2.22). For example one would take $\hat{\varphi} = \hat{\gamma}$ where $\hat{\gamma}$ is defined in Lemma 3.5 (see the observation concerning polynomiality following Theorem 4.1).

We note that if $\epsilon = 0$ then the above algorithm terminates only when an optimal solution $z \in \mathcal{F}^*$ is found. For any $\epsilon \geq 0$ we define

$$K_\epsilon = \begin{cases} k & \text{if Algorithm 2.3 terminates at step } k, \\ +\infty & \text{otherwise.} \end{cases}$$

By summarizing the discussions made in this section we obtain the following result.

THEOREM 2.4. *For any integer k, $1 \leq k \leq K_\epsilon$, we have $(x^k, s^k) \in \mathcal{N}_\alpha$ and the corresponding residuals satisfy*

$$p^k = \psi_k p^0, \; q^k = \psi_k q^0, \; \mu_k = \psi_k \mu_0,$$

where

$$\psi_k = \prod_{i=0}^{k-1} (1 - \hat{\varphi}_k).$$

3. Global convergence. In this section we prove that if the set $\mathcal{F}^*$ is nonempty then Algorithm 2.3 with $\epsilon = 0$ either finds a $z \in \mathcal{F}^*$ in a finite number of steps or generates a sequence $\{z^k\}$ such that

$$\lim_{k\to\infty} \mu_k = 0, \; \lim_{k\to\infty} p^k = 0, \; \lim_{k\to\infty} q^k = 0.$$

First let us show that the sequence (x^k, s^k) generated by this algorithm is bounded.

LEMMA 3.1. *If $\mathcal{F}^*$ is nonempty then the sequence (x^k, s^k) generated by Algorithm 2.3 satisfies*

$$\xi \|s^k\|_1 + \sigma \|x^k\|_1 \leq 2n\mu_0 + \xi \|s^*\|_1 + \sigma \|x^*\|_1$$

for any $(x^*, s^*, y^*) \in \mathcal{F}^*$ and any $k \leq K_\epsilon$.

Proof. For simplicity we drop the index k. From Theorem 2.4 we have

$$\mu = \psi \mu_0 = \psi \sigma \xi, \; p = \psi p^0, \; q = \psi q^0$$

so that for any $(x^*, s^*, y^*) \in \mathcal{F}^*$ we obtain

$$\begin{aligned} A(\psi x^0 + (1-\psi)x^* - x) &= \psi(Ax^0 - Ax^*) + A(x^* - x) \\ &= \psi(Ax^0 - b) + b - Ax \\ &= -\psi p^0 + p = 0 \end{aligned}$$

and

$$\begin{aligned} \psi s^0 + (1-\psi)s^* - s &= \psi(s^0 - s^*) - (s - s^*) \\ &= \psi\left(c - q^0 - (c - A^T y^*)\right) - \left(c - q - A^T y - (c - A^T y^*)\right) \\ &= \psi A^T y^* - \psi q^0 - A^T y^* + q + A^T y \\ &= (\psi - 1)A^T y^* + A^T y \in \text{Range} A^T. \end{aligned}$$

Hence

$$[\psi x^0 + (1-\psi)x^* - x]^T[\psi s^0 + (1-\psi)s^* - s] = 0.$$

By performing the multiplication and by substituting $x^0 = \xi e$, $s^0 = \sigma e$, $\mu_0 = \xi\sigma$, $\psi\xi\sigma = \mu$ we obtain

$$\begin{aligned} 0 = {} & \psi^2 n\xi\sigma + \psi(1-\psi)(\xi\|s^*\|_1 + \sigma\|x^*\|_1) \\ & - \psi(\xi\|s\|_1 + \sigma\|x\|_1) + x^T s - (1-\psi)(s^T x^* + x^T s^*). \end{aligned}$$

The last term is negative, so that by omitting it we deduce the inequality

$$0 \le n\psi\mu + \psi(1-\psi)(\xi\|s^*\|_1 + \sigma\|x^*\|_1) - \psi(\xi\|s\|_1 + \sigma\|x\|_1) + n\mu.$$

Therefore

$$\begin{aligned} \xi\|s\|_1 + \sigma\|x\|_1 &\le \frac{(1+\psi)}{\psi} n\mu + (1-\psi)(\xi\|s^*\|_1 + \sigma\|x^*\|_1) \\ &= (1+\psi)n\mu_0 + (1-\psi)(\xi\|s^*\|_1 + \sigma\|x^*\|_1) \\ &\le 2n\mu_0 + \xi\|s^*\|_1 + \sigma\|x^*\|_1. \qquad \square \end{aligned}$$

In this section we often use the following technical result which is essentially proved in [5].

LEMMA 3.2. *If $u, v \in \mathbf{R}^n$ and $u^T v = 0$ then*

$$\|Uv\| \le \frac{1}{\sqrt{2}}\|Uv\|_1 \le \frac{1}{\sqrt{2}}\|u\|\|v\| \le \frac{1}{2\sqrt{2}}(\|u\|^2 + \|v\|^2).$$

In the next lemma we give some facts about the solutions of linear systems of the type used in our algorithm. For a given $(x, s) > 0$ we denote

$$D = X^{-\frac{1}{2}} S^{\frac{1}{2}}, \; T = (XS)^{-\frac{1}{2}}.$$

Let P and Q denote the orthogonal projections on $N(AD^{-1})$ and $R(D^{-1}A^T)$, respectively. Because A is full rank we can write

$$Q = D^{-1}A^T(AD^{-2}A^T)^{-1}AD^{-1}, \; P = I - Q.$$

LEMMA 3.3. *If $(x, s) \in \mathbf{R}^{2n}_{++}$ and $A \in \mathbf{R}^{m\times n}$ has full rank then for any $(h, p, q) \in \mathbf{R}^{2n+m}$ the linear system*

$$Su + Xv = h, \; Au = p, \; v + A^T w = q$$

has a unique solution (u, v, w) given by

$$Du = P(Th) + QDA^{\dagger}p - P(D^{-1}q), \; D^{-1}v = Q(Th) - QDA^{\dagger}p + P(D^{-1}q).$$

Before we state the next lemma it is convenient to introduce the following notation:

$$\gamma_1 = \frac{1}{\xi}\|A^{\dagger}b\|_\infty, \; \gamma_2 = \frac{1}{\sigma}\|c\|_\infty. \tag{3.1}$$

Note that (2.2) implies

$$\gamma_1 \le 1, \; \gamma_2 \le 1. \tag{3.2}$$

LEMMA 3.4. *Let* $(x^*, s^*, y^*) \in \mathcal{F}^*$ *and denote*

$$\sigma^* = \frac{1}{n\sigma}\|s^*\|_1,\ \xi^* = \frac{1}{n\xi}\|x^*\|_1.$$

If $\bar{r} = (\bar{u}, \bar{v}, \bar{w})$ *is obtained in* A2.4 *at step* k *of Algorithm* 2.3, *then*

$$\bar{u} = u_p + u_q,\ \bar{v} = v_p + v_q, \tag{3.3}$$

$$Du_p = -D^{-1}v_p = QDA^\dagger p,\ D^{-1}v_q = -Du_q = PD^{-1}q, \tag{3.4}$$

$$\|Du_p\| \le \eta\mu^{\frac{1}{2}},\ \|D^{-1}v_q\| \le \nu\mu^{\frac{1}{2}}, \tag{3.5}$$

where

$$\eta = (1-\beta)^{-\frac{1}{2}}(\gamma_1 + 1)(2 + \sigma^* + \xi^*)n, \tag{3.6}$$

$$\nu = (1-\beta)^{-\frac{1}{2}}(\gamma_2 + 1)(2 + \sigma^* + \xi^*)n. \tag{3.7}$$

Proof. Equations (3.3) and (3.4) follow from Lemma 3.3, so that we only have to prove inequalities (3.5). We have

$$\begin{aligned} Du_p &= QDA^\dagger p = \psi QDA^\dagger p_0 \\ &= \psi QD^{-1}A^T(AD^{-2}A^T)^{-1}(AA^\dagger b - Ax^0) \\ &= \psi QD(A^\dagger b - x^0), \end{aligned}$$

$$\begin{aligned} \|Du_p\| &\le \psi\|QD(A^\dagger b - x^0)\| \le \psi\|D(A^\dagger b - x^0)\| \\ &\le \psi\|De\|_1(\|A^\dagger b\|_\infty + \|x^0\|_\infty) \le \psi\|De\|_1(\gamma_1 + 1)\xi, \\ \|De\|_1 &= \sum_{i=1}^n [x]_i^{-\frac{1}{2}}[s]_i^{\frac{1}{2}} = \sum_{i=1}^n ([x]_i[s]_i)^{-\frac{1}{2}}[s]_i \le ((1-\beta)\mu)^{-\frac{1}{2}}\|s\|_1. \end{aligned}$$

By virtue of Lemma 3.1, it follows that

$$\begin{aligned} \|Du_p\| &\le ((1-\beta)\mu)^{-\frac{1}{2}}(\gamma_1 + 1)\xi\|s\|_1\psi \\ &\le ((1-\beta)\mu)^{-\frac{1}{2}}(\gamma_1 + 1)\psi(2n\mu_0 + \xi\|s^*\|_1 + \sigma\|x^*\|_1) \\ &= ((1-\beta)\mu)^{-\frac{1}{2}}(\gamma_1 + 1)(2 + \sigma^* + \xi^*)n\mu \\ &= (1-\beta)^{-\frac{1}{2}}(\gamma_1 + 1)(2 + \sigma^* + \xi^*)n\mu^{\frac{1}{2}}, \end{aligned}$$

which proves the first inequality of (3.5). In order to prove the second inequality of (3.5) we write

$$\begin{aligned} \|D^{-1}v_q\| &\le \|D^{-1}q\| \le \|D^{-1}e\|_1\|q\|_\infty = \|D^{-1}e\|_1\|q^0\|_\infty\psi, \\ \|D^{-1}e\|_1 &= \sum_{i=1}^n [x]_i^{\frac{1}{2}}[s]_i^{-\frac{1}{2}} \le ((1-\beta)\mu)^{-\frac{1}{2}}\|x\|_1, \\ \|q^0\|_\infty &= \|c - s^0\|_\infty \le \|c\|_\infty + \|s^0\|_\infty = (\gamma_2 + 1)\sigma, \end{aligned}$$

and by using Lemma 3.1, we get

$$\begin{aligned} \|D^{-1}v_q\| &\le ((1-\beta)\mu)^{-\frac{1}{2}}(1 + \gamma_2)\sigma\|x\|_1\psi \\ &\le ((1-\beta)\mu)^{-\frac{1}{2}}(1 + \gamma_2)\psi(2n\mu_0 + \xi\|s^*\|_1 + \sigma\|x^*\|_1) \\ &= (1-\beta)^{-\frac{1}{2}}(1 + \gamma_2)(2 + \sigma^* + \xi^*)n\mu^{\frac{1}{2}}. \end{aligned}$$

□

LEMMA 3.5. *Let* $(x^*, s^*, y^*) \in \mathcal{F}^*$ *and consider the notation introduced in Lemma* 3.4. *If* $\hat{\varphi}_k$ *is the steplength obtained at* A2.5 *at step* k *of Algorithm* 2.3 *then*

$$\hat{\varphi}_k \geq \hat{\gamma},\ k = 0, 1, \ldots, K_\epsilon,$$

where

$$\begin{aligned}
\hat{\gamma} &= \min\left\{1, \frac{1}{2|\gamma|}, \frac{1}{\sqrt{|\tau|}}, \gamma_9\right\},\\
\gamma_9 &= \frac{4(\beta-\alpha)}{(\beta-\alpha)+\sqrt{(\beta-\alpha)^2+4(\beta-\alpha)\gamma_8}},\\
\gamma_8 &= \{2|\gamma+\tau| + 4\gamma_4 + 2\gamma_5 + \gamma_6\},\ \gamma_4 = \sqrt{2}n(1+\sqrt{|\tau|})^2,\\
\gamma_5 &= 2\sqrt{n}(\eta+\nu)(1+\sqrt{|\tau|}),\ \gamma_6 = (\eta^2 + \sqrt{2}\eta\nu + \nu^2),
\end{aligned} \tag{3.8}$$

with γ *and* τ *defined in* (2.16).

Proof. With the notation introduced at (2.13)–(2.19), we have the following inequality:

$$\|\check{X}\check{s} - \check{\mu}e\| \leq |1-\theta|\|f\| + \theta^2\|h\| + \varphi|\theta|\|\bar{g}\| + \varphi^2\|\bar{t}\|$$

for any $\theta \in \mathbf{R}, \varphi \in [0,1]$. From $(x,s) \in \mathcal{N}_\alpha$ we have

$$\|f\| \leq \alpha\mu,$$

and by applying Lemmas 3.2 and 3.3 we deduce

$$\|h\| \leq \frac{1}{2\sqrt{2}}(\|Du\|^2 + \|D^{-1}v\|^2) = \frac{n\mu}{2\sqrt{2}}.$$

On the other hand from Lemmas 3.2 and 3.4 we obtain

$$\begin{aligned}
\|\bar{g}\| \leq \|g\| &\leq 2(\|Du\|\|D^{-1}v_q\| + \|D^{-1}v\|\|Du_p\|) \leq 2\sqrt{n}(\eta+\nu)\mu,\\
\|\bar{t}\| \leq \|t\| &\leq \|U_p v_p\| + \|U_p v_q\| + \|U_q v_p\| + \|U_q v_q\|\\
&\leq \|Du_p\|^2 + \sqrt{2}\|Du_p\|\|D^{-1}v_q\| + \|D^{-1}v_q\|^2\\
&\leq (\eta^2 + \sqrt{2}\eta\nu + \nu^2)\mu = \gamma_6\mu.
\end{aligned}$$

Hence

$$\|\check{X}\check{s} - \check{\mu}e\| \leq \left[|1-\theta|\alpha + \theta^2\frac{n}{2\sqrt{2}} + \varphi|\theta|\sqrt{n}(\eta+\nu) + \varphi^2\gamma_6\right]\mu.$$

Consider now that θ is given by (2.18). From (2.16) and Lemma 3.4 we get

$$\begin{aligned}
|\gamma| &\leq \frac{|(Du)^T(D^{-1}v_q)| + |(D^{-1}v)^T(Du_p)|}{n\mu}\\
&\leq \frac{\nu+\eta}{\sqrt{n}},
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
0 \geq \tau &= \frac{(Du_p + Du_q)^T(D^{-1}v_p + D^{-1}v_q)}{n\mu}\\
&= -\frac{\|Du_p\|^2 + \|D^{-1}v_q\|^2}{n\mu}\\
&\geq -\frac{(\eta^2+\nu^2)}{n}.
\end{aligned} \tag{3.10}$$

If

$$\varphi \le \gamma_7 = \min\left\{1, \frac{1}{2|\gamma|}, \frac{1}{\sqrt{|\tau|}}\right\}, \tag{3.11}$$

then

$$1 - \theta = 1 - \phi - \frac{\gamma + \tau}{1 - \gamma\phi}\phi^2 \tag{3.12}$$

and

$$|\theta| = \frac{|1 + \tau\phi|}{|1 - \gamma\phi|}\phi \le 2|1 + \tau\phi|\phi \le 2(1 + \sqrt{|\tau|})\phi$$

so that we obtain

$$\begin{aligned}\|\check{X}\check{s} - \check{\mu}e\| &\le (1 - \varphi)\alpha\mu + (2|\gamma + \tau|\alpha + 4\gamma_4 + 2\gamma_5 + \gamma_6)\varphi^2\mu \\ &\le (1 - \varphi)\alpha\mu + \gamma_8\varphi^2\mu.\end{aligned}$$

Let γ_9 be the positive root of the equation

$$(1 - \varphi)\alpha + \gamma_8\varphi^2 = (1 - \varphi)\beta,$$

i.e.,

$$\gamma_9 = \frac{2(\beta - \alpha)}{(\beta - \alpha) + \sqrt{(\beta - \alpha)^2 + 4(\beta - \alpha)\gamma_8}}.$$

From the above discussion it follows that

$$\|\check{X}\check{s} - \check{\mu}e\| \le \beta\check{\mu} \text{ for all } 0 \le \varphi \le \hat{\gamma}.$$

Finally according to the definition of $\hat{\varphi}$ we must have $\hat{\varphi} \ge \hat{\gamma}$. □

Using Lemma 3.5 and Theorem 2.4 we can easily prove the main result of this section which basically states that Algorithm 2.3 is globally convergent at a linear rate.

THEOREM 3.6. *Suppose that the set $\mathcal{F}^*$ is nonempty.* (i) *If $\epsilon = 0$ then Algorithm 2.3 either finds an optimal solution $z^* \in \mathcal{F}^*$ in a finite number of steps or produces an infinite sequence $z^k = (x^k, s^k, y^k)$ such that*

$$\lim(x^k)^T s^k = 0, \ \lim(b - Ax^k) = 0, \ \lim(c - A^T y^k - s^k) = 0.$$

(ii) *If $\epsilon > 0$ then Algorithm 2.3 terminates with a $z \in \mathcal{F}_\epsilon$ in at most $\tilde{k_\epsilon}$ iterations with*

$$\tilde{k_\epsilon} = \left\lceil \frac{|\ln(\frac{\epsilon}{\epsilon_0})|}{|\ln(1 - \hat{\gamma})|} \right\rceil, \tag{3.13}$$

where $\epsilon_0 = \max\{\mu_0, \|p^0\|, \|q^0\|\}$ and $\lceil\alpha\rceil$ denotes the smallest integer greater than or equal to α.

4. Polynomial complexity. In what follows we show that if the parameters ξ and σ that define the starting point are big enough, in the sense that the quantities σ^*, ξ^* from Lemma 3.4 satisfy

$$\sigma^* = O(1),\ \xi^* = O(1), \tag{4.1}$$

then the number $\tilde{k_\epsilon}$ from Theorem 3.6 is of the form

$$\tilde{k_\epsilon} = O\left(n \ln\left(\frac{\epsilon_0}{\epsilon}\right)\right).$$

Hence with a proper choice of ξ and σ Algorithm 2.3 has polynomial complexity. First let us examine carefully the quantities from (3.8). They depend on η and ν defined by (3.6) and (3.7). From (3.2) and (4.1) it follows that there is a constant $\lambda_1 \geq 1$ independent of n such that

$$\eta \leq \lambda_1 n,\ \nu \leq \lambda_1 n. \tag{4.2}$$

Then from (3.9), (3.10), and (4.2) we obtain

$$\begin{aligned} |\gamma| &\leq 2\lambda_1 \sqrt{n}, \\ 0 \geq \tau &\geq -2\lambda_1^2 n. \end{aligned} \tag{4.3}$$

Now (3.11) and (4.3) show that

$$\gamma\tau \geq \frac{1}{4\lambda_1 \sqrt{n}},$$

while from (3.8), (4.2)–(4.3) we deduce that there is a constant $\hat{\lambda}$, independent of n such that

$$1 > \hat{\gamma} \geq \frac{1}{\hat{\lambda} n}.$$

But then

$$|\ln(1-\hat{\gamma})| \geq \hat{\gamma},$$

so that from (3.13) it follows that

$$\tilde{k_\epsilon} \leq \left\lceil \hat{\lambda} n \ln\left(\frac{\epsilon_0}{\epsilon}\right) \right\rceil.$$

By summarizing the above arguments we obtain the following complexity result.

THEOREM 4.1. *Suppose that $\mathcal{F}^*$ is not empty and that the parameters σ and ξ are chosen big enough so that*

$$\frac{\|s^*\|_1}{n\sigma} + \frac{\|x^*\|}{n\xi} = O(1)$$

for some $z^* = (x^*, s^*, y^*) \in \mathcal{F}^*$. Then Algorithm 2.3 terminates in at most $\tilde{k_\epsilon} = O\left(n \ln(\frac{\epsilon_0}{\epsilon})\right)$ steps, where $\epsilon_0 = \max\{\xi\sigma, \|b - \xi Ae\|, \|c - \sigma e\|\}$.

Let us note that the above theorem does not imply yet that Algorithm 2.3 has polynomial complexity for problems with integer data of bitlength L. Indeed the definition (2.22) of $\hat{\varphi}$ involves the solution of an eighth-order algebraic equation and

there is no polynomial algorithm for finding the exact solution. Nevertheless if in step A2.5 we take instead of $\hat{\varphi}$ a lower bound $\tilde{\gamma} = \Omega(1/n)$ that can be computed in polynomial time then Algorithm 2.3 has polynomial complexity. Such a bound $\tilde{\gamma}$ can be obtained, for example, if we replace all square roots of the form $\sqrt{\omega}$ from the definition (3.8) of $\hat{\gamma}$ by the smallest positive integer m_ω such that $m_\omega^2 \geq \omega$. Because all quantities ω satisfy $\omega = O(n^2)$, the integers m_ω can obviously be computed in polynomial time.

We conclude the paper by showing that the stopping criterion in Algorithm 2.3 can be modified in such a way that the algorithm terminates either with a $z_\epsilon \in \mathcal{F}_\epsilon$ or indicates that $\mathcal{F}^*$ has no elements in a given region.

THEOREM 4.2. *Suppose that the following instruction*

A2.25 *If* $\xi\|s\|_1 + \sigma\|x\|_1 > 2n\xi\sigma + \xi\bar{\sigma} + \sigma\bar{\xi}$ *then terminate*

is inserted in between instructions A2.2 *and* A.2.3 *of Algorithm* 2.3. *Then the new algorithm terminates either at* A2.2 *with a* $z \in \mathcal{F}_\epsilon$ *or at* A2.25, *and in the latter case there is no* $z^* = (x^*, s^*, y^*) \in \mathcal{F}^*$ *such that*

$$\|x^*\|_1 \leq \bar{\xi},\ \|s^*\|_1 \leq \bar{\sigma}.$$

The proof of this theorem follows immediately from Lemma 3.1 and Theorem 3.6. For comments on the applicability of results of this type we refer to the excellent paper of Kojima, Meggido, and Mizuno [2].

Acknowledgments. I would like to thank the Mathematics Department of the University of Karlsruhe for their support during my visit.

REFERENCES

[1] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[2] M. KOJIMA, N. MEGIDDO, AND S. MIZUNO, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Programming, 61 (1993), pp. 263–280.

[3] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[4] S. MIZUNO, *Polynomiality of Kojima–Megiddo–Mizuno infeasible interior point algorithm for linear programming*, Tech. Report, School of OR and IE, Cornell University, Ithaca, NY, May 1992. (Revised July 1992.)

[5] S. MIZUNO, M. J. TODD, AND Y. YE, *On adaptive-step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., 18 (1993), pp. 964–981.

[6] Y. ZHANG, *On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 33–46, February 1996
© 1996 Society for Industrial and Applied Mathematics
003

ON LONG STEP PATH FOLLOWING AND SUMT FOR LINEAR AND QUADRATIC PROGRAMMING*

KURT M. ANSTREICHER†

Abstract. We consider a long step barrier algorithm for the minimization of a convex quadratic objective subject to linear inequality constraints. The algorithm is a dual version of a method developed by Anstreicher et al. [*Algorithmica*, 10 (1993), pp. 365–382], and requires $O(nL)$ or $O(\sqrt{n}L)$ iterations to solve a problem with n constraints, depending on how the barrier parameter is reduced. As a corollary of our analysis we demonstrate that the classical SUMT algorithm, exactly as implemented in 1968, solves linear and quadratic programs in $O(\sqrt{n}L \log L)$ iterations, with proper initialization and choice of parameters.

Key words. linear programming, quadratic programming, barrier function, long step path following, sequential unconstrained minimization technique

AMS subject classification. 90C05

1. Introduction.

Long step path following algorithms for linear programming, based on a logarithmic barrier function, were first developed by Gonzaga (1991) and Roos and Vial (1990). The analysis of such algorithms permits a large reduction in the barrier parameter μ and a linesearch of the barrier function by bounding the number of Newton steps between successive visits to a neighborhood of the central trajectory. This represents a considerable improvement over short step path following algorithms, first developed by Renegar (1988), where the reduction in μ must be very small to ensure that a single pure Newton step will return the current point to a neighborhood of the central path. (The terms "long step" and "short step" refer simply to the allowable magnitude of changes in μ.) The great interest in path following methods stems from the fact that these algorithms can attain the best complexity to date for linear programming: $O(\sqrt{n}L)$ iterations for a standard form problem with n variables and integer data of total bit size L. By comparison, Karmarkar's (1984) projective method obtains a complexity of $O(nL)$ iterations for the same problems. The relationship between Karmarkar's method and algorithms based on log barrier functions was first noted in Gill et al. (1986).

Path following methods are also attractive because of their easy adaptability to quadratic programming, linear complementarity problems, and general convex nonlinear programming. We refer the reader to Anstreicher et al. (1993) for a complete discussion of these extensions and for references. Anstreicher et al. (1993) develops a long step barrier algorithm for the minimization of a convex quadratic objective, subject to linear equalities in nonnegative variables. Here we consider a "dual" form of the same problem, with general linear inequality constraints, and obtain similar results. Although there are some interesting aspects to this recasting of Anstreicher et al. (1993) (in particular, the analysis of proximity measures in §3), our main purpose in considering the inequality form problem is to permit an unambiguous comparison with the classical sequential unconstrained minimization technique (SUMT) of Fi-

*Received by the editors January 25, 1991; accepted for publication (in revised form) August 24, 1994.

†Department of Management Science, University of Iowa, Iowa City, IA 52242 (kanstrei@scout-po.biz.uiowa.edu). This paper was written while the author was a visiting research scholar at the Department of Operations Research, George Washington University, Washington D.C.

acco and McCormick (1968), as implemented in the SUMT-3 and SUMT-4 programs (Mylander et al. (1971)). Our main result is of interesting historical significance. In particular, we show that the SUMT algorithm, exactly as implemented in 1968 (with proper initialization and choice of parameters) requires $O(\sqrt{n}L\log L)$ iterations to solve a linear or quadratic programming problem with n inequality constraints and integer data of total bit size L.

An extension of long step path following analysis to the general problem of minimizing a linear function, subject to convex inequality constraints, is presented in den Hertog et al. (1992a). It is worthwhile to note that the algorithm we develop here is essentially identical to the specialization of the algorithm in den Hertog et al. (1992a) to quadratic programming (although we consider the quadratic objective directly, rather than incorporate it into the constraints). However, our analysis of SUMT in §6 relies critically on quadratic convergence properties analogous to those developed in Anstreicher et al. (1993), rather than the weaker but more general results obtained in den Hertog et al. (1992a). The development in §§2–5 of this paper is closely related to that in §§2–5 of Anstreicher et al. (1993), and consequently we abbreviate or omit the proofs of several results here.

2. The central path. We consider a quadratic programming problem of the form

$$\begin{aligned} \text{QP} \qquad \min \quad & q(y) = \frac{1}{2}y^TQy + b^Ty \\ \text{s.t.} \quad & A^Ty \geq c, \end{aligned}$$

where $b \in \Re^m$, A is an $m\times n$ matrix, and Q is an $m\times m$ symmetric, positive semidefinite matrix. For the purpose of obtaining a complexity result we may assume that the data in QP is integral, with total bit size L. We also assume that the feasible region of QP is bounded and has an interior. It follows from the boundedness assumption that $m < n$, and the rows of A are linearly independent. It is well known that associated with QP is a dual quadratic program

$$\begin{aligned} \text{QD} \qquad \max \quad & d(x,y) = c^Tx - \frac{1}{2}y^TQy \\ \text{s.t.} \quad & Ax - Qy = b, \\ & x \geq 0. \end{aligned}$$

For y feasible in QP and $(\bar{x},\bar{y})$ feasible in QD, it is straightforward to show that

$$q(y) - d(\bar{x},\bar{y}) = \bar{x}^Ts + \frac{1}{2}(y-\bar{y})^TQ(y-\bar{y}) \geq 0, \tag{2.1}$$

where $s = A^Ty - c$ is a vector of slacks for the inequality constraints of QP. It follows that $d(\bar{x},\bar{y}) \leq z^* \leq q(y)$, where z^* denotes the optimal objective value in QP. It is also well known that if y^* is optimal in QP, then there is an x^* such that (x^*,y^*) is optimal in QD, with $d(x^*,y^*) = q(y^*) = z^*$.

Next, consider a logarithmic barrier function, defined on the interior of the feasible region of QP:

$$f(y,\mu) = \frac{q(y)}{\mu} - \sum_{j=1}^{n}\ln(a_j^Ty - c_j) = \frac{q(y)}{\mu} - \sum_{j=1}^{n}\ln(s_j),$$

where a_j denotes the jth column of A and μ is a positive scalar. The gradient and Hessian of $f(\cdot,\mu)$ at a given (y,μ) are then given by

$$
\begin{aligned}
g = \nabla f(y,\mu)^T &= \frac{Qy+b}{\mu} - \sum_{j=1}^{n} \frac{a_j}{a_j^T y - c_j} = \frac{Qy+b}{\mu} - As^{-1}, \\
H = \nabla^2 f(y,\mu) &= \frac{1}{\mu} Q + AS^{-2}A^T,
\end{aligned} \tag{2.2}
$$

where s^{-1} denotes the vector whose jth component is $1/s_j$, and S denotes the diagonal matrix diag(s). Note that H is positive definite, since the rows of A are independent. In addition, the set of y having $A^T y \geq c$ is compact, and consequently $f(\cdot,\mu)$ has a unique minimizer $y(\mu)$ for every $\mu > 0$. Let $s(\mu) = A^T y(\mu) - c$, and let $x(\mu) = \mu s^{-1}(\mu)$. It then follows from $\nabla f(y(\mu),\mu) = 0$ that $Qy(\mu)+b = Ax(\mu)$, so $(x(\mu), y(\mu))$ is feasible in QD, and $q(y(\mu)) - d(x(\mu), y(\mu)) = n\mu$. The set of minimizers $y(\mu)$, for $\mu > 0$, is commonly referred to as the central path, or central trajectory, for QP.

LEMMA 2.1. *The objective $q(y(\mu))$ of* QP *decreases as μ decreases, and the objective $d(x(\mu), y(\mu))$ of* QD *increases as μ decreases.*

Proof. The fact that $q(y(\mu))$ decreases with μ is a classical result—see Fiacco and McCormick (1968, Theorem 25). That $d(x(\mu), y(\mu))$ increases also follows from this result, since for each $\mu > 0$, $(x(\mu), y(\mu))$ can easily be shown to be the solution of the following barrier function minimization problem associated with QD:

$$
\begin{aligned}
\min \quad & \frac{-d(x,y)}{\mu} - \sum_{j=1}^{n} \ln(x_j) \\
\text{s.t.} \quad & Ax - Qy = b. \qquad \square
\end{aligned}
$$

For a version of Lemma 2.1 in a very general context see den Hertog et al. (1992b).

3. Properties near the central path. In this section we describe proximity measures for the distance between y, an interior point of QP, and a point on the central path $y(\mu)$. We consider three measures. The first is a dual version of a measure introduced for linear programming in Roos and Vial (1992), and extended to quadratic programming in Anstreicher et al. (1993). This measure is

$$
\delta(y,\mu) = \min_{x \mid Ax = b + Qy} \left\| \frac{1}{\mu} Sx - e \right\|. \tag{3.1}
$$

(For $Q = 0$, the linear programming case, the measure $\delta(\cdot,\cdot)$ was introduced in §4 of Roos and Vial (1992).) Note that for any y, the system $Ax = b + Qy$ is feasible since A has independent rows. Let $x(y,\mu)$ denote the solution to the minimization problem in (3.1). From the previous section, it is clear that

$$
y = y(\mu) \iff \delta(y,\mu) = 0 \Rightarrow x(y,\mu) = x(\mu).
$$

The two other measures are associated with the Newton step for $f(\cdot,\mu)$ at y. We use p_y to denote the Newton step in the variables y, and p_s to denote the resulting step in the slack variables s. Letting g and H denote the gradient and Hessian of $f(\cdot,\mu)$ at a given (y,μ), as in (2.2), we have

$$
p_y = -H^{-1} g, \qquad p_s = A^T p_y.
$$

The second criterion is $\| S^{-1}p_s \|$, which may be viewed as the norm of the Newton step, in the slack variables, after a scaling of the constraints that moves the current slack vector s to e, the vector of ones. The third measure is $\sqrt{p_y^T H p_y}$, which we denote by $\| p_y \|_H$. The algorithm of §4, like the algorithms in Anstreicher et al. (1993) and den Hertog et al. (1992a), uses only the third measure, but all three measures enter into the analysis.

LEMMA 3.1. *For given y and μ, $\| S^{-1}p_s \|^2 \leq \| p_y \|_H^2 \leq \delta(y,\mu)^2$.*

Proof. We have

$$p_y^T H p_y = p_y^T \left(\frac{1}{\mu} Q + AS^{-2}A^T \right) p_y \geq p_y^T AS^{-2}A^T p_y = \| S^{-1}p_s \|^2,$$

establishing the first inequality. Next, note that $x = x(y,\mu)$ satisfies $Ax = b + Qy$, so

$$\begin{aligned} g^T H^{-1} g &= \left(\frac{Qy+b}{\mu} - As^{-1} \right)^T \left(\frac{1}{\mu} Q + AS^{-2}A^T \right)^{-1} \left(\frac{Qy+b}{\mu} - As^{-1} \right) \\ &= \left(A \left(\frac{x}{\mu} - s^{-1} \right) \right)^T \left(\frac{1}{\mu} Q + AS^{-2}A^T \right)^{-1} \left(A \left(\frac{x}{\mu} - s^{-1} \right) \right) \\ &= \left(\frac{Sx}{\mu} - e \right)^T S^{-1}A^T \left(\frac{1}{\mu} Q + AS^{-2}A^T \right)^{-1} AS^{-1} \left(\frac{Sx}{\mu} - e \right) \\ &\leq \sigma(P) \left\| \frac{Sx}{\mu} - e \right\|^2 = \sigma(P)\delta(y,\mu)^2, \end{aligned}$$

where $\sigma(P) > 0$ is the maximum eigenvalue of the matrix $P = \bar{A}^T(Q/\mu + \bar{A}\,\bar{A}^T)^{-1}\bar{A}$, and $\bar{A} = AS^{-1}$. We claim that $\sigma(P) \leq 1$. To see this, note that

$$P^2 = \bar{A}^T \left(\frac{1}{\mu} Q + \bar{A}\,\bar{A}^T \right)^{-1} \bar{A}\,\bar{A}^T \left(\frac{1}{\mu} Q + \bar{A}\,\bar{A}^T \right)^{-1} \bar{A},$$

so $\sigma(P^2) = (\sigma(P))^2$ is clearly less than or equal to the maximum eigenvalue of

$$\bar{A}^T \left(\frac{1}{\mu} Q + \bar{A}\,\bar{A}^T \right)^{-1} \left(\frac{1}{\mu} Q + \bar{A}\,\bar{A}^T \right) \left(\frac{1}{\mu} Q + \bar{A}\,\bar{A}^T \right)^{-1} \bar{A} = P.$$

Thus $(\sigma(P))^2 \leq \sigma(P)$, so $\sigma(P) \leq 1$. □

Note that in the case of linear programming ($Q = 0$), the inequalities in Lemma 3.1 all hold with equality, and the three measures are the same. The next lemma will establish quadratic convergence of all three measures in a neighborhood of the central trajectory. This quadratic convergence will be of immediate use in obtaining bounds on $f(\cdot,\mu)$ and $q(\cdot)$, and will also be essential in the analysis of SUMT in §6.

LEMMA 3.2. *Suppose that $\| S^{-1}p_s \| \leq 1$. Let $y^* = y + p_y$, $s^* = s + p_s$, and $S^* = \mathrm{diag}(s^*)$. Let H^* denote the Hessian of $f(\cdot,\mu)$ at y^*, p_{y^*} denote the Newton step for $f(\cdot,\mu)$ at y^*, and $p_{s^*} = A^T p_{y^*}$. Then y^* is strictly feasible in* QP, *and*

$$\| S^{*-1}p_{s^*} \| \leq \| p_{y^*} \|_{H^*} \leq \delta(y^*,\mu) \leq \| S^{-1}p_s \|^2 \leq \| p_y \|_H^2 \leq \delta(y,\mu)^2.$$

Proof. We have $s^* = s + p_s = S(e + S^{-1}p_s)$, so $\| S^{-1}p_s \| < 1$ clearly implies that $s^* > 0$, and y^* is strictly feasible. Next, by definition we have

$$\delta(y^*,\mu) = \min_{x \mid Ax = b + Qy^*} \left\| \frac{1}{\mu} S^* x - e \right\|.$$

It is easy to verify that

$$\frac{1}{\mu}(Qy - Qy^*) = AS^{-1}\left(\frac{Sx}{\mu} - e + S^{-1}p_s\right),$$

and therefore

$$\begin{aligned} Qy^* &= Qy - Ax + \mu AS^{-1}(e - S^{-1}p_s) \\ &= -b + \mu AS^{-1}(e - S^{-1}p_s). \end{aligned}$$

It follows that $x^* = \mu S^{-1}(e - S^{-1}p_s)$ satisfies $Ax^* = b + Qy^*$, so

$$\begin{aligned} \delta(y^*, \mu) &= \min_{x \mid Ax = b + Qy^*} \left\| \frac{1}{\mu} S^* x - e \right\| \\ &\leq \left\| \frac{S^* x^*}{\mu} - e \right\| \\ &= \| (S + P_s)S^{-1}(e - S^{-1}p_s) - e \| \\ &= \| (I + S^{-1}P_s)(e - S^{-1}p_s) - e \| \\ &= \| (S^{-1}p_s)^2 \| \leq \| S^{-1}p_s \|^2, \end{aligned}$$

where $P_s = \operatorname{diag}(p_s)$, and $(S^{-1}p_s)^2$ denotes the vector whose components are the squares of the components of $S^{-1}p_s$. This establishes the middle inequality of the lemma, and the remaining inequalities follow from Lemma 3.1. □

LEMMA 3.3. *If* $\| p_y \|_H < 1$, *then* $f(y, \mu) - f(y(\mu), \mu) \leq \| p_y \|_H^2 / (1 - \| p_y \|_H^2)$.

Proof. The barrier function is convex for fixed μ, so

$$f(y, \mu) - f(y + p_y, \mu) \leq -g^T p_y = \| p_y \|_H^2. \tag{3.2}$$

Now, let $y^0 = y$, and let $y^0, y^1, \ldots$ denote the sequence of points obtained by taking Newton steps, starting at y. Using (3.2) and Lemma 3.2, we obtain

$$\begin{aligned} f(y, \mu) - f(y(\mu), \mu) &= \sum_{i=0}^{\infty} (f(y^i, \mu) - f(y^{i+1}, \mu)) \\ &\leq \sum_{i=0}^{\infty} \| p_y \|_H^{2^{i+1}} \leq \sum_{i=0}^{\infty} \| p_y \|_H^{2(i+1)} \\ &= \frac{\| p_y \|_H^2}{1 - \| p_y \|_H^2}. \end{aligned}$$ □

LEMMA 3.4. *If* $\| p_y \|_H < 1$, *then*

$$| q(y) - q(y(\mu)) | \leq \mu \sqrt{n} \| p_y \|_H (1 + \| p_y \|_H) / (1 - \| p_y \|_H).$$

Proof. The proof is similar to the proof of Lemma 5 in Anstreicher et al. (1993). □

4. The algorithm. Below we give the long step path following algorithm which we analyze in the next section. The scalar quantities τ, ϵ, and θ are parameters of the algorithm that must be specified. The algorithm requires y^0, an initial interior solution of QP (so $s^0 = A^T y^0 - c > 0$), and μ_0, an initial value of the barrier parameter.

ALGORITHM (Long step path following for QP)
Given $k = 0$, $\mu_0 > 0$, y^0, $\epsilon > 0$, $0 < \tau < 1$, $0 < \theta < 1$
Do until $n\mu_k \leq \epsilon$ (Outer iteration)
  $\mu = \mu_{k+1} = (1-\theta)\mu_k$, $y = y^k$
  Do until $\| p_y \|_H \leq \tau$ (Inner iteration)
    $y = y + \alpha p_y$
  End
  $y^{k+1} = y$, $k = k+1$
End

The steplength α used on inner iterations can be any value such that $y + \alpha p_y$ is an interior solution of QP, and the decrease in $f(\cdot, \mu)$ is at least that guaranteed by the steplength $\bar{\alpha}$ in Lemma 5.2. In practice α would be computed using an approximate linesearch of $f(\cdot, \mu)$, and such a linesearch could involve any procedure with up to $O(n^3)$ operations.

For the complexity analysis in §5, we assume that the algorithm is initialized with a value $\mu_0 = 2^{O(L)}$, and uses a termination tolerance $\epsilon = 2^{-O(L)}$, where L is the number of bits in a binary encoding of an instance of QP having integer data. As a consequence of Theorem 5.1, an exact optimal solution to QP can then be computed from the output of the algorithm in $O(n^3)$ operations, using a well-known "rounding" procedure. We also assume that the initial values (y^0, μ_0) satisfy the proximity criterion $\| p_y \|_H \leq \tau$. (This can be weakened to an assumption on $f(y^0, \mu_0) - f(y(\mu_0), \mu_0)$ that depends on the choice of θ; see Anstreicher et al. (1993), §6.4, for details.) See, for example, Monteiro and Adler (1989) for details on initialization and termination procedures for quadratic programming.

5. Convergence analysis. In this section we obtain a complexity bound for the long step path following algorithm above. This bound is obtained in two parts: first, a straightforward bound on the number of outer iterations, and second a more complex bound on the number of inner iterations required on each outer iteration. The argument is a generalization of a proof technique first used for the linear programming case by Gonzaga (1991), and is essentially identical to the argument used in §5 of Anstreicher et al. (1993). We begin with the bound on the number of outer iterations.

THEOREM 5.1. *Let* $E_q = \tau(1+\tau)/(1-\tau)$, $\epsilon = 2^{-O(L)}$. *After* $K = O(L/\theta)$ *outer iterations, the algorithm obtains* $\mu = \mu_K$ *with* $n\mu_K \leq \epsilon$, *and a point* $y = y^K$ *with* $q(y) - z^* \leq (1 + E_q/\sqrt{n})\epsilon$.

Proof. Since $\mu_k = \mu_0(1-\theta)^k$, $n\mu_k \leq \epsilon$ is equivalent to

$$k \ln(1-\theta) \leq \ln\left(\frac{\epsilon}{n\mu_0}\right) = -O(L).$$

But $\ln(1-\theta) \leq -\theta$, so $n\mu_K \leq \epsilon$ for $K = O(L/\theta)$. Let $\mu = \mu_K$, $y = y^K$. Then by Lemma 3.4,

$$\begin{aligned} q(y) - z^* &= q(y(\mu)) - z^* + q(y) - q(y(\mu)) \\ &\leq q(y(\mu)) - d(x(\mu), y(\mu)) + |\, q(y) - q(y(\mu))\,| \\ &\leq n\mu + E_q \mu \sqrt{n} \\ &\leq (1 + E_q/\sqrt{n})\epsilon. \qquad \square \end{aligned}$$

We next consider the issue of bounding the number of inner iterations on one outer iteration. This requires a bound on the guaranteed decrease in $f(\cdot, \mu)$ on inner iterations, provided by the following lemma.

LEMMA 5.2. *Let* $\bar{\alpha} = (1 + \| p_y \|_H)^{-1}$. *Then*

$$f(y, \mu) - f(y + \bar{\alpha} p_y, \mu) \geq \| p_y \|_H - \ln(1 + \| p_y \|_H).$$

Proof. The Taylor series expansion for $f(y + \alpha p_y, \mu)$ is

$$f(y + \alpha p_y, \mu) = f(y, \mu) + \alpha g^T p_y + \frac{1}{2}\alpha^2 p_y^T H p_y + \sum_{k=3}^{\infty} t_k,$$

where t_k denotes the kth term in the expansion. Moreover it is straightforward to compute that t_k, $k \geq 3$ are given by

$$t_k = \frac{(-\alpha)^k}{k} \sum_{j=1}^{n} (S^{-1} p_s)_j^k,$$

where $(S^{-1}p_s)_j^k$ denotes the kth power of the jth component of $S^{-1}p_s$. Using Lemma 3.1, we then obtain

$$| t_k | \leq \frac{\alpha^k}{k} \sum_{j=1}^{n} | (S^{-1} p_s)_j^k | \leq \frac{\alpha^k}{k} \left(\sum_{j=1}^{n} | (S^{-1} p_s)_j |^2 \right)^{\frac{k}{2}} = \frac{\alpha^k}{k} \| S^{-1} p_s \|^k \leq \frac{\alpha^k}{k} \| p_y \|_H^k.$$

Using the fact that $g^T p_y = -\| p_y \|_H^2$, we then have

$$\begin{aligned} f(y + \alpha p_y, \mu) &\leq f(y, \mu) + \left(\frac{1}{2}\alpha^2 - \alpha \right) \| p_y \|_H^2 + \sum_{k=3}^{\infty} \frac{\alpha^k}{k} \| p_y \|_H^k \\ &= f(y, \mu) - \alpha \| p_y \|_H^2 - \ln(1 - \alpha \| p_y \|_H) - \alpha \| p_y \|_H. \end{aligned} \tag{5.1}$$

The right-hand side of (5.1) is minimal at $\alpha = \bar{\alpha} = (1 + \| p_y \|_H)^{-1}$, and substitution of this value into (5.1) completes the proof. □

Note that Lemma 5.2 implies that if $\| p_y \|_H \geq \tau > 0$, then a decrease of $\Delta = \tau - \ln(1 + \tau) > 0$ is definitely possible in $f(\cdot, \mu)$. Using this fact we can now prove our main theorem bounding the number of inner iterations on one outer iteration.

THEOREM 5.3. *Let* $\Delta = \tau - \ln(1 + \tau)$, $E_f = \tau^2/(1 - \tau^2)$, $E_q = \tau(1 + \tau)/(1 - \tau)$. *Then the number of inner iterations required for each outer iteration is at most*

$$N = \frac{1}{\Delta} \left(\frac{\theta}{1 - \theta} \left(\theta n + 2E_q \sqrt{n} \right) + E_f \right).$$

Proof. By Lemma 5.2, if N inner iterations are performed on outer iteration k, then

$$f(y^{k-1}, \mu_k) - f(y^k, \mu_k) \geq N\Delta, \tag{5.2}$$

where both (y^{k-1}, μ_{k-1}) and (y^k, μ_k) satisfy the proximity criterion $\| p_y \|_H \leq \tau$. In addition,

$$\begin{aligned} f(y, \mu_k) &= f(y, \mu_{k-1}) + q(y)/\mu_k - q(y)/\mu_{k-1} \\ &= f(y, \mu_{k-1}) + \frac{\theta}{1 - \theta} \frac{q(y)}{\mu_{k-1}}, \end{aligned} \tag{5.3}$$

for any y. Combining (5.2) and (5.3) obtains

$$f(y^{k-1},\mu_{k-1}) - f(y^k,\mu_{k-1}) + \frac{\theta}{1-\theta}\frac{q(y^{k-1}) - q(y^k)}{\mu_{k-1}} \geq N\Delta. \tag{5.4}$$

But y^{k-1} and y^k both satisfy the proximity criterion, for μ_{k-1} and μ_k, respectively, so

$$f(y^{k-1},\mu_{k-1}) - f(y^k,\mu_{k-1}) \leq E_f, \tag{5.5}$$

by Lemma 3.3. Moreover Lemma 3.4 obtains

$$\begin{aligned} q(y^{k-1}) - q(y^k) &\leq q(y(\mu_{k-1})) + E_q\mu_{k-1}\sqrt{n} - q(y(\mu_k)) + E_q\mu_k\sqrt{n} \\ &\leq n(\mu_{k-1} - \mu_k) + E_q\sqrt{n}(\mu_{k-1} + \mu_k) \\ &\leq \mu_{k-1}(\theta n + 2E_q\sqrt{n}), \end{aligned} \tag{5.6}$$

where the second inequality follows by subtracting and adding $d(x(\mu_{k-1}), y(\mu_{k-1}))$, and then noting that $d(x(\mu_{k-1}), y(\mu_{k-1})) \leq d(x(\mu_k), y(\mu_k))$, by Lemma 2.1. The theorem follows by combining (5.4), (5.5), and (5.6). □

Combining Theorems 5.1 and 5.3, the total number of Newton iterations required for the algorithm is bounded by

$$\frac{1}{\Delta}\left[\frac{\theta n + 2E_q\sqrt{n}}{1-\theta} + \frac{E_f}{\theta}\right] O(L),$$

exactly as in Anstreicher et al. (1993) (where $\tau = 1/2$ gives $\Delta > 1/11$, $E_q = 3/2$, $E_f = 1/3$). It follows that the algorithm requires a total of $O(nL)$ iterations if $\theta = O(1)$, and only $O(\sqrt{n}L)$ iterations if $\theta = O(1/\sqrt{n})$. In addition to a sequence of feasible solutions to QP, the algorithm also can generate a feasible solution to QD whenever the proximity criterion $\| p_y \|_H \leq 1$ is satisfied, as at the end of each sequence of inner iterations. The expense of this computation is an additional pure Newton step and a projection. The next lemma provides a sharp bound on the duality gap for the primal–dual pair generated in this fashion.

LEMMA 5.4. *Assume that* (y,μ) *satisfies* $\| p_y \|_H \leq 1$. *Let* $y^* = y + p_y$, $x^* = x(y^*,\mu)$, *and* $\delta^* = \delta(y^*,\mu)$. *Then* $\delta^* \leq \| p_y \|_H^2$, y^* *is feasible in* QP, (x^*,y^*) *is feasible in* QD, *and* $\mu(n - \delta^*\sqrt{n}) \leq q(y^*) - d(x^*,y^*) \leq \mu(n + \delta^*\sqrt{n})$.

Proof. By Lemmas 3.1 and 3.2, y^* is feasible and $\delta(y^*,\mu) \leq \| p_y \|_H^2$. Also $Ax^* = b + Qy^*$ by the definition of $x(y^*,\mu)$, and $\delta(y^*,\mu) = \| (1/\mu)S^*x^* - e \| \leq 1$ implies that $x^* \geq 0$. Thus (x^*,y^*) is feasible in QD. Finally, by (2.1) we have $q(y^*) - d(x^*,y^*) = (x^*)^T s^*$, and $| x^{*T}s^*/\mu - n | = | e^T(S^*x^*/\mu - e) | \leq \delta(y^*,\mu)\sqrt{n}$. □

6. The SUMT algorithm. Consider a general nonlinear programming problem

$$\begin{aligned} \text{NLP} \quad & \min \quad g_0(y) \\ & \text{s.t.} \quad g_j(y) \geq 0, \quad j = 1,\ldots,n, \end{aligned}$$

where $y \in \Re^m$, $g_0(\cdot)$ is a convex function, and each $g_j(\cdot)$, $j = 1,\ldots,n$ is a concave function. Consider also a logarithmic barrier function, defined on the interior of the feasible region of NLP:

$$P(y,\mu) = g_0(y) - \mu\sum_{j=1}^{n} \ln(g_j(y)),$$

where $\mu > 0$. The classical barrier function method for solving NLP involves approximately minimizing $P(\cdot, \mu)$ for some $\mu > 0$, reducing μ, and repeating the process until some final termination criterion is satisfied. Rudiments of such methods date back to the 1950s; see, for example, Fiacco and McCormick (1968) and den Hertog et al. (1992a) for historical discussions. Fiacco and McCormick (1968), which also considers exterior penalty functions and mixed barrier-penalty algorithms, uses the term sequential unconstrained minimization technique (SUMT) to describe such methods.

Key elements in the design of an SUMT algorithm include the method used to approximately minimize $P(\cdot, \mu)$ for fixed μ, the choice of criterion to terminate this approximate minimization, the scheme for reducing μ, and the final termination criterion. All but the second of these topics are considered in Chapter 8 of Fiacco and McCormick (1968), which formed the basis for the development of the SUMT-3 and SUMT-4 computer codes (Mylander et al. (1971)). (The earlier SUMT-1 and SUMT-2 codes were based on a barrier function $P(\cdot, \cdot)$ using $\mu \sum_{j=1}^{n} 1/g_j(y)$ in place of $-\mu \sum_{j=1}^{n} \ln(g_j(y))$; see Fiacco and McCormick (1963) and McCormick et al. (1965).) For the final ("outer") termination criterion, Fiacco and McCormick (1968) suggests two alternatives:

$$n\mu \leq \epsilon \quad \text{or} \tag{6.1}$$

$$\frac{n\mu}{|\, g_0(y) - n\mu \,|} \leq \epsilon, \tag{6.2}$$

where ϵ is a small positive number, and y is the approximate minimizer of $P(\cdot, \mu)$. The motivation for (6.1) and (6.2) stems from well-known results in Wolfe duality, of which the duality between QP and QD of §2 may be viewed as a special case. In particular, if y exactly minimized $P(\cdot, \mu)$ then $g_0(y) - n\mu$ would be a valid lower bound on z^*, the minimum objective value in NLP. Consequently (6.1) and (6.2) correspond to approximate absolute and relative error bounds for $q(y) - z^*$, respectively. For the approximate minimization of $P(\cdot, \mu)$, it is recommended that a linesearch of $P(\cdot, \mu)$ along the Newton direction be performed, and this process repeated until an approximate minimizer is obtained. Fiacco and McCormick (1968) does not mention an explicit "inner" criterion to terminate this Newton process, but obviously something must have been used in the computer implementations! In fact the SUMT codes offered a choice of three inner termination criteria, two of which are

$$\| \nabla P(y, \mu)^T \| \leq \tau' \quad \text{or} \tag{6.3}$$

$$\nabla P(y, \mu)(\nabla^2 P(y, \mu))^{-1} \nabla P(y, \mu)^T \leq \tau', \tag{6.4}$$

where τ' is a small positive number. The motivation for (6.3), which is also used by Polak (1971), is simply that $\nabla P(y, \mu) = 0$ at $y = y(\mu)$, the exact minimizer of $P(\cdot, \mu)$. The motivation for (6.4) is well known in nonlinear programming: the quantity on the left-hand side of (6.4) would be precisely twice $P(y, \mu) - P(y(\mu), \mu)$ if the quadratic approximation of $P(\cdot, \mu)$ at y was exact. (The third criterion, which we omit, compares the left-hand side of (6.4) with the decrease in $P(\cdot, \mu)$ achieved on the last Newton step.) Finally, for the reduction in μ following an approximate minimization of $P(\cdot, \mu)$, Fiacco and McCormick suggest an update of the form $\mu_{k+1} = \mu_k / \theta'$, where $\theta' > 1$. The SUMT-3 and SUMT-4 codes, described in Mylander et al. (1971), used exactly the algorithmic scheme described above, with (6.2) and (6.4) as the default outer and inner termination criteria, and ϵ, τ', and θ' as user-specified parameters. (The user could also specify an initial feasible point y^0 and an initial value for μ, μ_0.)

Consider now the case of QP, which is NLP with $g_0(y) = q(y) = \frac{1}{2}y^TQy + b^Ty$, and $g_j(y) = a_j^Ty - c_j$, $j \geq 1$. Note that $P(y,\mu) = \mu f(y,\mu)$, so the Newton directions for $P(\cdot,\mu)$ and $f(\cdot,\mu)$ coincide. Also, the outer termination criterion (6.1) is identical to the criterion used to terminate the algorithm of §4. Finally, (6.4) may be written in terms of $f(\cdot,\mu)$ as

$$\| p_y \|_H^2 \leq \frac{\tau'}{\mu}. \tag{6.5}$$

Thus SUMT, using (6.1) and (6.4), is identical to the algorithm of §4 (with $\theta = 1-1/\theta'$) except that (6.5) replaces the condition $\| p_y \|_H \leq \tau$ used to terminate the inner iterations. If the latter is viewed as the "right" criterion, then the problem with (6.5), using $\tau' = O(1)$, is that minimizations of $f(\cdot,\mu)$ are performed too accurately when μ is large, and not accurately enough when μ is small. However, note that by setting $\tau' = \tau^2\mu_K$, where μ_K is the final value of μ to be used by the algorithm, (y,μ) satisfying (6.5) will also always satisfy $\| p_y \|_H \leq \tau$. At first sight this may seem like a hopeless approach, since $\mu_0 = 2^{O(L)}$ and $\mu_K = 2^{-O(L)}$ means that (6.5) could be demanding $\| p_y \|_H \leq 2^{-O(L)}$. However, the quadratic convergence demonstrated in Lemma 3.2 is sufficient to obtain a complexity result for such a scheme, so long as the linesearch procedure used to approximately minimize $f(y + \alpha p_y, \mu)$, or equivalently $P(y + \alpha p_y, \mu)$, on each inner iteration satisfies the following simple assumption.

ASSUMPTION A. *There exist values $0 < \tau < 1$ and $\Delta > 0$, both independent of* QP, *such that* (i) *the algorithm takes a pure Newton step ($\alpha = 1$) whenever $\| p_y \|_H \leq \tau$, and* (ii) *the algorithm achieves $f(y + \alpha p_y, \mu) \leq f(y,\mu) - \Delta$ whenever $\| p_y \|_H > \tau$.*

Assumption A, part (i) is satisfied by any linesearch procedure that employs the common strategy of accepting a pure Newton step if it is feasible, and leads to a decrease in $f(\cdot,\mu)$. In particular, note that (5.1) implies that $\alpha = 1$ gives a decrease of at least $\| p_y \|_H + \| p_y \|_H^2 + \ln(1 - \| p_y \|_H)$, which is positive and increasing for $0 < \| p_y \|_H < 1/2$. We show below, using Lemma 6.1, that Assumption A, part (i) is also satisfied by any procedure which begins by trying $\alpha = 1$, and then performs $O(1)$ "sectioning" steps (for example bisection, or golden section) in an attempt to find a feasible point giving greater decrease in $f(\cdot,\mu)$. The SUMT-3 and SUMT-4 codes used exactly such a procedure, with golden section. Assumption A, part (ii) is then extremely mild; for example $\Delta = \tau - \ln(1+\tau)$ is assured by Lemma 5.2.

LEMMA 6.1. *Suppose that $\alpha \geq 0$, $\alpha \neq 1$. Let $r = |1 - \alpha|$. Then $f(y + p_y, \mu) < f(y + \alpha p_y, \mu)$ whenever $0 < \| p_y \|_H \leq .5r^2/(1+r)^3$.*

Proof. Proceeding as in the proof of Lemma 5.2, we conclude that for any $0 \leq \alpha < 1/\| p_y \|_H$,

$$f(y + \alpha p_y, \mu) \leq f(y,\mu) + \left(\frac{1}{2}\alpha^2 - \alpha\right)\| p_y \|_H^2 + \sum_{k=3}^{\infty}\frac{\alpha^k}{k}\| p_y \|_H^k,$$

$$f(y + \alpha p_y, \mu) \geq f(y,\mu) + \left(\frac{1}{2}\alpha^2 - \alpha\right)\| p_y \|_H^2 - \sum_{k=3}^{\infty}\frac{\alpha^k}{k}\| p_y \|_H^k,$$

where $\sum_{k=3}^{\infty}(\alpha^k/k)\| p_y \|_H^k = -\ln(1 - \alpha\| p_y \|_H) - (\alpha^2/2)\| p_y \|_H^2 - \alpha\| p_y \|_H$. Thus for $\| p_y \|_H < 1$, and $0 \leq \alpha < 1/\| p_y \|_H$,

$$f(y + p_y, \mu) \leq f(y,\mu) - (\| p_y \|_H + \| p_y \|_H^2) - \ln(1 - \| p_y \|_H),$$
$$f(y + \alpha p_y, \mu) \geq f(y,\mu) + \alpha^2\| p_y \|_H^2 + \alpha(\| p_y \|_H - \| p_y \|_H^2) + \ln(1 - \alpha\| p_y \|_H).$$

Using the standard estimate $\ln(1-\beta) \geq -\beta - .5\beta^2/(1-\beta)$, we next obtain

$$f(y+p_y,\mu) \leq f(y,\mu) - \| p_y \|_H^2 + \frac{\| p_y \|_H^2}{2(1-\| p_y \|_H)}, \tag{6.6}$$

$$f(y+\alpha p_y,\mu) \geq f(y,\mu) + \| p_y \|_H^2(\alpha^2-\alpha) - \frac{\alpha^2 \| p_y \|_H^2}{2(1-\alpha\| p_y \|_H)}. \tag{6.7}$$

From (6.6) and (6.7), to show $f(y+p_y,\mu) < f(y+\alpha p_y,\mu)$ it will suffice to have

$$-1 + \frac{1}{2(1-\| p_y \|_H)} < (\alpha^2-\alpha) - \frac{\alpha^2}{2(1-\alpha\| p_y \|_H)}. \tag{6.8}$$

To simplify notation, let $\tau = \| p_y \|_H < 1$. Then (6.8) is equivalent to each of the following inequalities:

$$\begin{aligned}
2(\alpha^2-\alpha+1) &> \frac{(1-\alpha\tau)+\alpha^2(1-\tau)}{(1-\tau)(1-\alpha\tau)}, \\
(1-\alpha)^2 &> \frac{(1-\alpha\tau)+\alpha^2(1-\tau)-(1+\alpha^2)(1-\tau)(1-\alpha\tau)}{(1-\tau)(1-\alpha\tau)}, \\
(1-\alpha)^2 &> \frac{\tau[(1-\alpha\tau)+\alpha^3(1-\tau)]}{(1-\tau)(1-\alpha\tau)},
\end{aligned}$$

where the last follows from simple algebra. Thus (6.8) holds if and only if

$$\tau < \frac{(1-\alpha)^2}{\left(\frac{1}{1-\tau}\right)+\left(\frac{\alpha^3}{1-\alpha\tau}\right)}. \tag{6.9}$$

Now let $r = |\,1-\alpha\,|$, and assume that $\tau \leq r^2/2(1+r)^3$. To prove the lemma, from (6.9) it will suffice to show that

$$2(1+r)^3 > \frac{1}{1-\tau} + \frac{\alpha^3}{1-\alpha\tau}. \tag{6.10}$$

However, it is certainly the case that

$$\frac{1}{1-\tau} + \frac{\alpha^3}{1-\alpha\tau} \leq \frac{1}{1-(1+r)\tau} + \frac{(1+r)^3}{1-(1+r)\tau} = \frac{1+(1+r)^3}{1-(1+r)\tau} \leq \frac{1+(1+r)^3}{1-\frac{r^2}{2(1+r)^2}}.$$

Hence to show (6.10) it will suffice to show that

$$2(1+r)^3 > \frac{1+(1+r)^3}{1-\frac{r^2}{2(1+r)^2}},$$

which is equivalent to each of the following inequalities:

$$\begin{aligned}
2(1+r)^3 &> \frac{2(1+r)^2(1+(1+r)^3)}{2(1+r)^2-r^2}, \\
(1+r)(2(1+r)^2-r^2) &> 1+(1+r)^3, \\
(1+r)((1+r)^2-r^2) &> 1,
\end{aligned}$$

and the last certainly holds since $r > 0$. □

COROLLARY 6.2. *Suppose that on inner iterations, the algorithm first tries $\alpha = 1$, and then uses trial steplengths $\alpha \neq 1$ each of which satisfies $|1 - \alpha| \geq r > 0$, where $r = \Omega(1)$. Then there is a $\tau > 0$, $\tau = \Omega(1)$ so that Assumption* A, *part* (i) *is satisfied.*

Proof. Assume first that $r < 1$, and let $\tau = .5r^2/(1+r)^3$. From Lemma 6.1, if $\| p_y \|_H \leq \tau$, then the steplengths $\alpha = 1 - r$ and $\alpha = 1 + r$ both produce $f(y + p_y, \mu) < f(y + \alpha p_y, \mu)$. It follows from the convexity of $f(\cdot, \mu)$ that no steplength $\alpha \neq 1$ tried by the algorithm will produce greater descent than $\alpha = 1$, so part (i) of Assumption A is satisfied. Next assume that $r \geq 1$. In this case any steplengths $\alpha \neq 1$ are of the form $\alpha \geq 2$. Letting $\tau = 1/16$, the value of $.5r^2/(1+r)^3$ for $r = 1$, Lemma 6.1 implies that $f(y + p_y, \mu) < f(y + 2p_y, \mu)$, and part (i) of Assumption A is satisfied. □

The conditions of Corollary 6.2 are immediately satisfied when the algorithm uses a "sectioning" procedure, with a fixed number of steps, for the linesearch, starting at $\alpha = 1$. For example, consider a procedure that takes 10 bisection steps starting with $\alpha = 1$. (If $\alpha = 1$ is infeasible, or $\nabla f(y + p_y)p_y > 0$ then the bisection procedure would consider steplengths $\alpha \in [0, 1]$. If $\nabla f(y + p_y)p_y < 0$, then the procedure would begin by trying "outward bisection" steplengths of $\alpha = 2, 4, 8, \ldots$ until a steplength $\bar{\alpha}$ is found which is either infeasible, or produces $\nabla f(y + \bar{\alpha}p_y)p_y > 0$. The remaining usual bisection steps would then be performed on the interval $\alpha \in [\bar{\alpha}/2, \bar{\alpha}]$.) Clearly such a procedure satisfies the assumptions of Corollary 6.2, with $r = 2^{-10}$. As mentioned earlier, the SUMT-3 and SUMT-4 codes used a similar sectioning procedure, based on a fixed number of golden section steps, starting at $\alpha = 1$.

THEOREM 6.3. *Suppose that the* SUMT *algorithm is applied to* QP, *using* (6.1) *and* (6.4) *as the outer and inner termination criteria, respectively. Let τ and Δ be as in Assumption* A, *$E_f = \tau^2/(1-\tau^2)$, $E_q = \tau(1+\tau)/(1-\tau)$. Assume that $\mu_0 = 2^{O(L)}$, and that $y = y^0$ satisfies $\| p_y \|_H \leq \tau$. Let $\epsilon = 2^{-O(L)}$, $\tau' = \tau^2\mu_K = 2^{-O(L)}$, $\theta = 1 - 1/\theta'$. Then the total number of inner iterations required by the algorithm is at most*

$$\left[\frac{1}{\Delta}\left(\frac{\theta n + 2E_q\sqrt{n}}{1-\theta} + \frac{E_f}{\theta}\right) + \frac{\log L}{\theta}\right] O(L).$$

Proof. The number of outer iterations required remains $O(L/\theta)$, exactly as in Theorem 5.1. Furthermore, by the choice of τ' and (6.5), we will have $\| p_y \|_H \leq \tau$ at the end of each sequence of inner iterations. Thus the number of inner iterations required to obtain $\| p_y \|_H \leq \tau$ on each outer iteration may be bounded exactly as in the proof of Theorem 5.3. It remains to bound the number of additional inner iterations required to go from $\| p_y \|_H \leq \tau$ to $\| p_y \|_H \leq \tau\sqrt{\mu_K/\mu}$, as demanded by (6.5). By Assumption A, part (i), the algorithm takes pure Newton steps ($\alpha = 1$) once $\| p_y \|_H \leq \tau$, so Lemma 3.2 implies that after i such steps we obtain a point y with $\| p_y \|_H \leq \tau^{2^i}$. Since $\mu_0 = 2^{O(L)}$, $\mu_K = 2^{-O(L)}$, (6.5) will certainly hold once $\tau^{2^i} \leq 2^{-O(L)}$, which requires $i = O(\log L)$ steps. □

By Theorem 6.3, the total number of inner iterations for SUMT, using (6.1) and (6.4), is $O(nL + L\log L)$ if $\theta = O(1)$, and $O(\sqrt{n}L\log L)$ if $\theta = O(1/\sqrt{n})$. Note also that the initial centering assumption $\| p_y \|_H \leq \tau$, while convenient for the statement of the theorem, is stronger than necessary; for example a weaker assumption on $f(y^0, \mu_0) - f(y(\mu_0), \mu_0)$ suffices (see Anstreicher et al. (1993), §6.4).

As noted above, Assumption A, part (i) is satisfied by the golden section linesearch procedure used in the SUMT-3 and SUMT-4 codes. Although in any case Assumption A, part (ii) is very mild, it is worth noting that with $\theta = O(1/\sqrt{n})$ the linesearch procedure of the SUMT code definitely satisfies (ii) as well. As mentioned earlier,

when $0 < \tau < \| p_y \|_H \leq 1/2$, the initial pure Newton step will certainly obtain a reduction of $\Delta = \tau + \tau^2 + \ln(1-\tau) > 0$ in $f(\cdot, \mu)$. Thus to guarantee a reduction of Δ whenever $\| p_y \|_H > \tau$, it suffices to prevent $\| p_y \|_H > 1/2$. This is easily accomplished by an appropriate choice of θ. In particular, note that from the proof of Theorem 5.3, the quantity $[\theta/(1-\theta)](\theta n + 2E_q\sqrt{n}) + E_f$ is an upper bound on the possible reduction in $f(\cdot, \mu)$, while Lemma 5.2 demonstrates that a reduction of $.5 - \ln(1.5) > .0945$ is possible if $\| p_y \|_H > 1/2$. Since E_f and E_q both go to zero with τ, for small τ and large n, setting $n\theta^2 \leq .0945$, or $\theta \leq .3/\sqrt{n}$, will suffice to ensure that $\| p_y \|_H \leq 1/2$. Of course this produces essentially a "short step" rather than a "long step" algorithm, but our concern is to show that the complexity of $O(\sqrt{n}L \log L)$ iterations can be made absolutely rigorous.

As our final topic, we consider the SUMT algorithm using (6.4) as the inner termination criterion, but with (6.2) replacing (6.1) as the outer termination criterion. As mentioned earlier, the use of (6.2) represents a relative, rather than absolute, error tolerance, and is implemented as the default option in the SUMT-3 and SUMT-4 codes. Naturally the use of a relative tolerance is problematic if $z^* = 0$. So long as $z^* \neq 0$, however, we will show that the algorithm using (6.2) can be analyzed with simple modifications to the analysis based on (6.1) above. Let $\epsilon = 2^{-O(L)}$ be the desired absolute error tolerance in (6.1), as above, but suppose that the algorithm is implemented using (6.2) with tolerance ϵ', so that the actual outer termination criterion is

$$\frac{n\mu}{|\, q(y) - n\mu \,|} \leq \epsilon'. \tag{6.11}$$

THEOREM 6.4. *Suppose that the* SUMT *algorithm is applied to* QP, *with* $z^* \neq 0$, *using* (6.11) *and* (6.4) *as the outer and inner termination criteria, respectively. Then* $\epsilon' = 2^{-O(L)}$ *may be chosen so that termination due to* (6.11) *implies that* (6.1) *is also satisfied, with* $\epsilon = 2^{-O(L)}$. *Moreover there is a* $K = O(L/\theta)$, *with* $\mu_K = 2^{-O(L)}$, *so that termination occurs in at most* K *outer iterations, and the total number of inner iterations is bounded exactly as in Theorem* 6.3.

Proof. Let $q_{\max} > 0$ denote any upper bound on $|\, q(y) \,|$ over all feasible y; theoretically we may take $q_{\max} = 2^{O(L)}$ since the feasible region is assumed bounded. Similarly, let $z_{\min} > 0$ be any lower bound on $|\, z^* \,|$, where theoretically $z_{\min} \geq 2^{-O(L)}$ under the assumption that $z^* \neq 0$. Since $\mu_0 = 2^{O(L)}$, it is certainly the case that $|\, q(y) - n\mu \,| \leq M \equiv q_{\max} + n\mu_0 \leq 2^{O(L)}$ throughout the algorithm. Hence setting $\epsilon' = \epsilon/M = 2^{-O(L)}$ ensures that termination due to (6.11) implies that (6.1) is satisfied as well.

Next we consider the maximum number of outer steps which could be required before (6.11) is satisfied. Assume first that $z^* > 0$. Then if $n\mu_k \leq z_{\min}/2 = 2^{-O(L)}$, we will certainly have $|\, q(y^k) - n\mu_k \,| \geq z_{\min}/2$, since otherwise $q(y^k) < z_{\min} \leq z^*$, a contradiction. If in addition $n\mu_k \leq \epsilon' z_{\min}/2 = 2^{-O(L)}$, (6.11) is clearly then satisfied as well. Thus the algorithm must terminate within K outer iterations, where μ_K satisfies $n\mu_K \leq \epsilon' z_{\min}/2$. It follows that $K = O(L/\theta)$, exactly as in Theorem 5.1. Next assume that $z^* < 0$. As in the proof of Theorem 5.1, $n\mu_k \leq .5 z_{\min}/(1 + E_q)$ would imply that $z^* \geq q(y^k) - z_{\min}/2$. Then $q(y_k) \leq -z_{\min}/2$, since $z^* \leq -z_{\min}$, so $|\, q(y^k) - n\mu_k \,| \geq z_{\min}/2$. The remainder of the argument is identical to the case $z^* > 0$.

The value μ_K is used to set τ' in (6.4), exactly as before, even though the algorithm may actually terminate in fewer than K outer iterations. Since $\mu_K = 2^{-O(L)}$,

the complexity analysis of Theorem 6.3 remains valid, and obtains the same bound on the number of inner iterations. □

Acknowledgments. The author would like to thank Kees Roos, Dick den Hertog, and Tamas Terlaky for introducing him to long step path following analysis, and Garth McCormick for explaining the details of SUMT. The paper has also benefitted from the extensive comments of two anonymous referees.

REFERENCES

K. M. ANSTREICHER, D. DEN HERTOG, C. ROOS, AND T. TERLAKY (1993), *A long-step barrier method for convex quadratic programming*, Algorithmica 10, pp. 365–382.

D. DEN HERTOG, C. ROOS, AND T. TERLAKY (1992a), *On the classical logarithmic barrier method for a class of smooth convex programming problems*, JOTA 73, pp. 1–25.

——— (1992b), *On the monotonicity of the dual objective along barrier paths*, COAL Bulletin 20, pp. 2–7.

A. V. FIACCO AND G. P. MCCORMICK (1963), *Programming under nonlinear constraints by unconstrained minimization: a primal-dual method*, Technical Paper RAC-TP-96, Research Analysis Corporation, McLean, VA.

——— (1968), *Nonlinear Programming, Sequential Unconstrained Minimization Techniques*, John Wiley, New York; reprinted as *Classics in Applied Mathematics Vol.* 4, Society for Industial and Applied Mathematics, Philadelphia, 1990.

P. E. GILL, W. MURRAY, M. A. SAUNDERS, J. A. TOMLIN, AND M. H. WRIGHT (1986), *On projected Newton barrier methods for linear programming and an equivalence with Karmarkar's projective method*, Math. Programming 36, pp. 183–209.

C. C. GONZAGA (1991), *Large-step path-following methods for linear programming, Part* I: *Barrier function method*, SIAM J. Optim. 1, pp. 268–279.

N. KARMARKAR (1984), *A new polynomial-time algorithm for linear programming*, Combinatorica 4, pp. 373–395.

G. P. MCCORMICK, W. C. MYLANDER, AND A. V. FIACCO (1965), *Computer program implementing the sequential unconstrained minimization technique for nonlinear programming*, Technical Paper RAC-TP-151, Research Analysis Corporation, McLean, VA.

R. C. MONTEIRO AND I. ADLER (1989), *Interior path following primal-dual algorithms. Part* II: *Convex quadratic programming*, Math. Programming 44, pp. 43–66.

W. C. MYLANDER, R. L. HOLMES, AND G. P. MCCORMICK (1971), *A guide to* SUMT-*Version* 4: *The computer program implementing the sequential unconstrained minimization technique for nonlinear programming*, Research Paper RAC-P-63, Research Analysis Corporation, McLean, VA.

E. POLAK (1971), *Computational Methods in Optimization*, Academic Press, New York.

J. RENEGAR (1988), *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Programming 40, pp. 59–93.

C. ROOS AND J-PH. VIAL (1990), *Long steps with the logarithmic penalty barrier function in linear programming*, in Economic Decision Making: Games, Economics, and Optimization, J. Gabszewicz, J.-F. Richard, and L. Wolsey, eds., Elsevier Science, Amsterdam, pp. 433–441.

——— (1992), *A polynomial method of approximate centers for linear programming*, Math. Programming 54, pp. 295–305.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 47–56, February 1996
© 1996 Society for Industrial and Applied Mathematics
004

THE MEHROTRA PREDICTOR-CORRECTOR INTERIOR-POINT METHOD AS A PERTURBED COMPOSITE NEWTON METHOD*

R. TAPIA†, Y. ZHANG‡, M. SALTZMAN§, AND A. WEISER¶

Abstract. It is well known that the celebrated Kojima–Mizuno–Yoshise primal-dual interior-point method for linear programming can be viewed as a damped perturbed Newton's method. Recently, Mehrotra suggested a predictor-corrector variant of this method. It is currently the interior-point method of choice for linear programming. The simplified Newton method, at the expense of fast convergence, reduces the work required by Newton's method by reusing the initial Jacobian matrix. The composite Newton method attempts to balance the trade-off between expense and fast convergence by composing one Newton step with one simplified Newton step. In this work we demonstrate that if the Newton component in the Kojima–Mizuno–Yoshise primal-dual method is replaced with a composite Newton component, then the resulting method is the Mehrotra predictor-corrector method.

Key words. linear programming, primal-dual interior-point algorithms, Newton's method

AMS subject classifications. 65K05, 90C05

1. Introduction.

In §1.1 we review the composite Newton method, in §1.2 we recall the primal-dual interior-point method, in §1.3 we present the Mehrotra predictor-corrector interior-point method, and in §1.4 we present our perturbed composite Newton interior-point method. Section 2 contains equivalence results between the Mehrotra predictor-corrector method and the perturbed level-1 composite Newton method. Since the level-1 composite Newton method is known to be cubically convergent, in §3 we study the cubic convergence aspect of the Mehrotra predictor-corrector interior-point method via our equivalence result. It is interesting to learn that the interior-point feature of the method, i.e., the step is damped so that iterates remain positive, precludes the standard proof of cubic convergence of the method. However, for nondegenerate problems it is possible to retain quadratic convergence. Recall that Zhang, Tapia, and Dennis [15] demonstrated that the primal-dual interior-point method can attain quadratic convergence for nondegenerate problems. We then prove that by choosing steplength one in a neighborhood of the solution, cubic convergence can be attained by the predictor-corrector interior-point method for nondegenerate problems.

Clearly an optimal implementation of the composite Newton interior-point method would allow m (the number of simplified Newton steps) to vary at each Newton step. This issue is not the subject of the current work, but probably merits further study. Finally, in §4 we give some concluding remarks.

* Received by the editors October 25, 1991; accepted for publication (in revised form) September 16, 1994. This research was supported in part by National Science Foundation Cooperative Agreement CCR-8809615 and was initiated early in 1990 while the second and third authors were at Rice University as visiting members of the Center for Research in Parallel Computation.

† Department of Computational and Applied Mathematics and Center for Research on Parallel Computation, Rice University, Houston, TX 77251-1892 (tapia@masc8.rice.edu). This author was supported in part by Air Force Office of Scientific Research grant 89-0363, Department of Energy grant DEFG05-86ER25017, and Army Research Office grant 9DAAL03-90-G-0093.

‡ Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD 21228. This author was supported in part by National Science Foundation grant DMS-9102761 and Department of Energy grant DE-FG05-91ER25100.

§ Department of Mathematical Sciences, Clemson University, Clemson, SC 29634.

¶ Center for Research on Parallel Computation, Rice University, Houston, TX 77251-1892.

1.1. The composite Newton method. Consider the nonlinear equation

$$F(x) = 0, \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$. By the *damped Newton method* for problem (1.1) we mean the iterative process

$$\begin{array}{lll} \text{solve} & F'(x_k)(\Delta x) = -F(x_k) & \text{for } \Delta x, \\ \text{set} & x_{k+1} = x_k + \alpha_k \Delta x\,, & k = 0, 1, \ldots\,. \end{array} \tag{1.2}$$

The flexibility of being able to choose α_k less than one is important from global convergence considerations. When the choice of steplength is $\alpha_k = 1$ we drop the qualifier *damped.*

Under standard assumptions Newton's method is known to give Q-quadratic convergence. Not counting the work required to evaluate the function F or its Jacobian, the algebra required per iteration is $O(n^3)$, since the dominant task is the factorizing of the $n \times n$ Jacobian matrix $F'(x_k)$. For large n this can be a very serious concern.

A particularly obvious technique for reducing the amount of algebra needed at each iteration is given by the *damped simplified Newton method*

$$\begin{array}{lll} \text{solve} & F'(x_0)(\Delta x) = -F(x_k) & \text{for } \Delta x, \\ \text{set} & x_{k+1} = x_k + \alpha_k \Delta x\,, & k = 0, 1, \ldots\,. \end{array} \tag{1.3}$$

The simplified Newton method requires an initial factorization of $F'(x_0)$ and then a solve at each iteration; hence it requires only $O(n^2)$ algebra per iteration. However, it gives only Q-linear convergence and it is not at all clear in what cases it should be preferred to Newton's method, since the slow convergence might force a prohibitive number of iterations.

In an effort to cover the middle ground between the extremes of Newton and simplified Newton it is very natural to consider the variant of Newton's method which takes m simplified Newton steps between every two Newton steps. By the *damped (level-m) composite Newton method* we mean the iterative procedure

$$\begin{array}{lll} \text{solve} & F'(x_k)(\Delta x_i) = -F(x_k + \Delta x_0 + \cdots + \Delta x_{i-1}) & \text{for } \Delta x_i\,, i = 0, \ldots, m, \\ \text{set} & x_{k+1} = x_k + \alpha_k(\Delta x_0 + \Delta x_1 + \cdots + x_m)\,, & k = 0, 1, \ldots\,. \end{array} \tag{1.4}$$

Of course it is possible to introduce a different steplength control $\alpha_{k,i}$ for each correction Δx_i, $i = 0, \ldots, m$; however we have no need to consider such flexibility.

It is reasonably well known that, under the standard Newton method assumptions, the level-m composite Newton method has a Q-convergence rate of $m+2$. A proof can be found in Chapter 10 of Ortega and Rheinboldt [10]. The damped level-1 composite Newton method where one Newton step is composed with one simplified Newton step is of particular interest to us. It can be written

$$\begin{array}{lll} \text{solve} & F'(x_k)(\Delta x_N) = -F(x_k) & \text{for } \Delta x_N, \\ \text{solve} & F'(x_k)(\Delta x_S) = -F(x_k + \Delta x_N) & \text{for } \Delta x_S, \\ \text{set} & x_{k+1} = x_k + \alpha_k(\Delta x_N + \Delta x_S)\,, & k = 0, 1, \ldots\,. \end{array} \tag{1.5}$$

Ortega and Rheinboldt [10] credit the cubic convergence of the level-1 composite Newton method to Traub [12]. However, the notion of composing Newton steps with simplified Newton steps is much older and a part of the folklore of Newton's method. It is generally felt by practitioners that the formulation of composite Newton steps is

of value when n is large and the function F can be evaluated cheaply; this is clearly the situation for the primal-dual interior-point method for linear programming described in §1.2.

Observe that each level-m composite Newton iterate can be viewed as a major iterate and is the result of $m+1$ inner iterations. The average amount of algebra per inner iteration is $O((n^3+mn^2)/(m+1))$ and is $O(n^2)$ for large m. The average convergence rate for the inner iterates is the $(m+1)$st root of $m+2$ and behaves like 1 for large m. It is no surprise then that for large m the level-m composite Newton method behaves like the simplified Newton method. It follows that an optimal implementation of composite Newton would not only vary m at each Newton step but would keep m relatively small.

1.2. The primal-dual interior-point method. Consider a linear program in the standard form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0, \end{array} \tag{1.6}$$

where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$ $(m < n)$ and A has full rank m.

The first-order optimality conditions for the linear program (1.6) can be written

$$F(x,y,\lambda) \equiv \begin{pmatrix} Ax-b \\ A^T\lambda + y - c \\ XYe \end{pmatrix} = 0\,, \quad (x,y) \geq 0, \tag{1.7}$$

where $y \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$ are dual variables, $X = \operatorname{diag}(x)$, $Y = \operatorname{diag}(y)$, and $e^T = (1,\ldots,1) \in \mathbb{R}^n$.

The point (x,y,λ) is said to be feasible for problem (1.7) if $Ax = b$, $A^T\lambda + y - c = 0$, and $(x,y) \geq 0$. A feasible point (x,y,λ) is strictly feasible if $(x,y) > 0$. We tacitly assume that strictly feasible points exist.

It is now well understood how the primal-dual interior-point method introduced by Kojima, Mizuno, and Yoshise [3] can be stated in the framework of a damped and perturbed Newton's method applied to problem (1.7). In presenting this algorithmic framework we write $z = (x,y,\lambda)$, $\Delta z = (\Delta x, \Delta y, \Delta\lambda)$, $\Delta X = \operatorname{diag}(\Delta x)$, and $\Delta Y = \operatorname{diag}(\Delta y)$. We also let $\min(u)$ denote the smallest component of the vector u and $\hat{e}$ denote the vector $(0,\ldots,0,1,\ldots,1)^T$ where the number of zeros is $n+m$ and the number of ones is n.

ALGORITHM 1 (Primal-dual interior-point method)

Given $z_0 = (x_0, y_0, \lambda_0)$ with $(x_0, y_0) > 0$, for $k = 0, 1, \ldots$, do

(1.8)

(1) Solve $F'(z_k)(\Delta z) = -F(z_k)$ for Δz_N.

(2) Choose $\mu_k > 0$ and

solve $F'(z_k)(\Delta z) = \mu_k \hat{e}$ for Δz_c.

(3) Set $\Delta z = \Delta z_N + \Delta z_c$.

(4) Choose $\tau_k \in (0,1)$ and set $\alpha_k = \min(1, \tau_k \hat{\alpha}_k)$ where

(1.9)
$$\hat{\alpha}_k = \min\left(\frac{-1}{\min(X_k^{-1}\Delta x)}, \frac{-1}{\min(Y_k^{-1}\Delta y)}\right).$$

(5) Set $z_{k+1} = z_k + \alpha_k \Delta z$.

Actually in most implementations the formula (1.9) for α_k is further broken down and one steplength is used to update the x-variable and another is used to update the y-variable and the λ-variable. While this distinction is of value in practice, it is not an issue in the present work and consequently will be ignored.

Recently, under mild assumptions, Zhang, Tapia, and Dennis [15] demonstrated that for nondegenerate and degenerate problems Q-superlinear convergence could be attained by Algorithm 1 by merely letting $\sigma_k \to 0$ and $\tau_k \to 1$, where σ_k is defined by $\mu_k = \sigma_k x_k^T y_k/n$. Moreover, for nondegenerate problems Q-quadratic convergence could be attained by letting $\sigma_k = O(x_k^T y_k)$ and $\tau_k = 1 + O(x_k^T y_k)$. Zhang and Tapia [14] showed that these results held under weaker assumptions.

The Newton step Δz_N defined in step (1) can very likely point toward the boundary of the positive orthant, necessitating a very small choice for the steplength α. The major role of the centering step Δz_c defined in step (2) is to remedy this situation. Hence it seems quite reasonable that the choice for the centering parameter μ_k should also be a function of the Newton step Δz_N. This is particularly true in delicate applications. For example, recently there was considerable speculation as to whether an instance of Algorithm 1 could have both polynomial complexity and superlinear convergence. In their original paper Kojima, Mizuno, and Yoshise [3] presented choices for τ_k and μ_k leading to polynomial complexity. Zhang, Tapia, and Dennis [15] presented conditions on τ_k and μ_k that guaranteed superlinear convergence. Zhang and Tapia [13] settled this concern by demonstrating the existence of choices for τ_k and μ_k that guaranteed both polynomial complexity and superlinear convergence. Ji et al. [11] extended this result to linear complementarity problems. In both these applications the choice of μ_k depended strongly on the Newton step Δz_N. Hence, the centering step Δz_c had to be calculated as described in step (2) above. However, in less delicate applications where μ_k depends only on z_k, steps (1) and (2) in Algorithm 1 can be combined and the combined step $\Delta z = \Delta z_N + \Delta z_c$ can be obtained as the solution of

$$F'(z_k)(\Delta z) = -F(z_k) + \mu_k \hat{e}_k \,. \tag{1.10}$$

In this way the backsolve required by step (2) can be saved. This is the more common presentation of the Kojima–Mizuno–Yoshise algorithm and is fine for restricted applications.

The sense in which the primal-dual interior-point method can be viewed as damped perturbed Newton should be clear. The qualifier damped speaks to the steplength $\alpha_k \leq 1$ in step (5). The qualifier perturbed speaks to the fact that the step Δz consists of the Newton step Δz_N perturbed by the centering step Δz_c; see step (3).

1.3. The predictor-corrector interior-point method. Mizuno, Todd, and Ye [8] suggested and studied an algorithm which they labeled a predictor-corrector algorithm. In their algorithm the predictor step is a damped Newton step for problem (1.7), producing a new strictly feasible iterate. The subsequent corrector step is a centered Newton step. In this corrector step, the choice of μ, the centering parameter, is based on the predictor step. Both the predictor and the corrector steps require essentially the same amount of work, namely, the evaluation and factorization of the Jacobian matrix.

Mehrotra [6] later presented the following variant of Algorithm 1, which he also referred to as a predictor-corrector method. A common feature in these two predictor-corrector approaches is that the value of the centering parameter in the corrector step depends on the predictor step. However, unlike Mizuno, Todd, and Ye's corrector step,

Mehrotra's corrector step does not evaluate a fresh Jacobian matrix. Instead, it reuses the Jacobian matrix used by the predictor step. Recall that $\hat{e} = (0, \ldots, 0, 1, \ldots, 1)^T$.

ALGORITHM 2 (Predictor-corrector interior-point method)
Given $z_0 = (x_0, y_0, \lambda_0)$ with $(x_0, y_0) > 0$, for $k = 0, 1, \ldots$ do

(1) Solve $F'(z_k)(\Delta z) = -F(z_k)$ for Δz_p.

(2) Solve $F'(z_k)(\Delta z) = -\begin{pmatrix} 0 \\ 0 \\ \Delta X_p \Delta y_p \end{pmatrix}$ for Δz_M.

(3) Choose $\mu_k > 0$ and
solve $F'(z_k)(\Delta z) = \mu_k \hat{e}$ for Δz_c.

(4) Set $\Delta z = \Delta z_p + \Delta z_M + \Delta z_c$.

(1.11) (5) Choose $\tau_k \in (0, 1)$ and set $\alpha_k = \min(1, \tau_k \hat{\alpha}_k)$ where
$$\hat{\alpha}_k = \min\left(\frac{-1}{\min X_k^{-1} \Delta x}, \frac{-1}{\min(Y_k^{-1} \Delta y)}\right).$$

(6) Set $z_{k+1} = z_k + \alpha_k \Delta z$.

While in this section we are not concerned with the specific choice of the initial iterate z_0 or the various algorithmic parameters, we emphasize that Mehrotra suggested choices that allowed him to obtain very impressive numerical results.

1.4. The perturbed composite Newton interior-point method. In this section we present our perturbed composite Newton interior-point method for problem (1.7). Recall that $\hat{e} = (0, \ldots, 0, 1, \ldots, 1)^T$. Our idea is to replace the Newton component in the primal-dual interior-point algorithm with a composite Newton component.

ALGORITHM 3 (Level-m perturbed composite Newton interior-point method)
Given $z_0 = (x_0, y_0, \lambda_0)$ with $(x_0, y_0) > 0$ for $k = 0, 1, \ldots,$ do

(1) Solve $F'(z_k)(\Delta z) = -F(z_k)$ for Δz_0.

(2) For $i = 1, \ldots, m$ do

(1.12) Solve $F'(z_k)(\Delta z) = -F\left(z_k + \sum_{j=0}^{i-1} \Delta z_j\right)$ for Δz_i.

(3) Choose $\mu_k > 0$ and solve $F'(z_k)(\Delta z) = \mu_k \hat{e}$ for Δz_c.

(4) Set $\Delta z = \sum_{i=0}^{m} \Delta z_i + \Delta z_c$.

(5) Choose $\tau_k \in (0, 1)$ and set $\alpha_k = \min(1, \tau_k \hat{\alpha}_k)$ where
$$\hat{\alpha}_k = \min\left(\frac{-1}{\min(X_k^{-1} \Delta x)}, \frac{-1}{\min(Y_k^{-1} \Delta y)}\right).$$

(6) Set $z_{k+1} = z_k + \alpha_k \Delta z$.

2. Predictor-corrector as perturbed composite Newton method. We say that two algorithms are equivalent if given a current iterate they produce the same subsequent iterate for the same choice of common algorithmic parameters.

THEOREM 2.1. *The predictor-corrector interior-point method (Algorithm 2) is equivalent to the level-1 perturbed composite Newton interior-point method (Algorithm 3).*

Proof. Let $z = (x, y, \lambda)$ be the current iterate and let $\Delta z_p = (\Delta x_p, \Delta y_p, \Delta \lambda_p)$ be the predictor step for problem (1.7); i.e., Δz_p is obtained from step (1) of Algorithm 2. By comparing Algorithm 2 with Algorithm 3 ($m = 1$), we see that our proof will be complete once we show that

$$F(z + \Delta z_p) = \begin{pmatrix} 0 \\ 0 \\ \Delta X_p \Delta y_p \end{pmatrix} . \tag{2.1}$$

Writing (2.1) in further detail gives

$$A(x + \Delta x_p) - b = 0, \tag{2.2}$$

$$A^T(\lambda + \Delta \lambda_p) + (y + \Delta y_p) - c = 0, \tag{2.3}$$

$$[x + \Delta x_p]_i [y + \Delta y_p]_i = [\Delta x_p]_i [\Delta y_p]_i \,, \quad i = 1, \ldots, n. \tag{2.4}$$

By expanding we see that (2.4) is equivalent to

$$[y]_i [\Delta x_p]_i + [x]_i [\Delta y_p]_i = -[x]_i [y]_i \,, \quad i = 1, \ldots, n. \tag{2.5}$$

However, (2.2), (2.3), and (2.5) are exactly the defining relations for the Newton step. Moreover, from the definition of Δz_p in step (2) of Algorithm 2 it is clear that Δz_p is the Newton step. Hence (2.1) holds and we have established the equivalence. □

While in Mehrotra [6] no explanation for the predictor-corrector method is given, in a more recent paper, Mehrotra [7], Mehrotra offers an interpretation of a related, but somewhat different, algorithm. Following the lead of Monteiro, Adler, and Resende [9] he constructs a standard homotopy in a parameter, say δ, between problem (1.7) and a problem which had the current iteration as its solution. The primal-dual trajectory path parametrized by δ gives the solution of problem (1.7) for $\delta = 0$ and the current iterate for $\delta = 1$. He then views the iterate obtained from the predictor-corrector method as a point on a quadratic path which approximates the primal-dual trajectory path.

The equivalence represented by Theorem 2.1 was conjectured while listening to Mehrotra discuss his predictor-corrector method at the Second Asilomar Workshop on Progress in Mathematical Programming, Monterey, California, February 4–7, 1990. After proving Theorem 2.1 and while preparing this paper we received the paper of Lustig, Marsten, and Shanno [4]. In this paper the authors describe a comprehensive implementation of the Mehrotra predictor-corrector method and present impressive numerical results.

Lustig, Marsten, and Shanno [4] motivate Mehrotra's predictor-corrector method in the following manner. Rather than applying Newton's method to (1.7) to generate correction terms to the current iterate, they consider $F(z+\Delta z) = \mu \hat{e}$ directly, yielding

$$A(x + \Delta x) = b, \tag{2.6a}$$

$$A^T(\lambda + \Delta \lambda) - (y + \Delta y) = c, \tag{2.6b}$$

$$[x + \Delta x]_i [y + \Delta y]_i = \mu \,, \quad i = 1, \ldots, n \,. \tag{2.6c}$$

Simple algebra reduces (2.6) to the equivalent system

$$A\Delta x = b - Ax, \tag{2.7a}$$

$$A^T\Delta\lambda - \Delta y = c - A^T\lambda + y, \tag{2.7b}$$

$$[x]_i[\Delta y]_i + [y]_i[\Delta x]_i = \mu - [x]_i[y]_i - [\Delta x]_i[\Delta y]_i\,, \quad i = 1,\ldots,n\,. \tag{2.7c}$$

Observe that (2.7) defines the step $(\Delta x, \Delta y, \Delta\lambda)$ implicitly, i.e., in a nonlinear manner. In order to determine a step approximately satisfying (2.7) it seems reasonable to first solve $F(z) = 0$ for the Newton (predictor) step $(\Delta x_p, \Delta y_p, \Delta\lambda_p)$ and then use Δx_p and Δy_p on the right-hand side of (2.7) to solve for an "improved" step from (2.7) with this modified right-hand side.

It should be clear that the presentation (2.7), with Δx and Δy replaced by Δx_p and Δy_p only on the right-hand side reflects the level-1 composite Newton method corresponding to an unperturbed version written in the form

$$F'(z_k)\Delta z = -[F(z_k) + F(z_k - F'(z_k)^{-1}F(z_k))]\,; \tag{2.8}$$

while Mehrotra's original presentation reflects form (1.5).

Lustig, Marsten, and Shanno [4] attempt an explanation of the predictor-corrector notion in terms of trajectories parametrized by the parameter μ. Their explanation contains some ambiguity in that it is not clear to what trajectories they are referring. Moreover, any explanation based on issues derived from μ cannot give a complete picture, since the predictor-corrector notion still makes sense even when the problem formulation is free of μ; i.e., $\mu = 0$ in all cases. However, implicit in their comments is the understanding that the corrector step can be viewed as a perturbed simplified Newton step.

3. Cubic convergence. Much of the following analysis follows directly from material in Dennis and Schnabel [1] and Ortega and Rheinboldt [10]. As before, we consider problem (1.7) and use the notation $z = (x, y, \lambda)$. Also, recall that $\hat{e} = (0,\ldots,0,1,\ldots,1)^T$, where the number of zeros is $n+m$ and the number of ones is n. The pure Newton method can be written

$$N(z) = z - F'(z)^{-1}F(z) \tag{3.1}$$

and the predictor-corrector interior-point method can be written

$$\hat{N}(z) = z - \alpha F'(z)^{-1}[F(z) + F(N(z)) - \mu\hat{e}]\,. \tag{3.2}$$

Therefore

$$\begin{aligned}
\hat{N}(z) - z_* &= z - z_* - F'(z)^{-1}[F(z) + F(N(z))] \\
&\quad +(1-\alpha)F'(z)^{-1}[F(z) + F(N(z))] + \alpha\mu F'(z)^{-1}\hat{e} \\
&= F'(z)^{-1}[F'(z)(N(z)) - F(N(z)) - F'(z)(z_*)] \\
&\quad +(1-\alpha)F'(z)[F(z) + F(N(z))] + \alpha\mu F'(z)^{-1}\hat{e} \\
&= -F'(z)^{-1}\{[F(N(z)) - F(z_*) - F'(z_*)(N(z) - z_*)] \\
&\quad +[F'(z_*) - F'(z)](N(z) - z_*)\} \\
&\quad +(1-\alpha)F'(z)[F(z) + F(N(z))] + \alpha\mu F'(z)^{-1}\hat{e}.
\end{aligned} \tag{3.3}$$

Now, locally, i.e., in a neighborhood of the solution z_*, we know from the standard Newton method analysis that

$$\|N(z) - z_*\| = O(\|z - z_*\|^2) .$$

Hence, we can rewrite the four terms on the right-hand side in (3.3) and obtain

$$\|\hat{N}(z) - z_*\| = O(\|z - z_*\|^4) + O(\|z - z_*\|^3) + |1 - \alpha| O(\|z - z_*\|) + \mu O(1) ,$$

which simplifies to

$$\|\hat{N}(z) - z_*\| = O(\|z - z_*\|^3) + |1 - \alpha| O(\|z - z_*\|) + \mu O(1) . \tag{3.4}$$

In deriving (3.4) we used the fact that $\|F(z)\| = O(\|z - z_*\|)$ and $\|F(N(z))\| = O(\|z - z_*\|^2)$.

The term $\mu O(1)$ can be made $O(\|z - z_*\|^3)$ by the choice of μ. Everything now hinges on the term $|1 - \alpha| O(\|z - z_*\|)$. We must therefore take a very close look at the quantity $1 - \alpha$. Clearly, for cubic convergence, we need $|1 - \alpha|$ to be $O(\|z - z_*\|^2)$.

Assuming strict complementarity, z_* is a nondegenerate vertex solution, and z_k is feasible. Zhang, Tapia, and Dennis [15] obtained the useful expression

$$1 - \alpha_k = \frac{1 - \tau_k + \sigma_k \theta_k}{1 - \sigma_k \theta_k} + O(x_k^T y_k) \tag{3.5}$$

for the Newton interior-point method. See (3.7) of Zhang, Tapia, and Dennis [15]. In (3.5), τ_k and σ_k are as in Algorithm 1, $\theta_k \in (\frac{1}{n}, 1]$ and $O(x_k^T y_k)$ is not necessarily zero and is exactly first order. Observe that $O(x_k^T y_k) = O(\|z - z_*\|)$, since for feasible z_k we have $x_k^T y_k = \|F(z_k)\|_1$.

For the present purpose of studying $1 - \alpha_k$, the predictor-corrector primal-dual interior point method and the primal-dual interior-point method are philosophically the same; i.e., both can be viewed as perturbed Newton. In the former case the perturbation to the right-hand side of the defining relation is $\mu \hat{e} - F(z - F'(z)^{-1} F(z))$, while in the latter case the perturbation is merely $\mu \hat{e}$. Observe that these two perturbation terms differ by a term which is order $O(\|z - z_*\|^2)$ or equivalently $O((x_k^T y_k)^2)$. Hence (3.5) is also valid for the Newton predictor-corrector interior-point method. It can now be seen from (3.5) that independent of the choices for τ_k and σ_k, the term $|1 - \alpha_k|$ is at best $O(\|z - z_*\|)$ and the Newton predictor-corrector interior-point method, even for nondegenerate problems, cannot be shown to be cubically convergent by the standard approach. However, by choosing $\alpha_k = 1$ near the solution and $\mu_k = O((x_k^T y_k)^3)$ we see from (3.4) that it is possible to obtain cubic convergence. We formally state these observations as the following theorem.

THEOREM 3.1. *Let $\{x_k, y_k, \lambda_k\}$ be produced by Mehrotra's predictor-corrector interior-point method with z_0 strictly feasible. Assume*

(i) *strict complementarity,*

(ii) *x_* is a nondegenerate vertex, and*

(iii) *$\{(x_k, y_k, \lambda_k)\}$ converges to (x_*, y_*, λ_*).*

If the choices of σ_k and τ_k satisfy

$$0 \le \sigma_k \le \min(\sigma, c_1(x_k^T y_k)) \tag{3.6}$$

and

$$0 < \tau_k \le \min(\tau, 1 - c_2 x_k^T y_k), \tag{3.7}$$

where $\sigma \in [0,1), \tau \in (0,1)$ *and* $c_1, c_2 > 0$, *then the convergence is Q-quadratic*; *i.e., there exist* $\gamma_2 > 0$ *such that for k large*

$$\|(x_{k+1}, y_{k+1}, \lambda_{k+1}) - (x_*, y_*, \lambda_*)\| \leq \gamma_2 \|(x_k, y_k, \lambda_k) - (x_*, y_*, \lambda_*)\|^2 .$$

On the other hand, if instead of (3.6) *we have*

$$0 \leq \sigma_k \leq \min(\sigma, c_1 (x_k^T y_k)^2) \tag{3.8}$$

and instead of (3.7) *we have that for large k*

$$\alpha_k = 1, \tag{3.9}$$

then the convergence is Q-cubic; *i.e., there exist* $\gamma_3 > 0$ *such that for k large*

$$\|(x_{k+1}, y_{k+1}, \lambda_{k+1}) - (x_*, y_*, \lambda_*)\| \leq \gamma_3 \|(x_k, y_k, \lambda_k) - (x_*, y_*, \lambda_*)\|^3 .$$

Proof. The proof follows from combining the discussion given above with the details given for the proof of Theorem 4.1 in Zhang, Tapia, and Dennis [15]. □

4. Concluding remarks. In this paper we have studied the Mehrotra predictor-corrector philosophy and demonstrated that it is equivalent to the level-1 perturbed composite Newton philosophy.

We were intrigued by the discovery that, while the level-1 composite Newton method is known to be cubically convergent, this standard convergence rate proof applied to the predictor-corrector interior-point method gives, at best, quadratic convergence. The limitation of the standard proof results from the constrictive steplength choice forced on the method by the interior-point philosophy, i.e., requiring the iterates to remain strictly feasible with respect to the nonnegativity constraints. We demonstrated that if one drops the interior-point aspect of the predictor-corrector method locally, i.e., in a neighborhood of the solution steplength one is selected, and also chooses the centering parameter to be of the order of the duality gap cubed, then cubic convergence can be attained for nondegenerate problems.

The research presented in Zhang, Tapia, and Dennis [15], in Zhang, Tapia, and Potra [16], and the present research leads us to conjecture that we should implement Newton interior-point methods and their variants in a manner, which near the solution, sets the centering parameter to zero and takes steplength one, i.e., as old-fashioned Newton.

Acknowledgment. The authors thank an anonymous referee for critical comments that forced them to give a more consistent treatment of various conceptual notions.

REFERENCES

[1] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983. Russian edition, Mir Publishing Office, Moscow, 1988, O. Burdakov, translator.

[2] A. El-Bakry, R. A. Tapia, and Y. Zhang, *A study of indicators for identifying zero variables in interior-point methods*, SIAM Rev., 36 (1994), pp. 45–72.

[3] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior-point method for linear programming*, in Progress in Mathematical Programming, Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[4] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *On implementing Mehrotra's predictor-corrector interior point method for linear programming*, Tech. Report SOR 90-03, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, April 1990.

[5] N. MEGIDDO, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 131–158.

[6] S. MEHROTRA, *On finding a vertex solution using interior-point methods*, Tech. Report 89-22, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1989. Revised, January 1990.

[7] ———, *On the implementation of a (primal-dual) interior point method*, Tech. Report 90-03, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, March 1990.

[8] S. MIZUNO, M. J. TODD, AND Y. YE, *Anticipated behavior of the path following algorithms for linear programming*, Tech. Report No. 878, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, December 1989.

[9] R. D. C. MONTEIRO, I. ADLER, AND M. G. C. RESENDE, *A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension*, ESRC Report 88-8, Department of Industrial Engineering and Operations Research, University of California, Berkeley, March 1988.

[10] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[11] J. JI, F. POTRA, R. A. TAPIA, AND Y. ZHANG, *An interior-point method with polynomial complexity and superlinear convergence for linear complementarity problems*, TR91-23, Department of Computational and Applied Mathematics, Rice University, Houston, TX, July 1991. Submitted.

[12] J. TRAUB, *Iterative Methods for the Solution of Equations*, Prentice Hall, Englewood Cliffs, NJ, 1964.

[13] Y. ZHANG AND R. A. TAPIA, *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming*, SIAM J. Optim., 3 (1993), pp. 118–133.

[14] ———, *Superlinear and quadratic convergence of primal-dual interior-point algorithms for linear programming revisited*, JOTA, 73 (1992), pp. 229–242.

[15] Y. ZHANG, R. A. TAPIA, AND J. E. DENNIS, *On the superlinear and quadratic convergence of primal-dual interior-point linear programming algorithms*, SIAM J. Optim., 2 (1992), pp. 304–324.

[16] Y. ZHANG, R. A. TAPIA, AND F. POTRA, *On the superlinear convergence of interior-point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 57–73, February 1996
© 1996 Society for Industrial and Applied Mathematics
005

ON THE RELATIONSHIP BETWEEN THE CURVATURE INTEGRAL AND THE COMPLEXITY OF PATH-FOLLOWING METHODS IN LINEAR PROGRAMMING *

GONGYUN ZHAO†

Abstract. In this paper we study the complexity of primal-dual path-following methods which are a particular kind of interior point method for solving linear programs. In particular we establish a relationship between the complexity and the integral of a weighted curvature along the central trajectory (path) defined in the interior of the feasible solution space of a linear program. An important property of the trajectory, viz., that its higher order derivatives are bounded by a geometric series, is presented. Applying this property we show that the complexity of such methods is bounded from below and from above by the curvature integral. This theorem reduces the complexity analysis to a relatively simple problem of estimating the curvature integral.

Key words. linear programming, central trajectory (path), derivatives, bounds of derivatives, path-following methods, complexity, curvature integral

AMS subject classifications. 90C60, 90C05, 49M35, 68Q25

1. Introduction.

In this paper we study the complexity of path-following methods, a kind of interior point method, for solving a linear program

$$\text{(LP)} \qquad \begin{aligned} \min_x \quad & c^T x \\ & Ax = b, \\ & x \geq 0 \end{aligned}$$

and its dual

$$\text{(LD)} \qquad \begin{aligned} \max_{y,s} \quad & b^T y \\ & A^T y + s = c, \\ & s \geq 0, \end{aligned}$$

where A is an $m \times n$-matrix with $m \leq n$, b is an m-vector and c is an n-vector. To simplify the discussion, we make the following two assumptions throughout this paper:

A1. Matrix A is of full rank, i.e., $\text{rank}(A) = m$;

A2. Problems (LP) and (LD) possess interior feasible solutions $x > 0$ and y, s with $s > 0$, respectively.

Many polynomial complexity methods have been proposed since Karmarkar presented his work [3]. (For a survey, cf. [5] and [11].) The complexity of these methods is defined as the number of iterations needed to achieve an approximating solution to certain accuracy times the number of operations needed at each iteration. If Newton iteration is carried out, as most interior point methods do, $O(n^3)$ operations are needed at each iteration, or $O(n^{2.5})$ operations on average if the one-dimensional adaption technique proposed by Karmarkar [3] is used. This part of the complexity is not

*Received by the editors May 14, 1993; accepted for publication (in revised form) October 6, 1994.

†Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, 0511 Singapore (matzgy@leonis.nus.sg).

the object of this paper. Our attention focuses on the first part of the complexity. Therefore, in the context of this paper, the complexity is referred to as the number of iterations. For primal-dual path-following methods, it is very convenient to use the duality gap $r = x^T s/n$ as a parameter of the central trajectory (path), which is followed to find the optimal solution. Every iteration (x_i, s_i) is associated to a parameter value $r_i = x_i^T s_i/n$. The complexity is therefore referred to as $N = N(r_0, \epsilon)$, the number of iterations $r_0 > r_1 > \cdots > r_N$, which are needed to reduce parameter r from an initial parameter value r_0 to a final value r_N satisfying $0 < r_N \leq \epsilon$. For certain given termination criterion $\epsilon > 0$, $r_i - r_{i+1}$ is called step size at iteration i, and it is determined by certain rules given in an algorithm. The conventional way of estimating the complexity is to find a lower bound of step sizes $r_i - r_{i+1}$, $i = 0, 1, \ldots, N-1$, and obtain an upper bound of the complexity as the quotient of the total length $r_0 - \epsilon$ divided by the lower bound of step sizes. Such a conservative estimation will largely overestimate the actual complexity, if step size varies enormously from iteration to iteration, as it does in practice.

A natural way to improve the estimation of the complexity is to consider all real step sizes instead of their lower bound. For this purpose we set a step size rule as follows. At iteration r_i, r_{i+1} is so determined that the intersection angle of the tangents of the central trajectory at r_i and r_{i+1}, i.e., the change of the angle from r_i to r_{i+1}, is equal to a *constant* which is independent of i. Stated equivalently, the integral of the curvature of the central trajectory from r_i to r_{i+1} should be a *constant*. Hence the total curvature integral from r_0 to r_N is equal to N times the constant. This implies the equivalence of the total curvature integral and the complexity (see Theorems 4.2 and 4.3). This equivalence opens a promising way for the complexity analysis which may be studied in the worst case sense as well as in a probability sense.

The relationship between the complexity (or $N(r_0, \epsilon)$) and curvature integral was independently conjectured by Sonnevend, appearing in the joint work with Stoer and Zhao [10], and by Karmarkar [4]. The first strict proof of this conjecture was provided by Zhao and Stoer [14].

In this paper we present a new proof of the complexity/integral relationship, different from that in [14]. We first present a theorem describing an interesting feature of the central trajectory: The size of the higher order derivatives of the central trajectory $z(r)$ does not grow faster than a geometric series, that is,

$$\left\| \frac{d^k z}{dr^k} \right\|_2 \leq C\beta^k \qquad \text{for all } k = 1, 2, \ldots,$$

where C is a constant and $\beta = \beta(r)$ can be roughly considered as the curvature of the central trajectory $z(r)$ at point r. This theorem is then applied to prove the complexity/integral relationship.

The rest of this paper is organized as follows. In §2, we study the higher order derivatives of the central trajectory. In §3, we present theorems describing the boundness of the higher order derivatives. These theorems will then be applied to prove the main theorem, the complexity/integral relationship which is presented in §4, together with a primal-dual path-following algorithm. Sections 5 and 6 contain a series of lemmas. Finally, in §7 the main theorem is proved. The Appendix at the end of the paper contains a proof of a lemma in §6 which is similar to a lemma in [14].

Throughout this paper we use the following notation. For $x, s \in R^n$ and $v \in R$, we define

$$x \circ s := (x_1 s_1, \ldots, x_n s_n)^T, \quad x^v := (x_1^v, \ldots, x_n^v)^T, \quad [x] := \operatorname{diag}(x).$$

2. The derivatives of the central trajectory. One of the essential ingredients of the interior point methods is the central trajectory (path). For problems (LP) and (LD) we define the ξ-trajectory $z(r,\xi) = (x(r,\xi), s(r,\xi))$, for any fixed positive n-vector ξ and parameter $r \geq 0$, as solutions of the following equations.

$$(\mathrm{LPD})_{r,\xi} \qquad \begin{aligned} x \circ s &= r\xi, \\ Ax &= b, \\ A^T y + s &= c, \\ x \geq 0, &\; s \geq 0. \end{aligned}$$

The *central trajectory* is the ξ-trajectory for $\xi = e = (1,\ldots,1)^T$. It has been proved that the central trajectory, indeed any ξ-trajectory, is analytic; cf. [9] and [1]. If $r = 0$, the solution of $(\mathrm{LPD})_{r,\xi}$ satisfies the Kuhn–Tucker condition and hence is an optimal solution of (LP) and (LD). Therefore, for any fixed $\xi > 0$, the ξ-trajectory always leads to an optimal solution as r tends to zero. This situation motivates path-following methods.

We note that $x(r,\xi)$ and $y(r,\xi)$ are each separately the minimum of a ξ-weighted Frisch-type logarithmic barrier function

$$\begin{aligned} x(r,\xi) &:= \operatorname{argmin}\Big\{ \frac{c^T x}{r} - \sum_{i=1}^{n} \xi_i \ln x_i \Big| b + Ax = 0,\ x > 0 \Big\}, \\ y(r,\xi) &:= \operatorname{argmax}\Big\{ \frac{b^T y}{r} - \sum_{i=1}^{n} \xi_i \ln(c_i - a_i^T y) \Big| c - A^T y > 0 \Big\}. \end{aligned} \tag{2.1}$$

These convex functions give us an intuitive interpretation of the ξ-trajectories.

Now we begin to compute the derivatives of the ξ-trajectory $z(r,\xi)$ using Taylor series expansions (also cf. [1]).

LEMMA 2.1. *For given $\xi > 0$ and $r > 0$ the ith derivatives (with respect to r)*

$$x_i := x_i(r,\xi) := \frac{x^{(i)}(r,\xi)}{i!},$$

s_i and y_i, $i = 1,2,\ldots$, satisfy the following equations:

$$\begin{aligned} & s \circ x_1 + x \circ s_1 = \xi, \\ & s \circ x_k + x \circ s_k = -\sum_{i=1}^{k-1} x_i \circ s_{k-i}, \qquad k \geq 2, \\ & Ax_i = 0, \\ & A^T y_i + s_i = 0, \qquad i = 1,2,\ldots, \end{aligned} \tag{2.2}$$

where $(x,s) := (x_0, s_0)$ are solutions of $(\mathrm{LPD})_{r,\xi}$.

Proof. For any $r > 0$ and $r + \epsilon > 0$, let formally $x(r+\epsilon) = \sum_{i=0}^{\infty} x_i \epsilon^i$ and $s(r+\epsilon) = \sum_{i=0}^{\infty} s_i \epsilon^i$ in the first equation of $(\mathrm{LPD})_{r+\epsilon,\xi}$. Then its right-hand side becomes

$$\begin{aligned} & \left(\sum_{i=0}^{\infty} x_i \epsilon^i \right) \circ \left(\sum_{i=0}^{\infty} s_i \epsilon^i \right) \\ & = x \circ s + \sum_{k=1}^{\infty} (s \circ x_k + x \circ s_k) \epsilon^k + \sum_{k=2}^{\infty} \sum_{i=1}^{k-1} x_i \circ s_{k-i} \epsilon^k. \end{aligned}$$

A comparison with the right-hand side $(r+\epsilon)\xi$ gives the first two equations of (2.2).

The last two equations of (2.2) are obtained immediately by differentiating the second and third equations of $(\mathrm{LPD})_{r,\xi}$. □

We introduce the weighted quantities

$$
\begin{aligned}
&p_i := r\xi^{1/2} \circ x^{-1} \circ x_i = \xi^{-1/2} \circ s \circ x_i, \\
&q_i := r\xi^{1/2} \circ s^{-1} \circ s_i = \xi^{-1/2} \circ x \circ s_i, \\
&\tilde{A} := A[\xi^{1/2} \circ s^{-1}] = \frac{1}{r} A[\xi^{-1/2} \circ x].
\end{aligned} \tag{2.3}
$$

The formulae of Lemma 2.1 can be simplified as follows.

$$
\begin{aligned}
&p_1 + q_1 = \xi^{1/2}, \\
&p_k + q_k = -\frac{1}{r}\xi^{-1/2} \circ \sum_{i=1}^{k-1} p_i \circ q_{k-i}, \qquad k \geq 2, \\
&\tilde{A}p_i = 0, \\
&r\tilde{A}^T y_i + q_i = 0, \qquad i = 1, 2, \ldots.
\end{aligned} \tag{2.4}
$$

The above equations show that p_i and q_i are orthogonal projections of a vector, respectively, on the null space and the image space of $\tilde{A}$, that is,

$$
\begin{aligned}
&p_1 = N\xi^{1/2}, \qquad q_1 = M\xi^{1/2}, \\
&p_k = -\frac{1}{r} N \left(\xi^{-1/2} \circ \sum_{i=1}^{k-1} p_i \circ q_{k-i} \right), \\
&q_k = -\frac{1}{r} M \left(\xi^{-1/2} \circ \sum_{i=1}^{k-1} p_i \circ q_{k-i} \right), \qquad k \geq 2,
\end{aligned} \tag{2.5}
$$

where $M := \tilde{A}^T(\tilde{A}\tilde{A}^T)^{-1}\tilde{A}$ and $N := I - M$ are orthogonal projections.

Variables p_i and q_i are useful and important because they are characterized both as weighted derivatives (2.3) and projections (2.5). Equation (2.5) shows that p_i and q_i can be recursively generated. From this recursive relation, we find in §3 the growth rate of the higher order derivatives of a ξ-trajectory.

3. Bounds of derivatives—geometrical growth. In this section we present an important property of ξ-trajectories $z(r) = z(r, \xi)$: The derivatives of $z(r)$ grow at most geometrically, that is,

$$
(\|p_k\|_2^2 + \|q_k\|_2^2)^{1/2} \leq C \cdot \beta^k \qquad \text{for all } k = 1, 2, \ldots,
$$

where C is a constant and $\beta := \beta(r, \xi)$, roughly speaking, is a weighted curvature. The following lemma plays the key role in the proof of the above property.

LEMMA 3.1. *The solution of the difference equation*

$$
v_k = \rho \sum_{i=1}^{k-1} v_i v_{k-i}, \qquad k = 2, 3, \ldots
$$

is given by

$$
v_k = 2(-1)^{k-1} \binom{1/2}{k} (4\rho)^{k-1} v_1^k, \qquad k \geq 2, \tag{3.1}
$$

resulting in the estimate

$$|v_k| \leq \frac{1}{k}(4\rho)^{k-1}|v_1|^k.$$

Proof. For given v_1, let $\epsilon > 0$ be so small that $|4\rho v_1 \epsilon| < 1$. Consider the generating function

$$f(\epsilon) = \sum_{k=1}^{\infty} v_k \epsilon^k \ .$$

Then from

$$\sum_{k=2}^{\infty} v_k \epsilon^k = \rho \sum_{k=2}^{\infty} \sum_{i=1}^{k-1} v_i v_{k-i} \epsilon^k$$

follows

$$f(\epsilon) - v_1 \epsilon = \rho f(\epsilon)^2 \ .$$

This gives

$$f(\epsilon) = \frac{1 - \sqrt{1 - 4\rho\epsilon v_1}}{2\rho} \ .$$

Taylor expansion of the right-hand side then gives v_k of (3.1). □

COROLLARY 3.2. *If* $0 \leq u_k \leq \rho \sum_{i=1}^{k-1} u_i u_{k-i}$ *for all* $k \geq 2$ *then*

$$u_k \leq v_k \qquad \text{for } k = 1, 2, \ldots,$$

where v_k *are as* (3.1) *and* $v_1 := u_1$.

Proof. Use induction. If $u_i \leq v_i$ for all $1 \leq i \leq k-1$, then

$$u_k \leq \rho \sum_{i=1}^{k-1} u_i u_{k-i} \leq \rho \sum_{i=1}^{k-1} v_i v_{k-i} = v_k. \qquad \square$$

Denote

$$\sigma_k := p_k + q_k$$

and, since p_k and q_k are orthogonal, we have

$$\|\sigma_k\|_2 = (\|p_k\|_2^2 + \|q_k\|_2^2)^{1/2} \geq \max\{\|p_k\|_2, \|q_k\|_2\}.$$

Then from (2.4) follows immediately

$$\|\sigma_k\|_2 = \frac{1}{r}\left\|\xi^{-1/2} \circ \sum_{i=1}^{k-1} p_i \circ q_{k-i}\right\|_2 \leq \frac{\|\xi^{-1/2}\|_\infty}{r} \sum_{i=1}^{k-1} \|\sigma_i\|_2 \|\sigma_{k-i}\|_2.$$

Then according to Corollary 3.2, one obtains for some constant C

$$\|\sigma_k\|_2 \leq C(4\rho\|\sigma_1\|_2)^k.$$

But this estimation is not very useful, because even in the ideal case $\xi = e$, i.e., on the central trajectory, we have

$$\|\sigma_1\|_2 = \left\|\xi^{1/2}\right\|_2 = \|e\|_2 = \sqrt{n},$$

and the upper bound $(\sqrt{n})^k$ for $\|\sigma_k\|_2$ is too large for good complexity estimates. The reason for the large value $\sqrt{n}$ is the dimension n of σ_1; therefore, we hope that a dimension-independent norm like $\|\cdot\|_\infty$ for p_1 and q_1 could improve this situation.

Let's introduce notation

$$\kappa := \kappa(r,\xi) = \frac{1}{\sqrt{r}}\|x_1 \circ s_1\|_2^{1/2},$$

which has several equivalent expressions

$$\kappa^2 = \frac{1}{r}\|x_1 \circ s_1\|_2 = \frac{1}{r}\|\dot{x} \circ \dot{s}\|_2 = \frac{1}{r^2}\|p_1 \circ q_1\|_2 = \frac{1}{r}\|\xi^{1/2} \circ \sigma_2\|_2. \tag{3.2}$$

We call κ (weighted) *curvature* of the ξ-trajectory since, in essence, it is the second order derivative σ_2 in Euclidean norm.

From

$$p_1 \circ (\xi^{1/2} - p_1) = q_1 \circ (\xi^{1/2} - q_1) = p_1 \circ q_1,$$

we have

$$\|p_1\|_\infty^2 - \|\xi\|_\infty^{1/2}\|p_1\|_\infty \le \|p_1 \circ q_1\|_2 = (r\kappa)^2.$$

It follows from the above quadratic inequality that

$$\|p_1\|_\infty \le \frac{\|\xi\|_\infty^{1/2} + \sqrt{\|\xi\|_\infty + 4(r\kappa)^2}}{2}.$$

Using the inequality $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for any nonnegative numbers a and b, we obtain (using the same proof for q_1)

$$\max\{\|p_1\|_\infty, \|q_1\|_\infty\} \le \beta, \tag{3.3}$$

where $\beta := \beta(r,\xi) = r\kappa + \|\xi\|_\infty^{1/2}$.

Now we give one of the main theorems in this paper.

THEOREM 3.3. *For any $\xi > 0$ and for all $k = 2, 3, \ldots$, we have*

$$\max\{\|p_k\|_2, \|q_k\|_2\} \le \|\sigma_k\|_2 \le v_k \le \frac{1}{k}(4\rho)^{k-1}\beta^k,$$

where v_k is as (3.1), *$v_1 = \beta$, and*

$$\rho := \rho(r,\xi) := \|\xi^{-1/2}\|_\infty / r, \quad \beta := \beta(r,\xi) := r\kappa + \|\xi\|_\infty^{1/2}. \tag{3.4}$$

Proof. For $k = 2$ the inequality can be checked directly. For $k \ge 3$ we have by (2.4)

$$\begin{aligned}\|\sigma_k\|_2 &\le \frac{1}{r}\|\xi^{-1/2}\|_\infty \sum_{i=1}^{k-1} \|p_i \circ q_{k-i}\|_2 \\ &\le \frac{1}{r}\|\xi^{-1/2}\|_\infty \Big(\sum_{i=2}^{k-2} \|\sigma_i\|_2 \cdot \|\sigma_{k-i}\|_2 + (\|p_1\|_\infty + \|q_1\|_\infty)\,\|\sigma_{k-1}\|_2 \Big).\end{aligned}$$

Let $u_1 := \beta$ and $u_k := \|\sigma_k\|_2$ for $k \geq 2$. Then using (3.3)

$$u_k \leq \rho \sum_{i=1}^{k-1} u_i u_{k-i}$$

and we complete the proof by Corollary 3.2. □

The above theorem essentially describes the growth of the derivatives. For practical use we still need the following theorem about the higher order derivatives of σ_k.

THEOREM 3.4. *For any $\xi > 0$ and for all $k = 2, 3, \ldots$ and $u = 0, 1, \ldots$, we have*

$$\frac{1}{u!}\|\sigma_k^{(u)}\|_2 \leq \binom{k+u}{k} v_{k+u} \leq \frac{(k+u-1)!}{k!u!}(4\rho)^{k+u-1}\beta^{k+u},$$

where v_k, ρ and β are as in Theorem 3.3.

Proof. Using the Leibniz formula we obtain the derivatives of $\sigma_k = p_k + q_k = -\xi^{-1/2} \circ \sum_{i=1}^{k-1} \frac{1}{i!(k-i)!} x^{(i)} \circ s^{(k-i)}$:

$$\begin{aligned}
\sigma_k^{(u)} &= -\xi^{-1/2} \circ \sum_{i=1}^{k-1} \sum_{j=0}^{u} \frac{\binom{u}{j}}{i!(k-i)!} x^{(i+j)} \circ s^{(k+u-i-j)} \\
&= -\frac{(k+u)!}{k! \cdot r} \xi^{-1/2} \circ \sum_{i=1}^{k-1} \sum_{j=0}^{u} P(k,u;i,j) p_{i+j} \circ q_{k+u-i-j} \qquad \text{by (2.3)},
\end{aligned}$$

where $P(k,u;i,j) := \binom{k}{i}\binom{u}{j}/\binom{k+u}{i+j}$. By Theorem 3.3

$$\begin{aligned}
\frac{1}{u!}\|\sigma_k^{(u)}\|_2 &\leq \binom{k+u}{k} \rho \sum_{i=1}^{k-1} \sum_{j=0}^{u} P(k,u;i,j) v_{i+j} v_{k+u-i-j} \\
&= \binom{k+u}{k} \rho \sum_{l=1}^{k+u-1} \sum_{i+j=l, 1 \leq i \leq k-1, 0 \leq j \leq u} P(k,u;i,j) v_l v_{k+u-l}.
\end{aligned}$$

For fixed l $(0 \leq l \leq k+u)$

$$\sum_{i+j=l, 0 \leq i \leq k, 0 \leq j \leq u} P(k,u;i,j) = 1.$$

Hence by Lemma 3.1,

$$\begin{aligned}
\frac{1}{u!}\|\sigma_k^{(u)}\|_2 &\leq \binom{k+u}{k} \rho \sum_{l=1}^{k+u-1} v_l v_{k+u-l} \\
&= \binom{k+u}{k} v_{k+u} \\
&\leq \frac{(k+u-1)!}{k!u!}(4\rho)^{k+u-1}\beta^{k+u}. \qquad \square
\end{aligned}$$

4. The algorithm and the main theorems. The idea of using a primal-dual algorithm to follow the central path, in conjunction with the step length formula (4.1),

in order to find complexity estimates, is essentially due to Sonnevend [10]. Here, we generalize this idea by considering not only the central trajectory but also neighboring ξ-trajectories. In this paper we analyse only this algorithm; however, the method developed here can certainly be used to analyse other algorithms.

For $z = (x, s) > 0$, $r > 0$ and a vector $\xi > 0$ the number

$$d(z, r, \xi) = d(x, s, r, \xi) := \left\| \frac{1}{r} x \circ s - \xi \right\|_2$$

defines a distance of the point z from the point $z(r, \xi)$ on the ξ-trajectory. Specifically we write $d(z, r)$ for $\xi = e$. This distance measure, first proposed by Roos and Vial [8], turns out to be of great convenience.

ALGORITHM 4.1.

0. *Given* $\epsilon > 0$, $T_1 > T_0 > 0$ *and an interior feasible solution* $z^{[0]} = (x^{[0]}, s^{[0]}) > 0$ *which satisfies*

$$d(z^{[0]}, r_0) \leq T_0,$$

where $r_0 := (x^{[0]})^T (s^{[0]})/n$.

For $k = 0, 1, 2, \ldots$, *let* $r_k := (x^{[k]})^T (s^{[k]})/n$ *and* $\xi_k := \frac{1}{r_k} x^{[k]} \circ s^{[k]}$. *Compute* r_{k+1}, $x^{[k+1]}$, $s^{[k+1]}$ *by:*

1. (Predictor-Step) *Compute the derivatives* $\dot{z}^{[k]}(r_k)$ *of the* ξ_k*-trajectory* $z^{[k]}(r) := z(r, \xi_k)$ *at* r_k, *and consider the line* $\bar{z}^{[k]}(r) := z^{[k]}(r_k) + (r - r_k)\dot{z}^{[k]}(r_k)$. *Determine* $r_{k+1} := r$ *so that*

$$d(\bar{z}^{[k]}(r), r, \xi_k) = T_1,$$

which is explicitly given by (cf. (2.4) in [14])

$$\|\dot{x}^{[k]} \circ \dot{s}^{[k]}(r_k)\|_2 \frac{(r - r_k)^2}{r} = T_1. \tag{4.1}$$

2. (Corrector-Step) *Apply Newton's method to equation* $(\mathrm{LPD})_{r_{k+1}, e}$, *with starting point* $\bar{z}^{[k]}(r_{k+1})$, *to find an approximate feasible solution* $z^{[k+1]} > 0$, *which satisfies* $d(z^{[k+1]}, r_{k+1}) \leq T_0$.

If $r_{k+1} > \epsilon$, *go to* 1.

Remarks. 1. The starting point $z^{[0]}$ is given by a phase I algorithm, which will not be discussed here and is referred to, e.g., in [2].

2. It is worth noting that $\|\dot{x}^{[k]} \circ \dot{s}^{[k]}(r_k)\|_2^{1/2} = \sqrt{r_k}\kappa(r_k)$; hence the step size at r_k is inversely proportional to the curvature at r_k.

3. Using (3.2) we can rewrite (4.1) as

$$r_k - r_{k+1} = \sqrt{\frac{T_1 r_k r_{k+1}}{\|p_1^{[k]} \circ q_1^{[k]}(r_k)\|_2}} = O\left(\frac{r_k}{\|p_1^{[k]} \circ q_1^{[k]}(r_k)\|_2^{1/2}} \right).$$

It is well known that $\|p_1 \circ q_1(r)\|_2$ can be as small as $O(1)$, (in fact, it tends to zero as r tends to zero; cf. [12, 13]) and it is bounded from above by $O(n)$ according to (2.5). Therefore the step size $|r_k - r_{k+1}|$ defined by (4.1) varies between $O(r_k/\sqrt{n})$ and $O(r_k)$.

We now present two theorems that describe a relationship between the number of iterations N and an integral. This result was first conjectured and almost exactly formulated by Sonnevend. He suggested also a proposition similar to Lemma 5.2 as a

way to prove these theorems (cf. [10]). Independently, Karmarkar [4] also conjectured similar results.

THEOREM 4.2. *There are universal constants $T_1 > 0$ and $T_0 > 0$, such that if Algorithm* 4.1 *is run using these constants then for any $r_0 > \epsilon > 0$, the number $N := N(r_0, \epsilon)$ of iteration steps $r_k \to r_{k+1}$ of Algorithm* 4.1 *with*

$$r_0 > r_1 > \cdots > r_N > \epsilon \geq r_{N+1}$$

is bounded above by

$$N \leq C_1 \int_{\epsilon}^{r_0} \kappa(r, e)dr + C_2 \ln \frac{r_0}{\epsilon}, \tag{4.2}$$

where C_1 and C_2 are absolute constants.

THEOREM 4.3. *If the curvature $\kappa(r, e)$ on the central trajectory of a problem $(\mathrm{LPD})_{r,e}$ satisfies*

$$r\kappa(r, e) \geq \nu \quad \forall r \in [\epsilon, r_0], \tag{4.3}$$

for some $\nu > 0$, then there exist positive constants T_0, T_1, C_3, depending only on ν, such that the number N of iterations needed by Algorithm 4.1 *is also bounded below by*

$$N \geq C_3 \int_{\epsilon}^{r_0} \kappa(r, e)dr - 1. \tag{4.4}$$

Remark. As shown in Remark 3 to Algorithm 4.1, $r\kappa(r, e)$ varies between $O(1)$ and $O(\sqrt{n})$. Hence the integral $\int_{\epsilon}^{r_0} \kappa(r, e)dr$ varies between $O(\ln(r_0/\epsilon))$ and $O(\sqrt{n}\ln(r_0/\epsilon))$. These theorems then tell us that the iteration complexity N is between $O(\ln(r_0/\epsilon))$ and $O(\sqrt{n}\ln(r_0/\epsilon))$.

Theorems 4.2 and 4.3 are proved in §7 after a series of lemmas prepared in §§5 and 6. It is worthwhile to note that the constants in the following lemmas are not optimally selected. They are mainly used theoretically to show the *existence* of constants C_1, C_2, and C_3 in these theorems for certain choice of T_0 and T_1.

5. Variation of the curvature along the central trajectory. In this section we estimate the difference $|\kappa(r_1, e) - \kappa(r_2, e)|$ of the curvature $\kappa(r, e)$. Then we show that $\int_{r_2}^{r_1} \kappa(r, e)dr \geq \gamma$ if r_1 and r_2 satisfy certain conditions, where $\gamma > 0$ is a constant.

For simplicity we write in this section $z(r) := z(r, e)$, $\sigma_k(r) := \sigma_k(r, e)$ $\kappa(r) := \kappa(r, e)$, etc., by omitting the fixed vector e. Now we make use of the results developed in §3 to prove the following lemma.

LEMMA 5.1. *Let $0 < T_2 < 1/5$. If*

$$r_1\kappa(r_1) \geq 4 \tag{5.1}$$

then for

$$r_2 := r_1 - \frac{T_2}{\kappa(r_1)}$$

the following inequality holds for all $k \geq 2$:

$$\|\sigma_k(r_1) - \sigma_k(r)\|_2 \leq C_k(T_2)(\kappa(r_1))^{k+1}r_1(r_1 - r) \qquad \forall r \in [r_2, r_1],$$

where

$$C_k(T_2) := \frac{5^{k+1}}{4(1-5T_2)^k}.$$

Proof. Using Taylor expansion and Theorem 3.4 we have for any $r \in [r_2, r_1]$ and any $k \geq 2$,

$$\begin{aligned} \|\sigma_k(r_1) - \sigma_k(r)\|_2 &\leq \sum_{u=1}^{\infty} \frac{1}{u!} \|\sigma_k^{(u)}(r_1)\|_2 (r_1 - r)^u \\ &\leq \frac{(4\rho\beta)^k}{4\rho} \sum_{u=1}^{\infty} \frac{(k+u-1)!}{k!u!} (4\rho\beta)^u (r_1 - r)^u, \end{aligned} \tag{5.2}$$

where $\rho = 1/r_1$ and $\beta = r_1\kappa(r_1) + 1$. Differentiating

$$\sum_{u=0}^{\infty} t^u = \frac{1}{1-t}$$

$k-1$ times with respect to $t \in [0, 1[$, we obtain

$$\sum_{u=1}^{\infty} \frac{(k+u-1)!}{(k-1)!u!} t^u = \frac{1}{(1-t)^k} - 1 \leq \frac{kt}{(1-t)^k}. \tag{5.3}$$

By condition (5.1)

$$4\rho\beta = \frac{4}{r_1}(r_1\kappa(r_1) + 1) = 4\kappa(r_1)\Big(1 + \frac{1}{r_1\kappa(r_1)}\Big) \leq 5\kappa(r_1),$$

and since $r \geq r_2$,

$$4\rho\beta(r_1 - r) \leq 5\kappa(r_1)(r_1 - r) \leq 5T_2 < 1.$$

Then it follows from (5.2) and (5.3) with $t = 4\rho\beta(r_1 - r)$ that

$$\|\sigma_k(r_1) - \sigma_k(r)\|_2 \leq \frac{(5\kappa(r_1))^{k+1}(r_1 - r)}{4\rho(1-5T_2)^k}. \qquad \square$$

For the first order algorithm (e.g., Algorithm 4.1) we need only the result above with $k = 2$. In this case, noting that $r\kappa(r)^2 = \|\sigma_2(r)\|_2$, we have

$$\|\sigma_2(r_1) - \sigma_2(r)\|_2 \leq C_2(T_2)T_2\|\sigma_2(r_1)\|_2 \frac{(r_1 - r)}{(r_1 - r_2)} \quad \forall r \in [r_2, r_1] \tag{5.4}$$

where $C_2(T_2) = 5^3/(4(1-5T_2)^2)$. If $T_2 \leq 1/41$, then $C_2(T_2) \leq 41$ and $C_2(T_2)T_2 \leq 1$.

LEMMA 5.2. *Let* $0 < T_2 \leq 1/41$. *If*

$$r_1\kappa(r_1) \geq 4$$

then for

$$r_2 := r_1 - \frac{T_2}{\kappa(r_1)}$$

the following inequalities hold:

$$\frac{2}{3}T_2 \le \int_{r_2}^{r_1} \kappa(r)dr \le \frac{3}{2}T_2. \tag{5.5}$$

Proof. From $\kappa(r)^2 = \|\sigma_2(r)\|_2/r$ and (5.4), and noting that $C_2(T_2)T_2 \le 1$, we have

$$\begin{aligned}
\int_{r_2}^{r_1} \kappa(r)dr &= \int_{r_2}^{r_1} \sqrt{\frac{1}{r}\|\sigma_2(r)\|_2}dr \\
&\ge \int_{r_2}^{r_1} \sqrt{\frac{1}{r_1}(\|\sigma_2(r_1)\|_2 - \|\sigma_2(r_1) - \sigma_2(r)\|_2)}dr \\
&\ge \kappa(r_1)\int_{r_2}^{r_1} \sqrt{1 - \frac{r_1 - r}{r_1 - r_2}}dr \\
&= \frac{2}{3}\kappa(r_1)(r_1 - r_2) \\
&= \frac{2}{3}T_2.
\end{aligned}$$

This proves the first inequality in (5.5). To prove the second inequality in (5.5), we notice that

$$\frac{r_1}{r} \le \frac{r_1}{r_2} = \frac{r_1\kappa(r_1)}{r_1\kappa(r_1) - T_2} \le \frac{4}{4 - T_2} \le \frac{164}{163}.$$

Then similar to the above we can show that

$$\begin{aligned}
\int_{r_2}^{r_1} \kappa(r)dr &\le \sqrt{\frac{r_1}{r_2}}\int_{r_2}^{r_1} \sqrt{\frac{1}{r_1}(\|\sigma_2(r_1)\|_2 + \|\sigma_2(r_1) - \sigma_2(r)\|_2)}dr \\
&\le \sqrt{\frac{r_1}{r_2}}\kappa(r_1)\int_{r_2}^{r_1} \sqrt{1 + \frac{r_1 - r}{r_1 - r_2}}dr \\
&\le \frac{3}{2}T_2. \qquad \square
\end{aligned}$$

6. Other lemmas. In §5 we estimated the curvature integral along the *central* trajectory. However, iterates generated by our algorithm, actually by any practical algorithm, are not exactly on the central trajectory. Therefore, the difference $|\kappa(r,\xi) - \kappa(r,e)|$ needs to be analysed. The following lemma estimates this difference. A similar result, but not the same, can be found in Zhao and Stoer [14]. A proof of the lemma is provided in the Appendix.

LEMMA 6.1. *If* $\|\xi - e\|_2 \le 1/60$ *and* $r\kappa(r,e) \ge 4$, *then the following inequality holds.*

$$\left|\kappa(r,\xi)^2 - \kappa(r,e)^2\right| \le 30\kappa(r,e)^2\|\xi - e\|_2.$$

We will derive a similar result to Lemma 5.2. However, the interval $[r_2, r_1]$ is now determined by the curvature $\kappa(r_1,\xi)$ on the ξ-trajectory instead of the central trajectory.

LEMMA 6.2. *Let* $0 < T_2 \le 1/(41\sqrt{2})$. *For any* $0 < \xi \in R^n$ *with* $\|\xi - e\|_2 \le 1/60$, *any* $r_1 > 0$ *satisfying* $r_1\kappa(r_1,e) \ge 4$, *and* $r_2 := r_1 - T_2/\kappa(r_1,\xi)$, *the following inequalities hold:*

$$\left(\frac{2}{3}\right)^{3/2} T_2 \le \int_{r_2}^{r_1} \kappa(r,e)dr \le \frac{3}{\sqrt{2}}T_2. \tag{6.1}$$

Proof. By Lemma 6.1 we have

$$|\kappa(r,\xi)^2 - \kappa(r,e)^2| \le \frac{1}{2}\kappa(r,e)^2;$$

hence

$$\frac{1}{\sqrt{2}}\kappa(r,e) \le \kappa(r,\xi) \le \sqrt{\frac{3}{2}}\kappa(r,e).$$

Then we have the following estimate for r_2:

$$r_1 - \frac{\sqrt{2}T_2}{\kappa(r,e)} \le r_2 \le r_1 - \frac{\sqrt{\frac{2}{3}}T_2}{\kappa(r,e)}.$$

Let $\sqrt{2}T_2$ and $\sqrt{\frac{2}{3}}T_2$, respectively, be the T_2 in Lemma 5.2. Then we obtain the inequalities (6.1). □

The following lemma refers to the number of iterations needed at each Corrector-Step of Algorithm 4.1. The proof has become well known; hence we omit it in this paper. The reader may refer to Theorem 2.3 in Zhao and Stoer [14].

LEMMA 6.3. *Let $0 < T_0 < T_1' < 1/2$. Suppose $\bar{z} := (\bar{x}, \bar{s})$ is a feasible solution satisfying $d(\bar{z}, r) \le T_1'$, where $r = \bar{x}^T\bar{s}/n$. Then it needs only a bounded number of Newton iterations to obtain from $\bar{z}$ an interior feasible solution $z = (x, s)$, with $d(z, r) \le T_0$. The number of iterations depends only on T_0 and T_1'.*

Lemma 6.3 shows that the Corrector-Step of Algorithm 4.1 needs only a universally bounded number of Newton iterations if $T_0 + T_1 < 1/2$. Therefore, up to a factor depending only on T_0 and T_1, the total complexity of the algorithm is measured by the number of iterations $N := N(r_0, \epsilon)$ of Theorem 4.2.

7. Proof of Theorems 4.2 and 4.3. According to whether

$$r_k\kappa(r_k,e) \ge 4 \quad \text{and} \quad \frac{r_{k+1}}{r_k} \ge \frac{1}{2} \tag{7.1}$$

holds or not, the iteration steps $r_k \to r_{k+1}$, $k = 0, 1, \ldots, N-1$, will be divided in two parts:

$$I_1 := \{k \mid (7.1) \text{ is satisfied for } k\}, \quad I_2 := \{k \mid (7.1) \text{ is not satisfied for } k\}.$$

Denote by $N_1 = |I_1|$ the number of iterations in I_1, and $N_2 = |I_2|$. Obviously $N = N_1 + N_2$. We will show that

$$N_1 \le C_1 \int_{\epsilon}^{r_0} \kappa(r,e)dr \tag{7.2}$$

and

$$N_2 \le C_2 \ln\frac{r_0}{\epsilon}, \tag{7.3}$$

where C_1 and C_2 are positive constants depending only on the choice of T_0 and T_1. We select the positive numbers T_0 and T_1 in Algorithm 4.1 as follows:

$$0 < T_0 \le \frac{1}{60}, \qquad 0 < T_1 \le \frac{1}{200}.$$

To prove (7.2) let $k \in I_1$. By (4.1)

$$\kappa(r_k, \xi_k)(r_k - r_{k+1})\sqrt{\frac{r_k}{r_{k+1}}} = \sqrt{T_1}$$

or

$$\kappa(r_k, \xi_k)(r_k - r_{k+1}) \geq \sqrt{\frac{T_1}{2}}.$$

Hence by Lemma 6.2,

$$\int_{r_{k+1}}^{r_k} \kappa(r, e)dr \geq \left(\frac{2}{3}\right)^{3/2} \sqrt{\frac{T_1}{2}}.$$

Denote $C_1 := \sqrt{\frac{27}{4T_1}}$. Then we have

$$\int_{\epsilon}^{r_0} \kappa(r, e)dr \geq \sum_{k=0}^{N-1} \int_{r_{k+1}}^{r_k} \kappa(r, e)dr \geq \sum_{k \in I_1} \int_{r_{k+1}}^{r_k} \kappa(r, e)dr \geq \frac{1}{C_1} N_1.$$

This shows (7.2).

To prove (7.3) let $k \in I_2$. Then

$$\text{either} \quad r_k \kappa(r_k, \xi_k) < 2 \text{ or } \quad \frac{r_k}{r_{k+1}} > 2.$$

In the first case, we get by (4.1)

$$r_k \kappa(r_k, \xi_k) \frac{r_k - r_{k+1}}{\sqrt{r_k r_{k+1}}} = \sqrt{T_1},$$

and solving a quadratic equation

$$\sqrt{\frac{r_k}{r_{k+1}}} = \frac{\sqrt{T_1}}{2r_k \kappa(r_k, \xi_k)} + \sqrt{\frac{T_1}{4r_k^2 \kappa(r_k, \xi_k)^2} + 1} \geq \frac{\sqrt{T_1}}{4} + 1, \tag{7.4}$$

so that

$$\frac{r_k}{r_{k+1}} \geq \left(1 + \frac{\sqrt{T_1}}{4}\right)^2 \geq 1 + \frac{\sqrt{T_1}}{2}.$$

So we have in either case $r_k/r_{k+1} \geq 1 + \sqrt{T_1}/2$. We then obtain the bound

$$\left(1 + \frac{\sqrt{T_1}}{2}\right)^{N_2} \leq \prod_{k \in I_2} \frac{r_k}{r_{k+1}} \leq \prod_{k=0}^{N-1} \frac{r_k}{r_{k+1}} \leq \frac{r_0}{\epsilon},$$

that is

$$N_2 \leq \frac{\ln(r_0/\epsilon)}{\ln(1 + \sqrt{T_1}/2)} \leq \frac{4}{\sqrt{T_1}} \ln \frac{r_0}{\epsilon}.$$

This proves (7.3) by setting $C_2 := 4/\sqrt{T_1}$. Theorem 4.2 is proved.

Now we turn to the lower bound of N in Theorem 4.3. We consider only the case $\nu \geq 4$. The general case can be handled similarly.

For every k, we have

$$r_k\kappa(r_k, e) \geq 4, \qquad \kappa(r_k, \xi_k)(r_{k+1} - r_k) \leq \sqrt{T_1},$$

so that Lemma 6.2 applies with $\xi = \xi_k$, giving

$$\int_{r_{k+1}}^{r_k} \kappa(r, e)dr \leq 3\sqrt{\frac{T_1}{2}} \quad \text{for all } 0 \leq k \leq N.$$

Hence

$$\int_{\epsilon}^{r_0} \kappa(r, e)dr \leq \sum_{k=0}^{N} \int_{r_{k+1}}^{r_k} \kappa(r, e)dr \leq 3\sqrt{\frac{T_1}{2}}(N + 1).$$

This proves (4.4) by setting $C_3 := \frac{\sqrt{2}}{3\sqrt{T_1}}$, and the proof of Theorem 4.3 is completed. □

8. Conclusions. In this paper we have bounded the complexity of path-following methods for linear programming in the form of a curvature integral along the central trajectory. Now we briefly discuss the estimation of the curvature integral.

We observed earlier that $r\kappa(r) \leq \sqrt{n}$. However it may be that $r\kappa(r)$ is much less than $\sqrt{n}$ for some problems. $r\kappa(r)$ will, in effect, tend to zero as r tends to zero, as shown by Witzgall, Boggs, and Domich [12]. Also Mizuno, Todd, and Ye [6] and [7] showed that for any $r > 0$, the expected value of $r\kappa(r) \approx n^{0.25}$, for certain distribution of data of linear programs. So the curvature integral may be expected to be less than $\sqrt{n}\ln(r_0/\epsilon)$, at least for some classes of linear programming problems, since it is the average of the curvature.

The estimation of the curvature integral in the form

$$n^{\alpha} \ln \frac{r_0}{\epsilon} \leq \int_{\epsilon}^{r_0} \kappa(r, e)dr \leq n^{\lambda} \ln \frac{r_0}{\epsilon} \tag{8.1}$$

was started by Sonnevend, Stoer, and Zhao [10], mainly due to Sonnevend, and by Karmarkar [4]. Some smaller λ ($< 1/2$) in the estimation (8.1) has been obtained for a number of classes of linear programs. Also a lower bound of the curvature integral has been found by an artificially constructed linear program, showing $\alpha \geq 1/3$ in (8.1). However, what α and λ are for general linear programming problems is still an open problem.

The above discussion of the curvature integral is in the worst case sense. As for the estimation of the curvature integral in the probability sense, so far there is no literature.

9. Appendix. Variation of curvatures between different trajectories. In this appendix we estimate the difference $|\kappa(r, \xi) - \kappa(r, e)|$. Let $\xi_1 := \xi$ and $\xi_2 := e$, and denote by (x_i, s_i), $i = 1, 2$, the solution of the following equations:

$$(\text{LPD})_{r,\xi_i} \qquad \begin{aligned} x \circ s &= r\xi_i, \\ Ax &= b, \\ A^T y + s &= c, \\ x \geq 0, s &\geq 0. \end{aligned}$$

(Note that (x_i, s_i) in this appendix are not derivatives as in the previous sections.) Similarly $p_i, q_i, \tilde{A}_i$, etc., are referred to as the corresponding quantities for $\xi = \xi_i$, i.e.,

$$\tilde{A}_i := A[\xi_i^{1/2} \circ s_i^{-1}], \qquad H_i := \tilde{A}_i \tilde{A}_i^T, \qquad M_i := \tilde{A}_i^T H_i^{-1} \tilde{A}_i.$$

$$q_i := M_i \xi_i^{1/2}, \qquad p_i := \xi_i^{1/2} - q_i. \tag{A1}$$

Furthermore, we write $\Delta x := x_1 - x_2$, $\Delta s := s_1 - s_2$, $\Delta\xi := \xi_1 - \xi_2$ etc. and $\widetilde{\Delta_i x} := \xi_i^{1/2} \circ x_i^{-1} \circ \Delta x$, $\widetilde{\Delta_i s} := \xi_i^{1/2} \circ s_i^{-1} \circ \Delta s$ for $i = 1, 2$. Because $A\Delta x = 0$ and $\Delta s = -A^T \Delta y$, we have $\Delta x^T \Delta s = 0$ and $\widetilde{\Delta_i x}^T \widetilde{\Delta_i s} = 0$; hence for $i = 1, 2$,

$$\tau_i := \|\widetilde{\Delta_i x} + \widetilde{\Delta_i s}\|_2 = \left(\|\widetilde{\Delta_i x}\|_2^2 + \|\widetilde{\Delta_i s}\|_2^2\right)^{1/2} \geq \max\{\|\widetilde{\Delta_i x}\|_2, \|\widetilde{\Delta_i s}\|_2\}.$$

The following lemma shows that τ_i and $\|\Delta\xi\|_2$ are equivalent: they differ by a factor only depending on $\|\xi_i^{\pm 1}\|_\infty$.

LEMMA A1. *With the above notation, it holds that*

$$\|\Delta\xi\|_2 \leq \|\xi_i\|_\infty^{1/2} \tau_i + \tau_i^2 \qquad \textit{for } i = 1, 2. \tag{A2}$$

If $4\|\xi_i^{-1}\|_\infty \|\Delta\xi\|_2 < 1$ *then also*

$$\tau_i \leq 2\|\xi_i^{-1/2}\|_\infty \|\Delta\xi\|_2 \qquad \textit{for } i = 1, 2. \tag{A3}$$

Proof. We prove the lemma only for $i = 1$. Forming the difference of $(\mathrm{LPD})_{r,\xi_1}$ and $(\mathrm{LPD})_{r,\xi_2}$ it gives

$$s_1 \circ \Delta x + x_1 \circ \Delta s - \Delta x \circ \Delta s = r\Delta\xi$$

or

$$\widetilde{\Delta_1 x} + \widetilde{\Delta_1 s} - \xi_1^{-1/2} \circ \widetilde{\Delta_1 x} \circ \widetilde{\Delta_1 s} = \xi_1^{-1/2} \circ \Delta\xi,$$

which proves (A2) and

$$\tau_1 \leq \|\xi_1^{-1/2}\|_\infty (\|\Delta\xi\|_2 + \tau_1^2).$$

If in addition $4\|\xi_1^{-1}\|_\infty \|\Delta\xi\|_2 < 1$ the last inequality implies

$$\tau_1 \leq \frac{1 - \sqrt{1 - 4\|\xi_1^{-1}\|_\infty \|\Delta\xi\|_2}}{2\|\xi_1^{-1/2}\|_\infty} \tag{$*$}$$

or

$$\tau_1 \geq \frac{1 + \sqrt{1 - 4\|\xi_1^{-1}\|_\infty \|\Delta\xi\|_2}}{2\|\xi_1^{-1/2}\|_\infty}. \tag{$**$}$$

The solution (x, s) of $(\mathrm{LPD})_{r,\xi}$ is continuous with respect to ξ; therefore $\tau_1 = (\|\widetilde{\Delta_1 x}\|_2^2 + \|\widetilde{\Delta_1 s}\|_2^2)^{1/2}$ tends continuously to zero as $\|\Delta\xi\|_2$ tends to zero. Hence $(**)$ must be ruled out. From $(*)$ follows immediately (A3). □

Proof of Lemma 6.1. Let $\xi_1 = \xi$ and $\xi_2 = e$. By (A1) we have

$$\begin{aligned}\|\Delta q\|_2 \le &\|(\Delta\tilde{A})^T H_2^{-1}\tilde{A}_2\xi_2^{1/2}\|_2 + \|\tilde{A}_1^T(\Delta(H^{-1}))\tilde{A}_2\xi_2^{1/2}\|_2 \\ &+ \|\tilde{A}_1^T H_1^{-1}(\Delta\tilde{A})\xi_2^{1/2}\|_2 + \|M_1(\xi_1^{1/2} - \xi_2^{1/2})\|_2.\end{aligned} \tag{A4}$$

Observing $H_i = A[\xi_i \circ s_i^{-2}]A^T$ we have

$$\begin{aligned}&\tilde{A}_1^T(H_1^{-1} - H_2^{-1})\tilde{A}_2\xi_2^{1/2} \\ &\quad = \tilde{A}_1^T H_1^{-1}(H_2 - H_1)H_2^{-1}\tilde{A}_2\xi_2^{1/2} \\ &\quad = M_1[\xi_1^{-1/2} \circ \xi_2^{1/2} \circ s_1 \circ s_2^{-1} - \xi_1^{1/2} \circ \xi_2^{-1/2} \circ s_1^{-1} \circ s_2]q_2 \\ &\quad = M_1[\xi_1^{-1/2} \circ \widetilde{\Delta_2 s} + \xi_2^{-1/2} \circ \widetilde{\Delta_1 s} - \xi_1^{-1/2} \circ \xi_2^{-1/2} \circ \Delta\xi]q_2.\end{aligned}$$

Then by (A3) we get

$$\|\tilde{A}_1^T\Delta(H^{-1})\tilde{A}_2\xi_2^{1/2}\|_2 \le 5\|\xi^{-1/2}\|_\infty\|q_2\|_\infty\|\Delta\xi\|_2.$$

Analogously, but more easily, we can get

$$\|(\Delta\tilde{A})^T H_2^{-1}\tilde{A}_2\xi_2^{1/2}\|_2 \le 3\|\xi^{-1/2}\|_\infty\|q_2\|_\infty\|\Delta\xi\|_2,$$

$$\|\tilde{A}_1^T H_1^{-1}(\Delta\tilde{A})\xi_2^{1/2}\|_2 \le 3\|\xi^{-1/2}\|_\infty\|\Delta\xi\|_2,$$

$$\|M_1(\xi_1^{1/2} - \xi_2^{1/2})\|_2 \le \|\Delta\xi\|_2.$$

Summing up the above inequalities and using $r\kappa(r,e) \ge 4$ and $\|q_2\|_\infty \le \frac{5}{4}r\kappa(r,e)$ (see (3.3)), we have

$$\begin{aligned}\|\Delta q\|_2 &\le (8\|\xi^{-1/2}\|_\infty\|q_2\|_\infty + 3\|\xi^{-1/2}\|_\infty + 1)\|\Delta\xi\|_2 \\ &\le \frac{1}{4}(43\|\xi^{-1/2}\|_\infty + 1)r\kappa(r,e)\|\Delta\xi\|_2.\end{aligned}$$

Because $\|\xi^{-1}\|_\infty \le (1 - \|\xi - e\|_\infty)^{-1} \le 60/59$, we have

$$\|\Delta q\|_2 \le 11.1 \cdot r\kappa(r,e)\|\Delta\xi\|_2.$$

From $p = \xi^{1/2} - q$ we can get an inequality for $\|\Delta p\|_2$ similar to (A4) but with M_1 replaced by N_1 $(= I - M_1)$. Therefore we also have

$$\|\Delta p\|_2 \le 11.1 \cdot r\kappa(r,e)\|\Delta\xi\|_2.$$

Using the above estimations we have

$$\begin{aligned}\|p_1 \circ q_1 - p_2 \circ q_2\|_2 &\le \|p_1 \circ \Delta q\|_2 + \|\Delta p \circ q_1\|_2 + \|\Delta p \circ \Delta q\|_2 \\ &\le 30(r\kappa(r,e))^2\|\Delta\xi\|_2.\end{aligned}$$

The proof is completed by using the notation $\kappa(r,\xi_i)^2 = \|p_i \circ q_i\|_2/r^2$. □

Acknowledgments. The author would like to thank J. Stoer for many valuable suggestions and improvements to this work. Also G. Sonnevend, Fl. Jarre, and J. Mennicken contributed many suggestions and pointed out important literature. The

author also would like to thank the anonymous referees of this paper for their careful reading and suggestions and comments.

REFERENCES

[1] D. A. BAYER AND J. C. LAGRIAS, *The nonlinear geometry of linear programming*, I, II, III, Trans. Amer. Math. Soc., 314 (1989), pp. 499–526, 527–581 and 320 (1990), pp. 193–225.

[2] FL. JARRE, *The method of analytic centers for smooth convex programs*, Ph.D. thesis, Universität Würzburg, Grottenthaler Verlag, Bamberg, Germany, 1989.

[3] N. KARMARKAR, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[4] ———, *Riemannian geometry underlying interior-point methods for linear programming*, in Mathematical Developments Arising from Linear Programming, J. C. Lagarias and M. Todd, eds., Contemporary Mathematics, Vol. 114, American Mathematical Society, Providence, RI, 1990, pp. 51–76.

[5] M. KOJIMA, N. MEGIDDO, T. NOMA, AND A. YOSHISE, *A unified approach to interior point algorithms for linear complementarity problems*, Research report RJ 7493 (70008), IBM Almaden Research Division, San Jose, CA, 1990.

[6] S. MIZUNO, M. J. TODD, AND Y. YE, *Anticipated behavior of path-following algorithm for linear programming*, Tech. Report No. 878, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1989.

[7] ———, *On adaptive-step primal-dual interior-point algorithm for linear programming*, Tech. Report No. 944, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1990. Math. Oper. Res., to appear.

[8] C. ROOS AND I.-PH. VIAL, *A polynomial method of approximate weighted centers for linear programming*, Tech. Report 89-13, Techn. Univ. Delft, Delft, the Netherlands, 1988.

[9] G. SONNEVEND, *An analytic center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, Lecture Notes in Control and Information Sciences, Vol. 84, Springer-Verlag, 1985, pp. 866–876.

[10] G. SONNEVEND, J. STOER, AND G. ZHAO, *On the complexity of following the central path of linear programs by linear extrapolation* II, Math. Programming, 52 (1991), pp. 527–553.

[11] M. TODD, *Recent developments and new directions in linear programming*, Mathematical Programming — Recent Developments and Applications, M. Iri and K. Tanabe, eds., Kluwer Academic Publishers, Boston, 1989, pp. 109–157.

[12] C. WITZGALL, P. T. BOGGS, AND P. D. DOMICH, *On the convergence behavior of trajectories for linear programming*, in Mathematical Developments Arising from Linear Programming Algorithms, J. C. Lagarias and M. J. Todd, eds., Contemporary Mathematics, Vol. 114, American Mathematical Society, Providence, RI, 1991, pp. 161–187.

[13] Y. YE, O. GÜLER, R. A. TAPIA, AND Y. ZHANG, *A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming*, Working Paper No. 91-14, College of Business Administration, The University of Iowa, Iowa City, 1991.

[14] G. ZHAO AND J. STOER, *Estimating the complexity of a class of path-following methods for solving linear programs by curvature integrals*, Appl. Math. Optimization, 27 (1993), pp. 85–103.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 74–95, February 1996
© 1996 Society for Industrial and Applied Mathematics
006

AN EFFICIENT NEWTON BARRIER METHOD FOR MINIMIZING A SUM OF EUCLIDEAN NORMS*

KNUD D. ANDERSEN†

Abstract. A new Newton barrier method is proposed for minimizing a sum of Euclidean norms (MSN), $F(x) = \sum_{i=1}^{\kappa} \left\| A_i^T x - b_i \right\|_2$. MSN is a nonsmooth problem because F is not differentiable at any point x where any of the norms is zero. The method used is based on approximating F with a smooth function, which in the limit has the same optimal value as F. MSN is shown to have a dual problem with properties very similar to duality theory in linear programming. This is used in the development of the method and to give a proof of when an optimal solution for the smooth approximation is ϵ-optimal (measured in the duality gap) for the original problem. An implementation of the algorithm is described for large sparse problems and numerical results are presented for problems with more than 270,000 nonlinear variables. These problems arise from plastic collapse analysis.

Key words. sum of norms, nonsmooth optimization, duality, Newton barrier method

AMS subject classifications. 90C06, 90C25, 90C30

1. Introduction.

Consider the problem of minimizing the function

$$F(\boldsymbol{x}) = \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i(\boldsymbol{x})\|_2 , \tag{1}$$

where

$$\boldsymbol{r}_i(\boldsymbol{x}) = \boldsymbol{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i, \qquad i = 1, \ldots, \kappa \tag{2}$$

with $\boldsymbol{x} \in R^n$, $\boldsymbol{A}_i \in R^{n \times \nu}$, $\boldsymbol{b}_i \in R^{\nu}$, $i = 1, \ldots, \kappa$. In most applications $\nu = 2$ or $\nu = 3$. The function F is convex but is not differentiable at any point $\boldsymbol{x}$ such that $\boldsymbol{r}_i(\boldsymbol{x}) = \boldsymbol{0}$ for some $i \in \{1, \ldots, \kappa\}$. Three classes of methods have been used for the minimizing of (1).

The first class of methods has been presented in [5, 26]. The essential idea of these algorithms is as follows. In each iteration a search direction is computed by Newton's method projected into a linear manifold along which F is locally differentiable. The advantage of this approach is the quadratic convergence and the avoidance of approximation techniques for (1). However, we have found it difficult to use this method due to the dynamic structure of the linear manifold into which the method projects the search direction. The difficulty is that every time terms are added or deleted from the active set, the size and the sparsity structure of the problem changes. Furthermore, in our experience, very high accuracy in the dynamic constraints is required for large problems, and at least one of the applications in which we are interested is quite ill conditioned.

The second class of methods is Weiszfeld's method of iteratively reweighted least squares (IRLS), described in [9, 18, 29]. In every iteration a least squares system is solved where the weights in the linear equations can change from iteration to iteration. However in [9], is it not well defined what to do if norms are close to zero but not

* Received by the editors October 25, 1993; accepted for publication (in revised form) August 22, 1994. This work was supported by a Danish Natural Science Research Council Ph.D. Studentship.

† Department of Mathematics and Computer Science, Odense University, Campusvej 55, 5230 Odense M, Denmark (kda@dash.co.uk).

identically zero. This difficulty for IRLS is also noted in [19], where an interior point method is described for the related L_p approximation problem.

The third class of methods uses smoothing. An example is the hyperbolic approximation procedure (HAP) [10] which finds a solution to the following convex smooth approximation to (1):

$$\sum_{i=1}^{\kappa}(\|\boldsymbol{r}_i(\boldsymbol{x})\|_2^2+\epsilon^2)^{\frac{1}{2}}, \tag{3}$$

where ϵ is a positive scalar. Standard second order (Newton) or first order (gradient) methods for unconstrained minimization can be used to solve (3). The scalar ϵ is reduced in the HAP until it has reached a certain level. However, this class of methods suffers from the lack of a good termination criterion: When is a solution to the smooth problem nearly optimal for the original problem? Note that sparse matrix techniques actually can be used for the solution of (3) (see §4), but to our knowledge this has not been implemented.

In this paper we develop a new smoothing algorithm to solve (1). The essential idea is to treat ϵ as a variable with a positive lower bound in (3). In order to ensure that ϵ stays positive, we add the barrier term $\mu \ln(\epsilon)$ to (3) and solve the approximate problem with Newton's method for a decreasing sequence of the positive barrier parameter μ. Such Newton barrier methods (interior point methods) have been used with great success to solve linear programming (LP) problems; see [23, 21].

The outline of the paper is as follows. In §2 we present a duality theory for (1). In §3, we use duality theory to develop an algorithm to solve (3) and to relate a solution to (3) to the original problem (1). An implementation of the method is described in §4, and finally, numerical results are presented in §5.

Notation. $\boldsymbol{I}_p$ denotes the identity matrix of dimension $p \times p$. Let $\mathcal{K} = \{1, \ldots, \kappa\}$. If P is a finite set, let $|P|$ denote the number of elements in P. The vectors $\boldsymbol{r}$ and $\boldsymbol{y}$ are, respectively, residuals and Lagrange multipliers for the linear system $\boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} = \boldsymbol{b}$, where $\boldsymbol{r}^T = [\boldsymbol{r}_1^T, \ldots, \boldsymbol{r}_\kappa^T] \in R^{\kappa\nu}$, $\boldsymbol{y}^T = [\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_\kappa^T] \in R^{\kappa\nu}$, $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\kappa]$ and $\boldsymbol{b}^T = [\boldsymbol{b}_1^T, \ldots, \boldsymbol{b}_\kappa^T]$. Note that $\boldsymbol{y}_i, \boldsymbol{r}_i \in R^\nu$. The active set (named as in [26]) is defined as $J(\boldsymbol{r}) = \{i \in \mathcal{K} \mid \|\boldsymbol{r}_i\|_2 = 0\}$. $[\boldsymbol{B}]_{ij}$ is the element in $\boldsymbol{B}$ which is in row i and column j. The vector $\boldsymbol{e}$ has all elements equal to 1.

2. Duality theory for MSN. In this section we study duality for (1). Duality gives a measure of optimality for a pair of primal and dual solutions, and therefore makes it possible to develop an efficient algorithm for (1). The use of duality for this type of problem has been studied in [17] for the following general Fermat problem posed in the seventeenth century:

$$\min_{x \in R^2} \sum_{i=1}^{\kappa} \omega_i \|\boldsymbol{x} - \boldsymbol{b}_i\|_2, \tag{4}$$

which is a special case of (1). Duality has furthermore been used in [26, pp. 37–38] and [6] for (1). Define

$$X^p = \{(\boldsymbol{x}, \boldsymbol{r}) \in R^n \times R^{\kappa\nu} \mid \boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} = \boldsymbol{b}\}. \tag{5}$$

Then the problem (1) can be rewritten in the equivalent form

$$(P) \qquad \begin{aligned} &\min \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 \\ &\text{s.t.} \\ &\qquad (\boldsymbol{x}, \boldsymbol{r}) \in X^p, \end{aligned}$$

where the components of $\boldsymbol{r}$ are variables. Note that $\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2$ is a block separable function, and that it must be minimized subject to a set of linear constraints. We will assume that $\boldsymbol{A}$ has full row rank. The problem can otherwise be reduced by a linear transformation.

The Lagrangian function for (P) is

$$L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) = \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 + \boldsymbol{y}^T(\boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b}), \tag{6}$$

where $\boldsymbol{y} \in R^{\kappa\nu}$ is the vector of Lagrange multipliers and $(\boldsymbol{x}, \boldsymbol{r}) \in R^n \times R^{\kappa\nu}$. Define

$$w(\boldsymbol{y}) = \inf\{L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) \mid (\boldsymbol{x}, \boldsymbol{r}) \in R^n \times R^{\kappa\nu}\}.$$

If the minimum of the convex function $L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y})$ is finite, the minimum in $\boldsymbol{x}$ is defined by

$$\frac{\partial L}{\partial \boldsymbol{x}} = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}. \tag{7}$$

In the following lemma we prove that $w(\boldsymbol{y})$ is finite if and only if $\boldsymbol{y}$ belongs to the set

$$Y^d = \{\boldsymbol{y} \in R^{\kappa\nu} \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}, \quad \|\boldsymbol{y}_i\|_2 \leq 1, \quad i = 1, \ldots, \kappa\}. \tag{8}$$

The Lagrangian dual of (P) (see [24, pp. 202–203]) is given by

$$(D_L) \qquad \begin{aligned} &\max_{\boldsymbol{y}} w(\boldsymbol{y}) = \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x},\boldsymbol{r}} L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y})\} \\ &\text{s.t.} \\ &\qquad \boldsymbol{y} \in R^{\kappa\nu}. \end{aligned}$$

The following lemma is a simple extension of [17].

LEMMA 2.1. *There exists a saddle point $(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y})$ for the Lagrange function L in (D_L), and (D_L) is equivalent to the problem*

$$(D) \qquad \begin{aligned} &\max \left(-\boldsymbol{b}^T\boldsymbol{y}\right) \\ &\text{s.t.} \\ &\qquad \boldsymbol{y} \in Y^d. \end{aligned}$$

Proof. Clearly from (6) $w(\boldsymbol{y}) = -\infty$, unless $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}$, in which case we have

$$w(\boldsymbol{y}) = \left(-\boldsymbol{y}^T\boldsymbol{b} + \inf_{\boldsymbol{r}} \left(\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 - \boldsymbol{y}_i^T\boldsymbol{r}_i\right)\right). \tag{9}$$

If there exists $j \in \mathcal{K}$ such that $\|\boldsymbol{y}_j\|_2 > 1$, set $\boldsymbol{r}_j \leftarrow \alpha\boldsymbol{y}_j$. Then

$$\|\boldsymbol{r}_j\|_2 - \boldsymbol{y}_j^T\boldsymbol{r}_j \to -\infty \qquad \text{for } \alpha \to \infty.$$

It follows that the infimum in (9) is finite if and only if

$$\|\boldsymbol{y}_i\|_2 \leq 1 \quad \forall i \in \mathcal{K},$$

in which case Cauchy–Schwarz gives

$$\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 - \boldsymbol{y}_i^T \boldsymbol{r}_i \geq \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 - \|\boldsymbol{r}_i\|_2 \|\boldsymbol{y}_i\|_2 = \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 (1 - \|\boldsymbol{y}_i\|_2) \geq 0 \tag{10}$$

and

$$\min_{r} \left(\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 - \boldsymbol{y}_i^T \boldsymbol{r}_i \right) = 0.$$

This completes the proof. □

Note that neither X^p nor Y^d is empty. Using this fact and the above lemma we state a duality theorem for (P) and (D). Let $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ be an optimal solution to (P), and let $J = J(\boldsymbol{r}^*)$. We define $\widehat{\boldsymbol{A}} = [\boldsymbol{A}_{j_1}, \ldots, \boldsymbol{A}_{j_t}]$ where $J = [j_1, \ldots, j_t]$. In the following it is assumed that $\widehat{\boldsymbol{A}}$ has full column rank. This assumption can be removed; see Appendix B.

THEOREM 2.2. *Let $(\boldsymbol{x}, \boldsymbol{r}) \in X^p$, $\boldsymbol{y} \in Y^d$ be feasible and let $(\boldsymbol{x}^*, \boldsymbol{r}^*) \in X^p$, $\boldsymbol{y}^* \in Y^d$ be optimal solutions to (P) and (D), respectively. Then*

$$\text{(a)} \qquad -\boldsymbol{b}^T \boldsymbol{y} \leq \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 \qquad \textit{weak duality}$$

and

$$\text{(b)} \qquad -\boldsymbol{b}^T \boldsymbol{y}^* = \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i^*\|_2 \qquad \textit{strong duality.}$$

Proof. (a) Let $(\boldsymbol{x}, \boldsymbol{r}) \in X^p$ and $\boldsymbol{y} \in Y^d$. By the Cauchy–Schwarz inequality

$$-\boldsymbol{b}^T \boldsymbol{y} = \boldsymbol{r}^T \boldsymbol{y} - \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y} = \sum_{i=1}^{\kappa} \boldsymbol{r}_i^T \boldsymbol{y}_i \leq \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 .$$

(b) Let $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ be an optimal solution to (P), and consider the problem

$$\begin{aligned} \min \quad & \sum_{i \in (\mathcal{K} \setminus J)} \|\boldsymbol{r}_i\|_2 \\ \text{s.t.} \quad & \\ & (\boldsymbol{x}, \boldsymbol{r}) \in X^p, \\ & \boldsymbol{r}_i = \boldsymbol{0} \quad \forall i \in J. \end{aligned} \tag{11}$$

Note that $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ is necessarily optimal for (11). The Lagrangian function for (11) is given by

$$L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) = \sum_{i \in (\mathcal{K} \setminus J)} \|\boldsymbol{r}_i\|_2 + \sum_{i \in (\mathcal{K} \setminus J)} \boldsymbol{y}_i^T (\boldsymbol{A}_i^T \boldsymbol{x} - \boldsymbol{r}_i - \boldsymbol{b}_i) + \sum_{i \in J} \boldsymbol{y}_i^T (\boldsymbol{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i). \tag{12}$$

The convex function $L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y})$ is differentiable and optimal in $(\boldsymbol{x}^*, \boldsymbol{r}^*)$. It follows from the necessary Karush–Kuhn–Tucker (KKT) condition that there exist numbers $\boldsymbol{y}^x \in R^{\kappa\nu}$ such that

$$\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{r}^*)}{\partial \boldsymbol{x}} = \boldsymbol{A}\boldsymbol{y}^x = \boldsymbol{0}, \tag{13}$$

$$\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{r}^*)}{\partial \boldsymbol{r}_i} = \frac{1}{\|\boldsymbol{r}_i^*\|_2}\boldsymbol{r}_i^* - \boldsymbol{y}_i^x = \boldsymbol{0}, \quad i \in (\mathcal{K} \setminus J). \tag{14}$$

Using the assumption that $\widehat{\boldsymbol{A}}$ is of full column rank, it follows that $\boldsymbol{y}^x$ is uniquely defined by (13) and (14). It is now shown that $\boldsymbol{y}^x \in Y^d$. We have from (14)

$$\boldsymbol{y}_i^x = \frac{1}{\|\boldsymbol{r}_i^*\|_2}\boldsymbol{r}_i^* \quad \Rightarrow \|\boldsymbol{y}_i^x\|_2 = 1 \qquad \text{for} \quad i \in (\mathcal{K} \setminus J).$$

Now assume that $\|\boldsymbol{y}_j^x\|_2 > 1$ for some $j \in J$. Let Z denote the set of feasible directions

$$(\Delta\boldsymbol{r}, \Delta\boldsymbol{x}) \in Z = \{\Delta\boldsymbol{r} \in R^{\kappa\nu}, \Delta\boldsymbol{x} \in R^n \mid \Delta\boldsymbol{r} = \boldsymbol{A}^T\Delta\boldsymbol{x}\}.$$

The total directional derivative of $\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2$ at $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ for any direction $(\Delta\boldsymbol{r}, \Delta\boldsymbol{x}) \in Z$ is found in [26, p. 38] to be

$$\begin{aligned}
&\sum_{i\in(\mathcal{K}\setminus J)} \frac{1}{\|\boldsymbol{r}_i^*\|_2}(\boldsymbol{r}_i^*)^T\Delta\boldsymbol{r}_i + \sum_{i\in J}\|\Delta\boldsymbol{r}_i\|_2 \\
&= \sum_{i\in(\mathcal{K}\setminus J)} (\boldsymbol{y}_i^x\boldsymbol{A}_i)^T\Delta\boldsymbol{x} + \sum_{i\in J}\|\Delta\boldsymbol{r}_i\|_2 \\
&= -\sum_{i\in J}(\boldsymbol{y}_i^x\boldsymbol{A}_i)^T\Delta\boldsymbol{x} + \sum_{i\in J}\|\Delta\boldsymbol{r}_i\|_2 \\
&= \sum_{i\in J}(-(\boldsymbol{y}_i^x)^T\Delta\boldsymbol{r}_i + \|\Delta\boldsymbol{r}_i\|_2) \\
&\geq \sum_{i\in J}\|\Delta\boldsymbol{r}_i\|_2\,(1 - \|\boldsymbol{y}_i^x\|_2),
\end{aligned}$$

where we have used $\boldsymbol{0} = \boldsymbol{A}\boldsymbol{y}^x = \sum_{i=1}^{\kappa}\boldsymbol{A}_i\boldsymbol{y}_i^x$ and $\Delta\boldsymbol{r}_i = \boldsymbol{A}_i^T\Delta\boldsymbol{x}$ for $(\Delta\boldsymbol{x}, \Delta\boldsymbol{r}) \in Z$. Now choose

$$\begin{aligned}
&\Delta\boldsymbol{r}_i = \boldsymbol{0} \quad \forall i \neq j, i \in J, \\
&\Delta\boldsymbol{r}_j = -\boldsymbol{y}_j^x.
\end{aligned}$$

Since $\widehat{\boldsymbol{A}}$ has full column rank there exists $\Delta\boldsymbol{x}$, such that $\Delta\boldsymbol{r}_i = \boldsymbol{A}_i^T\Delta\boldsymbol{x}$ for $i \in J$, and if we put $\Delta\boldsymbol{r}_i = \boldsymbol{A}_i^T\Delta\boldsymbol{x}$ for $i \in \mathcal{K} \setminus J$, then $(\Delta\boldsymbol{x}, \Delta\boldsymbol{r}) \in Z$. Since $\|\boldsymbol{y}_j^x\|_2 > 1$ the total derivative is not zero, which contradicts the optimality of $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ for (P). It follows that $\boldsymbol{y}^x \in Y^d$. Finally we observe that

$$-\boldsymbol{b}^T\boldsymbol{y}^x = (\boldsymbol{r}^* - \boldsymbol{A}^T\boldsymbol{x}^*)^T\boldsymbol{y}^x = \sum_{i=1}^{\kappa}\boldsymbol{r}_i^{*T}\boldsymbol{y}_i^x = \sum_{i\in J}\|\boldsymbol{r}_i^*\|_2 = \sum_{i=1}^{\kappa}\|\boldsymbol{r}_i^*\|_2,$$

proving that strong duality holds. □

The above theorem shows that if $(\boldsymbol{x}, \boldsymbol{r})$ is primal feasible and $\boldsymbol{y}$ is dual feasible, then the duality gap $(\sum_{i=1}^{\kappa}\|\boldsymbol{r}_i\|_2 + \boldsymbol{b}^T\boldsymbol{y})$ is positive and they are optimal if and only if the duality gap is zero.

In order to link optimal solutions of the perturbed problem (3) to the problem (P), we introduce the following measure of optimality for (P).

DEFINITION 1. *A solution* $(\boldsymbol{x}, \boldsymbol{r}) \in X^p$ *and* $\boldsymbol{y} \in Y^d$ *is called* ζ*-optimal to* (P) *and* (D) *if*

$$\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 + \boldsymbol{b}^T \boldsymbol{y} \leq \zeta.$$

This gives a direct link between solutions to the original problem and its smooth approximation. It is then very important to develop an algorithm that also generates dual solutions which approach a feasible optimal solution for (D).

As already noted, we do not want to develop an algorithm that works directly on (P), but instead we wish to make use of a smooth approximation:

$$(P_\epsilon) \qquad \begin{aligned} &\min \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2 \right)^{\frac{1}{2}} \\ &\text{s.t.} \\ &\qquad (\boldsymbol{x}, \boldsymbol{r}) \in X^p. \end{aligned}$$

The Lagrangian for (P_ϵ) is given by

$$L_\epsilon(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) = \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2 \right)^{\frac{1}{2}} + \boldsymbol{y}^T (\boldsymbol{A}^T \boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b}).$$

An equivalent form of (P_ϵ) is

$$\begin{aligned} &\min \sum_{i=1}^{\kappa} \left\| \begin{array}{c} \boldsymbol{r}_i \\ (\boldsymbol{r}_\epsilon)_i \end{array} \right\|_2 \\ &\text{s.t.} \\ &\qquad \begin{bmatrix} \boldsymbol{A}^T \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{r}_\epsilon \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \epsilon \boldsymbol{e} \end{bmatrix}, \end{aligned} \tag{15}$$

where $\boldsymbol{r}_\epsilon \in R^\kappa$. It then follows from Lemma 2.1 applied to (15) that (P_ϵ) has the following dual problem:

$$\begin{aligned} &\max \left(-\boldsymbol{b}^T \boldsymbol{y} + \epsilon \sum_{i=1}^{\kappa} s_i \right) \\ &\text{s.t.} \\ &\qquad \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}, \\ &\qquad \|\boldsymbol{y}_i\|_2^2 + s_i^2 \leq 1, \qquad i = 1, \ldots, \kappa, \end{aligned} \tag{16}$$

where $\boldsymbol{s} \in R^\kappa$. This dual problem can we rewritten

$$(D_\epsilon) \qquad \begin{aligned} &\max \left(-\boldsymbol{b}^T \boldsymbol{y} + \epsilon \sum_{i=1}^{\kappa} (1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} \right) \\ &\text{s.t.} \\ &\qquad \boldsymbol{y} \in Y^d. \end{aligned} \tag{17}$$

Thus the introduction of ϵ in (P) has the same effect as introducing a cost on the dual slacks $(1 - \|\boldsymbol{y}_i\|_2^2)^{1/2}$ in (D).

The function $\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{1/2}$ is both differentiable and convex. A necessary condition for $(\boldsymbol{x},\boldsymbol{r})\in X^p$ to be optimal for (P_ϵ) is that there exist $\boldsymbol{y}\in R^{\kappa\nu}$ such that

$$\frac{\partial L(\boldsymbol{x},\boldsymbol{r},\boldsymbol{y})}{\partial \boldsymbol{x}}=\boldsymbol{A}\boldsymbol{y}=\boldsymbol{0}, \tag{18}$$

$$\frac{\partial L(\boldsymbol{x},\boldsymbol{r},\boldsymbol{y})}{\partial \boldsymbol{r}_i}=\frac{1}{(\|\boldsymbol{r}_i\|_2^2+\epsilon^2)^{\frac{1}{2}}}\boldsymbol{r}_i-\boldsymbol{y}_i=\boldsymbol{0}. \tag{19}$$

These are the well-known KKT conditions. We can now state a duality theorem for (D_ϵ) and (P_ϵ)

THEOREM 2.3. *Let $(\boldsymbol{x},\boldsymbol{r})\in X^p$, $\boldsymbol{y}\in Y^d$ be feasible and let $(\boldsymbol{x}^*,\boldsymbol{r}^*)\in X^p$ and $\boldsymbol{y}^*\in Y^d$ be optimal solutions to, respectively, (P_ϵ) and (D_ϵ). Then*

$$\text{(a)}\qquad -\boldsymbol{b}^T\boldsymbol{y}+\epsilon\sum_{i=1}^{\kappa}(1-\|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\le\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{1}{2}}$$

and

$$\text{(b)}\qquad -\boldsymbol{b}^T\boldsymbol{y}^*+\epsilon\sum_{i=1}^{\kappa}(1-\|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}}=\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i^*\|_2^2+\epsilon^2\right)^{\frac{1}{2}}.$$

Proof. (a) Let $(\boldsymbol{x},\boldsymbol{r})\in X^p$ and $\boldsymbol{y}\in Y^d$. Then

$$\begin{aligned}
-\boldsymbol{b}^T\boldsymbol{y}+\epsilon\sum_{i=1}^{\kappa}(1-\|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} &= (\boldsymbol{r}^T-\boldsymbol{x}\boldsymbol{A})\boldsymbol{y}+\epsilon\sum_{i=1}^{\kappa}(1-\|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\\
&=\sum_{i=1}^{\kappa}\begin{bmatrix}\boldsymbol{r}_i\\ \epsilon\end{bmatrix}^T\begin{bmatrix}\boldsymbol{y}_i\\ (1-\|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\end{bmatrix}\\
&\le\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{1}{2}}(\|\boldsymbol{y}_i\|_2^2+(1-\|\boldsymbol{y}_i\|_2^2))^{\frac{1}{2}}\\
&=\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{1}{2}}.
\end{aligned}$$

(b) Let $(\boldsymbol{x}^*,\boldsymbol{r}^*)$ be an optimal solution to (P_ϵ) and set

$$\boldsymbol{y}_i^*=\frac{1}{\left(\|\boldsymbol{r}_i^*\|_2^2+\epsilon^2\right)^{\frac{1}{2}}}\boldsymbol{r}_i^*\qquad\text{for } i=1,\dots,\kappa. \tag{20}$$

It follows that $\boldsymbol{y}^*\in Y^d$ and that

$$\begin{aligned}
-\boldsymbol{b}^T\boldsymbol{y}^*+\epsilon\sum_{i=1}^{\kappa}(1-\|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}} &=\sum_{i=1}^{\kappa}\left(\boldsymbol{r}_i^{*T}\boldsymbol{y}_i^*+\epsilon(1-\|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}}\right)\\
&=\sum_{i=1}^{\kappa}\left(\frac{\|\boldsymbol{r}_i^*\|_2^2}{(\|\boldsymbol{r}_i^*\|_2^2+\epsilon^2)^{\frac{1}{2}}}+\frac{\epsilon^2}{(\|\boldsymbol{r}_i^*\|_2^2+\epsilon^2)^{\frac{1}{2}}}\right)\\
&=\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i^*\|_2^2+\epsilon^2\right)^{\frac{1}{2}}. \qquad \square
\end{aligned}$$

THEOREM 2.4. *Let $(\boldsymbol{x}^*,\boldsymbol{r}^*)$ be an optimal solution to (P_ϵ) and $\boldsymbol{y}^*$ an optimal solution to (D_ϵ). Then $(\boldsymbol{x}^*,\boldsymbol{r}^*,\boldsymbol{y}^*)$ is a $(\kappa\epsilon)$-optimal solution to (P) and (D).*

Proof. Let the optimal solutions $(\boldsymbol{x}^*, \boldsymbol{r}^*) \in X^p$ and $\boldsymbol{y}^* \in Y^d$ be given. From optimality it follows that

$$\sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon^2 \right)^{\frac{1}{2}} + \boldsymbol{b}^T \boldsymbol{y}^* - \epsilon \sum_{i=1}^{\kappa} (1 - \|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}} = 0,$$

which implies

$$\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i^*\|_2 + \boldsymbol{b}^T \boldsymbol{y}^* = \epsilon \sum_{i=1}^{\kappa} (1 - \|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}} \leq \kappa\epsilon. \qquad \square$$

In order to obtain a ζ-optimal solution, ϵ must be chosen such that $\epsilon \leq \frac{\zeta}{\kappa}$. This is a measure for how small ϵ must be to ensure a given optimality. We can even give a better approximative bound for ϵ: For an optimal point $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ to (P_ϵ) with $\boldsymbol{y}^*$ given by (20), we have for any $c \geq 1$

$$\|\boldsymbol{y}_i^*\|_2 = \frac{\|\boldsymbol{r}_i^*\|_2}{\left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon^2 \right)^{\frac{1}{2}}} \geq \frac{c}{\sqrt{c^2+1}} \quad \text{for } i \in N = \{ j \in \mathcal{K} \mid \epsilon c \ll \|\boldsymbol{r}_j^*\|_2 \}.$$

With $c = \sqrt{|N| - 1}$ we get

$$\begin{aligned} \left| \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon^2 \right)^{\frac{1}{2}} + \boldsymbol{b}^T \boldsymbol{y}^* \right| &= \epsilon \sum_{i=1}^{\kappa} (1 - \|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}} \\ &\leq \epsilon \sum_{i \in (\mathcal{K} \setminus N)} (1 - \|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}} + \epsilon \sum_{i \in N} \left(1 - \frac{c^2}{1+c^2} \right)^{\frac{1}{2}} \\ &\leq \epsilon \sum_{i \in (\mathcal{K} \setminus N)} |\mathcal{K} \setminus N| + \epsilon |N| (1 + c^2)^{-\frac{1}{2}} \\ &\leq |\mathcal{K} \setminus N| \epsilon + \epsilon. \end{aligned} \tag{21}$$

The approximative bound for ϵ to ensure a ζ-optimal solution is $\epsilon \leq \frac{\zeta}{|\mathcal{K} \setminus N| + 1}$. If ϵ is small enough and $\boldsymbol{r}_i^*$ is an optimal solution to (P) then $N = \{ j \in \mathcal{K} \mid \|\boldsymbol{r}_i^*\|_2 > 0 \}$. It follows that $|\mathcal{K} \setminus N|$ is the number of terms which are nondifferentiable at the optimal solution.

3. A Newton barrier method for MSN. In this section we will propose a Newton barrier method for MSN. The algorithm reduces the smooth parameter ϵ in such a way that convergence to an optimal solution takes place in a small number of iterations. The idea is to treat the smoothing parameter ϵ as a variable instead of a constant as long as it is larger than a given small tolerance ϵ_l. It is known from §2 that ϵ_l can be chosen such that it ensures a given optimality. We consider the following modification of (P_ϵ):

$$\begin{aligned} &\min \sum_{i=1}^{\kappa} (\|\boldsymbol{r}_i\|_2^2 + \epsilon^2)^{\frac{1}{2}} \\ &\text{s.t.} \\ &\quad (\boldsymbol{x}, \boldsymbol{r}) \in X^p, \\ &\quad \epsilon \geq \epsilon_l. \end{aligned} \tag{22}$$

The purpose of barrier methods is to keep the iterates away from the boundary of the feasible region. Many barrier functions have been suggested [11] but in this paper only the logarithmic barrier function is considered. First suggested in [13], the log barrier function has been used with great success for LP during the past few years [16, 23]. The simple bound in (22) is removed and the term $\mu \ln(\epsilon - \epsilon_l)$ is subtracted from the objective function. The parameter μ is a positive scalar acting as a weight on the barrier term. In fact, we may put $\epsilon_l = 0$ in the barrier term $\ln(\epsilon - \epsilon_l)$. The barrier subproblem to be solved is then

$$\begin{aligned} &\min F_\mu(\boldsymbol{r}, \epsilon) = \sum_{i=1}^{\kappa} (\|\boldsymbol{r}_i\|_2^2 + \epsilon^2)^{\frac{1}{2}} - \mu \ln \epsilon \\ &\text{s.t.} \\ &\qquad (\boldsymbol{x}, \boldsymbol{r}) \in X^p. \end{aligned} \tag{23}$$

The Lagrangian for (23) is given by

$$L(\boldsymbol{x}, \boldsymbol{r}, \epsilon, \boldsymbol{y}) = F_\mu(\boldsymbol{r}, \epsilon) + \boldsymbol{y}^T(\boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b}). \tag{24}$$

The necessary first order (KKT) conditions for $(\boldsymbol{x}, \boldsymbol{r}, \epsilon, \boldsymbol{y})$ to be optimal for (24) are

$$\frac{\partial L}{\partial \boldsymbol{r}_i} = [\boldsymbol{g}_r]_i - \boldsymbol{y}_i = \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} \boldsymbol{r}_i - \boldsymbol{y}_i = \boldsymbol{0}, \quad \text{for } i = 1, \ldots, \kappa, \tag{25}$$

$$\frac{\partial L}{\partial \boldsymbol{x}} = \boldsymbol{g}_x = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}, \tag{26}$$

$$\frac{\partial L}{\partial \epsilon} = g_\epsilon = \epsilon \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} - \mu\epsilon^{-1} = 0. \tag{27}$$

For a fixed optimal value ϵ the first order conditions (25) and (26) are the same as (18) and (19) implying that the dual problems to (22) and (23) have the same optimal solutions. The Hessian for L is given by

$$\frac{\partial^2 L}{\partial \boldsymbol{r}_i \partial \boldsymbol{r}_j} = [\boldsymbol{H}_r]_{ij} = \begin{cases} \dfrac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} \boldsymbol{I}_\nu - \dfrac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{3}{2}}} \boldsymbol{r}_i \boldsymbol{r}_i^T & \text{for } j = i, \\ \boldsymbol{O}_\nu & \text{for } j \neq i, \end{cases} \tag{28}$$

$$\frac{\partial^2 L}{\partial \boldsymbol{r}_i \partial \epsilon} = [\boldsymbol{H}_{r\epsilon}]_i = \frac{-\epsilon}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{3}{2}}} \boldsymbol{r}_i \quad \text{for } i = 1, \ldots, \kappa, \tag{29}$$

$$\frac{\partial^2 L}{\partial \epsilon^2} = \boldsymbol{H}_\epsilon = \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} - \epsilon^2 \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{3}{2}}} + \mu\epsilon^{-2}. \tag{30}$$

We use a *feasible-point descent method* which is a standard approach [16, p. 186] for finding a solution of (23). Let $Z = \{(\Delta\boldsymbol{x}, \Delta\boldsymbol{r}) \mid \boldsymbol{A}^T\Delta\boldsymbol{x} - \Delta\boldsymbol{r} = \boldsymbol{0}\}$, $(\boldsymbol{x}^0, \boldsymbol{r}^0) \in X^p$ and $\epsilon^0 > 0$. The next iterate $(\boldsymbol{x}^1, \boldsymbol{r}^1, \epsilon^1)$ is defined as

$$\boldsymbol{x}^1 = \boldsymbol{x}^0 + \alpha\Delta\boldsymbol{x}, \qquad \boldsymbol{r}^1 = \boldsymbol{A}^T\boldsymbol{x}^1 - \boldsymbol{b}, \qquad \epsilon^1 = \epsilon^0 + \alpha\Delta\epsilon,$$

where the direction $(\Delta\boldsymbol{x}, \Delta\boldsymbol{r}) \in Z$, and α is a positive scalar. The computation of the direction $p = (\Delta\boldsymbol{x}, \Delta\boldsymbol{r}, \Delta\epsilon)$ and the steplength α must ensure $F_\mu(\boldsymbol{r}^1, \epsilon^1) < F_\mu(\boldsymbol{r}^0, \epsilon^0)$ and $\epsilon^1 > 0$.

We use the Newton search direction $p = (\Delta \boldsymbol{x}, \Delta \boldsymbol{r}, \Delta\epsilon)$, which is the solution to the following quadratic program:

$$\begin{aligned} &\min F_Q = \boldsymbol{g}_r^T \Delta \boldsymbol{r} + g_\epsilon^T \Delta\epsilon + \tfrac{1}{2}\Delta \boldsymbol{r}^T \boldsymbol{H}_r \Delta \boldsymbol{r} + \tfrac{1}{2}\Delta\epsilon \boldsymbol{H}_\epsilon \Delta\epsilon + \Delta \boldsymbol{r}^T \boldsymbol{H}_{r\epsilon} \Delta\epsilon \\ &\text{s.t.} \\ &\quad \boldsymbol{A}^T \Delta \boldsymbol{x} = \Delta \boldsymbol{r}. \end{aligned} \tag{31}$$

The Lagrangian for (31) is given by

$$L_Q(\Delta \boldsymbol{x}, \Delta \boldsymbol{r}, \Delta\epsilon, \boldsymbol{y}) = F_Q(\Delta \boldsymbol{r}, \Delta\epsilon) + \boldsymbol{y}^T(\boldsymbol{A}^T \Delta \boldsymbol{x} - \Delta \boldsymbol{r}). \tag{32}$$

The necessary first order (KKT) conditions for $(\Delta \boldsymbol{x}, \Delta \boldsymbol{r}, \Delta\epsilon, \boldsymbol{y})$ to be optimal for (32) are

$$\frac{\partial L_Q}{\partial \Delta \boldsymbol{r}} = \boldsymbol{g}_r + \boldsymbol{H}_r \Delta \boldsymbol{r} + \boldsymbol{H}_{r\epsilon} \Delta\epsilon - \boldsymbol{y} = \boldsymbol{0}, \tag{33}$$

$$\frac{\partial L_Q}{\partial \Delta \boldsymbol{x}} = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}, \tag{34}$$

$$\frac{\partial L_Q}{\partial \Delta\epsilon} = g_\epsilon + \boldsymbol{H}_\epsilon \Delta\epsilon + \boldsymbol{H}_{r\epsilon}^T \Delta \boldsymbol{r} = 0, \tag{35}$$

$$\frac{\partial L_Q}{\partial \boldsymbol{y}} = \boldsymbol{A}^T \Delta \boldsymbol{x} - \Delta \boldsymbol{r} = \boldsymbol{0}. \tag{36}$$

The Lagrange multipliers $\boldsymbol{y}$ converge towards the Lagrange multipliers for the constraints in the original problem (23). Using (33)–(36), we obtain

$$\boldsymbol{A}(\boldsymbol{H}_r - \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}\boldsymbol{H}_{r\epsilon}^T)\boldsymbol{A}^T \Delta \boldsymbol{x} = -\boldsymbol{A}\boldsymbol{g}_r + \boldsymbol{A}\boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1} g_\epsilon, \tag{37}$$

$$\Delta \boldsymbol{r} = \boldsymbol{A}^T \Delta \boldsymbol{x}, \tag{38}$$

$$\Delta\epsilon = -\boldsymbol{H}_\epsilon^{-1}(g_\epsilon + \boldsymbol{H}_{r\epsilon}^T \Delta \boldsymbol{r}), \tag{39}$$

$$\boldsymbol{y} = \boldsymbol{g}_r + \boldsymbol{H}_r \Delta \boldsymbol{r} + \boldsymbol{H}_{r\epsilon} \Delta\epsilon. \tag{40}$$

Using (33)–(36) we may eliminate $\Delta \boldsymbol{r}$ and $\Delta\epsilon$ and compute the following analogue of the Hessian $\boldsymbol{H} = \boldsymbol{H}_r - \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}\boldsymbol{H}_{r\epsilon}^T$. $\boldsymbol{H}$ is a block diagonal matrix where every block is of size $\nu \times \nu$. This is due to the block separability of F_μ, and it follows that $\boldsymbol{H}$ is very sparse if $\nu \ll m$. It is assumed that $\boldsymbol{H}_\epsilon \neq 0$ which is always the case if $\mu > 0$:

$$\begin{aligned} \boldsymbol{H}_\epsilon &= \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} - \epsilon^2 \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{3}{2}}} + \mu\epsilon^{-2} \\ &= \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} \left(1 - \frac{\epsilon^2}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)}\right) + \mu\epsilon^{-2} > 0. \end{aligned}$$

We now show that $\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T$ is a positive definite matrix:

$$\boldsymbol{x}^T H \boldsymbol{x} = \sum_{i=1}^{\kappa} \boldsymbol{x}_i^T \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} \left(\boldsymbol{I}_\nu - \frac{\boldsymbol{r}_i \boldsymbol{r}_i^T}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)} - \boldsymbol{H}_\epsilon^{-1} \frac{\epsilon^2 \boldsymbol{r}_i \boldsymbol{r}_i^T}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{5}{2}}}\right) \boldsymbol{x}_i$$

$$\geq \sum_{i=1}^{\kappa} \frac{\|\boldsymbol{x}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{1}{2}}}\left(1-\frac{\|\boldsymbol{r}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)}-\boldsymbol{H}_\epsilon^{-1}\frac{\epsilon^2\|\boldsymbol{r}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{5}{2}}}\right)$$

$$\geq \sum_{i\in J}\frac{\|\boldsymbol{x}_i\|_2^2}{\epsilon} \tag{41}$$

$$+\sum_{i\in(\mathcal{K}\setminus J)}\frac{\|\boldsymbol{x}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{1}{2}}}\left(1-\frac{\|\boldsymbol{r}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)}-\frac{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{3}{2}}}{\|\boldsymbol{r}_i\|_2^2}\frac{\epsilon^2\|\boldsymbol{r}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{5}{2}}}\right)$$

$$\geq 0, \tag{42}$$

where the first inequality follows from the Cauchy–Schwarz inequality and the second follows from

$$\boldsymbol{H}_\epsilon=\sum_{i=1}^{\kappa}\frac{\|\boldsymbol{r}_i\|_2^2}{\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{3}{2}}}+\mu\epsilon^{-2}>\frac{\|\boldsymbol{r}_j\|_2^2}{\left(\|\boldsymbol{r}_j\|_2^2+\epsilon^2\right)^{\frac{3}{2}}}\geq 0\quad\forall j\in\mathcal{K}. \tag{43}$$

For $\boldsymbol{x}\neq\boldsymbol{0}$ (42) holds with strong inequality except if J is empty, in which case (41) holds with strong inequality. From the assumption that $\boldsymbol{A}$ has full row rank it follows that

$$\boldsymbol{x}^T\left(\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T\right)\boldsymbol{x}>0\quad\text{if }\boldsymbol{x}\neq\boldsymbol{0}. \tag{44}$$

Hence $\Delta\boldsymbol{x}$ is uniquely defined by (37). Ignoring second order terms, changes in F_μ are given by

$$\begin{aligned}\boldsymbol{g}_r^T\Delta\boldsymbol{r}+\boldsymbol{g}_\epsilon^T\Delta\epsilon=&-(\boldsymbol{g}_r-\boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}\boldsymbol{g}_\epsilon)^T\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T)^{-1}\boldsymbol{A}(\boldsymbol{g}_r-\boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}\boldsymbol{g}_\epsilon)\\&-\boldsymbol{g}_\epsilon^T\boldsymbol{H}_\epsilon^{-1}\boldsymbol{g}_\epsilon\leq 0.\end{aligned}$$

Thus $(\Delta\boldsymbol{x},\Delta\boldsymbol{r},\Delta\epsilon)$ is a descent direction of F_μ except if $g_\epsilon=0$ and $\boldsymbol{A}\boldsymbol{g}_r=\boldsymbol{0}$.

An algorithm BARSUB for the solution of the barrier subproblem (23) is given in Fig. 1.

3.1. The central path and reduction of the barrier parameter μ. We can find a solution to (23) for a given value of μ with BARSUB. Subsequently the parameter μ must be reduced, until ϵ is at its lower bound ϵ_l or a near optimal solution $(\boldsymbol{x},\boldsymbol{r},\boldsymbol{y})$ has been found for (P) and (D). One approach would be to accurately solve a sequence of problems, each for a different value of μ, with $\mu\to 0$. However, we may save computation by reducing μ at every iteration, meaning that the barrier problem is redefined for each iteration of BARSUB. Let $w(\mu)\equiv(\boldsymbol{r}^*(\mu),\boldsymbol{x}^*(\mu),\boldsymbol{y}^*(\mu),\epsilon^*(\mu))$ be the optimal values of the barrier subproblem for a given value of μ. The optimal values $(\boldsymbol{r}^*(\mu),\boldsymbol{x}^*(\mu),\boldsymbol{y}^*(\mu),\epsilon^*(\mu))$ are uniquely defined by the first order conditions (25)–(27) for (23). This follows from the fact that F_μ is strictly convex (the Hessian for F_μ is positive definite) and that $\boldsymbol{A}$ by assumption has full row rank. Furthermore, F_μ always has a finite optimal solution for $\mu>0$, as follows from the inequality

$$F_\mu(\boldsymbol{r},\epsilon)=\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2+\epsilon^2\right)^{\frac{1}{2}}-\mu\ln(\epsilon)\geq\kappa\epsilon-\mu\ln(\epsilon)\geq\mu\left(1-\ln\left(\frac{\mu}{\kappa}\right)\right).$$

Procedure BARSUB $(\boldsymbol{A}, \boldsymbol{b}, (\boldsymbol{x}^0, \epsilon^0)$, stopping criterion, $\mu, \lambda)$
  $k \leftarrow 0$
  repeat
    $\boldsymbol{r}^k \leftarrow \boldsymbol{A}^T\boldsymbol{x}^k - \boldsymbol{b}$.
    $\Delta\boldsymbol{x} \leftarrow (\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T)^{-1}(-\boldsymbol{A}\boldsymbol{g}_r + \boldsymbol{A}\boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}g_\epsilon)$.
    $\Delta\boldsymbol{r} \leftarrow \boldsymbol{A}^T\Delta\boldsymbol{x}$.
    $\Delta\epsilon \leftarrow -\boldsymbol{H}_\epsilon^{-1}(g_\epsilon + \boldsymbol{H}_{r\epsilon}^T\Delta\boldsymbol{r})$.
    $\boldsymbol{y} \leftarrow \boldsymbol{g}_r + \boldsymbol{H}_r\Delta\boldsymbol{r} + \boldsymbol{H}_{r\epsilon}\Delta\epsilon$.
    if $\Delta\epsilon < 0$ then $\alpha_{\max} \leftarrow -\lambda\frac{\epsilon^k}{\Delta\epsilon}$ else $\alpha_{\max} \leftarrow \infty$.
    Linesearch (see next section): Find α such that:
      $$F_\mu(\boldsymbol{r}^k + \alpha\Delta\boldsymbol{r}, \epsilon^k + \alpha\Delta\epsilon) \approx \min_{0<\theta\le\alpha_{\max}} F_\mu(\boldsymbol{r}^k + \theta\Delta\boldsymbol{r}, \epsilon^k + \theta\Delta\epsilon).$$
    $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \alpha\Delta\boldsymbol{x}$.
    $\epsilon^{k+1} \leftarrow \epsilon^k + \alpha\Delta\epsilon$.
    $k \leftarrow k + 1$.
  until stopping criterion satisfied
end BARSUB,

where $0 < \lambda < 1$ is a scalar to ensure that $\epsilon^{k+1} > 0$.

FIG. 1. *An algorithm for the solution of* (23).

The first inequality is obvious, and the second follows from the fact that $\kappa\epsilon - \mu\ln(\epsilon)$ has minimum for $\epsilon = \frac{\mu}{\kappa}$ for positive μ. The set of all solutions $w(\mu)$ with $\mu > 0$ is called the central path. In LP, the central path has been studied intensively [22]. Rewriting (27), we get the following relation between μ and a point on the central path:

$$\mu = (\epsilon^*(\mu))^2 \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i^*(\mu)\|_2^2 + (\epsilon^*(\mu))^2\right)^{\frac{1}{2}}}. \tag{45}$$

Note that $(\boldsymbol{x}^*(\mu), \boldsymbol{r}^*(\mu)) \in X^p$ and $\boldsymbol{y}^*(\mu) \in Y^d$ are optimal solutions to (P_ϵ), respectively, (D_ϵ) for $\epsilon = \epsilon^*(\mu)$. It follows from (25) and (45) that

$$\begin{aligned}\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i^*(\mu)\|_2 + \boldsymbol{b}^T\boldsymbol{y}^*(\mu) &< \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i^*(\mu)\|_2^2 + (\epsilon^*(\mu))^2\right)^{\frac{1}{2}} + \boldsymbol{b}^T\boldsymbol{y}^*(\mu) \\ &= \epsilon^*(\mu)\sum_{i=1}^{\kappa}(1 - \|\boldsymbol{y}_i^*(\mu)\|_2^2)^{\frac{1}{2}} = \mu.\end{aligned} \tag{46}$$

Hence μ is a bound on the duality gap for (P) and (D). It follows that as $\mu \to 0$ we have $\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i^*(\mu)\|_2 + \boldsymbol{b}^T\boldsymbol{y}^*(\mu) \to 0$. In view of this property, an efficient algorithm should follow the central path approximately. This is the idea in the so-called interior path-following algorithm that was first proposed in [28] for LP. Let μ^k be the value of μ at the kth iteration. If we assume that a given value of $(\boldsymbol{x}^k, \boldsymbol{r}^k, \epsilon^k)$ is optimal for μ^k, i.e., $(\boldsymbol{x}^k, \boldsymbol{r}^k)$ is on the central path, then the relation (45) suggests the following

definition of μ^{k+1}:

$$\mu^{k+1} = \min\left(\beta\epsilon^2 \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}}, \mu^k\right), \tag{47}$$

where $0 < \beta < 1$ is a scalar. The minimum is taken as a precaution: if μ is not reduced, we iterate with the same value. If μ is reduced in the kth iteration of BARSUB then

$$g_\epsilon = \epsilon_k \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i^k\|_2^2 + \epsilon_k^2\right)^{\frac{1}{2}}} - \mu^k \epsilon_k{}^{-1} = \epsilon_k(1-\beta) \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i^k\|_2^2 + \epsilon_k^2\right)^{\frac{1}{2}}} > 0. \tag{48}$$

It remains to choose β. We know that for a constant value of μ, Newton's method will eventually reach a domain in which the unit step along the direction will reduce the barrier function [4, pp. 309–312]. The iterates can thereafter be expected to converge quadratically to the optimal value. We can then take the steplength α as a "measure" of how close the point is to the central path and set

$$\beta = (1 - \min(\alpha, \gamma)), \qquad 0 < \gamma < 1, \tag{49}$$

where α is the steplength and $1 - \gamma$ defines a lower bound for β. Note that β gets smaller as α approaches one. For a constant μ convergence follows from the properties of Newton's method. In (47) μ is either reduced or is constant. If μ is reduced the method converges because μ is a bound on the duality gap. If we keep μ constant, the method will converge to an optimal point for this value of μ meaning that (45) will be satisfied in a finite number of iterations, and since $0 < \beta < 1$ it follows that after some iterations μ will be reduced.

In practice, however, we have observed that with μ defined by

$$\mu^{k+1} = \beta\epsilon^2 \sum_{i=1}^{\kappa} \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} \tag{50}$$

in every iteration, the number of iterations is reduced. This is used in our implementation. In order to justify this compromise between theoretical convergence and practical efficiency we note that with μ^{k+1} defined as above, it follows from (48) and (50) that $g_\epsilon > 0$, and since the search direction is a descent direction $\Delta\epsilon < 0$. If ϵ remains fixed through some iterations Newton's method gives local quadratic convergence.

4. Implementation of the Newton barrier method. The method was implemented in ANSI-C, and has been ported to several different computers. An MPS-like format was used as input format.

4.1. The line search method. The steplength α in BARSUB must be calculated such that

$$F_\mu(r^k + \alpha\Delta\boldsymbol{r}, \epsilon^k + \alpha\Delta\epsilon) \approx \min_{0<\theta\leq\alpha_{\max}} F_\mu(r^k + \theta\Delta\boldsymbol{r}, \epsilon + \theta\Delta\epsilon)$$

and $F_\mu(r^k + \alpha\Delta\boldsymbol{r}, \epsilon^k + \alpha\Delta\epsilon) < F_\mu(r^k, \epsilon^k)$. In [26, pp. 42–46] a line search procedure is described which uses the special structure of the nonsmooth function F. Using the same notation as in [26], we define

$$\phi_i(\alpha) = \left(\left\|\boldsymbol{r}_i^k + \alpha\Delta\boldsymbol{r}_i\right\|_2^2 + \left(\epsilon^k + \alpha\Delta\epsilon\right)^2\right)^{\frac{1}{2}}.$$

By differentiation we get

$$\phi_i'(\alpha) = \frac{1}{\phi_i(\alpha)} \left(\left(\boldsymbol{r}_i^k\right)^T \Delta \boldsymbol{r}_i + \alpha \left\| \Delta \boldsymbol{r}_i \right\|_2^2 + \epsilon^k \Delta\epsilon + \alpha(\Delta\epsilon)^2 \right)$$

and

$$\phi_i''(\alpha) = \frac{1}{\phi_i(\alpha)} (\|\Delta \boldsymbol{r}_i\|_2^2 + (\Delta\epsilon)^2 - (\phi'(\alpha))^2).$$

Since $\alpha \leq \alpha_{\max}$ ϕ_i' and ϕ_i'' are always defined. However, as $\epsilon \to 0$ they can grow very large. Now let

$$f(\alpha) = F_\mu(\boldsymbol{r}^k + \alpha\Delta\boldsymbol{r}, \epsilon^k + \alpha\Delta\epsilon) = \sum_{i=1}^{\kappa} \phi_i(\alpha) - \mu \ln(\epsilon^k + \alpha\Delta\epsilon).$$

The gradient of $f(\alpha)$ is (see [26, p. 43])

$$f'(\alpha) = g(\alpha) = \sum_{i=1}^{\kappa} \phi_i'(\alpha) - \mu\Delta\epsilon(\epsilon^k + \alpha\Delta\epsilon)^{-1}$$

and the Hessian is

$$f''(\alpha) = h(\alpha) = \sum_{i=1}^{\kappa} \phi_i''(\alpha) + \mu(\Delta\epsilon)^2 \left(\epsilon^k + \alpha\Delta\epsilon\right)^{-2}.$$

With g and h defined as above, we use the same line search as in [26] except that in the unsafeguarded step (Step 1, [26, p. 44]) we always estimate the minimum of $f(\alpha)$ with the standard quadratic (Newton) approximation $\Delta\alpha = h^{-1}(\alpha^0)g(\alpha^0)$ as long as ϵ is larger than its lower bound ϵ_l, because we then expect f and its derivatives to be well conditioned. From the convexity of $f(\alpha)$ it is known that $g(0) < 0$ if the current point is not optimal, with $g(0) = 0$ otherwise. In practice, however, it may happen that $g(0) > 0$ in a neighbourhood of the optimal solution because of rounding errors. In this case the line search procedure puts $\alpha = 1$ (Newton steplength) and checks if this reduces the objective function. This was found to work well. Our experience from numerical experiments is that the number of iterations used in the method is closely related to the efficiency of the line search procedure. A more intensive study of this part of the algorithm could lead to an improvement of efficiency. It was observed for the test problems that the line search was cheap in comparison with the rest of the method, and that the line search normally required less than 20 steps (about 10 on average) to satisfy the following stopping criterion:

$$|g(\alpha)| \leq -\zeta g(0), \tag{51}$$

$$f(0) - f(\alpha) > -10^{-4}\alpha g(0), \tag{52}$$

which ensure a "sufficient decrease" in f [26, p. 45]. The parameter ζ controls how accurately the minimum of f is estimated in the given direction. In the numerical tests presented in this paper we set $\zeta = 0.01$, because we have observed it to be important that an accurate line search is used.

4.2. The initial point. The initial point x^0 is set to the solution of the least squares system:

$$\min_{x^0} \left\| A^T x^0 - b \right\|_2^2 \Leftrightarrow x^0 = (AA^T)^{-1}Ab. \tag{53}$$

Note that if $\|A^T x^0 - b\|_2 = 0$ then x^0 is also an optimal solution to (P). The initial value of ϵ^0 is set to

$$\epsilon^0 = \frac{\sum_{i=1}^{\kappa} \|r_i^0\|_2}{\kappa}, \qquad \text{where } r^0 = A^T x^0 - b. \tag{54}$$

Let $\eta = \max\{\|r_i^0\|_2 \mid i \in \mathcal{K}\}$. We have

$$\frac{\eta}{\kappa} \leq \epsilon^0 \leq \left(\|r_i^0\|_2^2 + (\epsilon^0)^2 \right)^{\frac{1}{2}} \leq 2\eta.$$

In this way it is ensured that the initial Hessian is well conditioned.

4.3. The stopping criterion. An estimate for the dual vector $y = g_r + H_r \Delta r + H_{r\epsilon} \Delta\epsilon$ is computed after each iteration. It follows from (34) that $Ay = 0$. A measure of the dual infeasibility ($y \in Y^d$) is given by

$$D_{\text{fea}} = \frac{\left(\sum_{i \in D} (\|y_i\|_2 - 1)^2 \right)^{\frac{1}{2}}}{\sqrt{\kappa}} < \epsilon_{\text{fea}}, \tag{55}$$

where $D = \{j \in \mathcal{K} \mid \|y_j\|_2 > 1\}$ and ϵ_{fea} is the feasibility tolerance. The algorithm is terminated if $D_{\text{fea}} < \epsilon_{\text{fea}}$ and the relative duality gap satisfies

$$\frac{\left| b^T y + \sum_{i=1}^{\kappa} \left(\|r_i^k\|_2^2 + (\epsilon^k)^2 \right)^{1/2} \right|}{\max(-b^T y, 1)} < \epsilon_{\text{opt}}. \tag{56}$$

Both ϵ_{fea} and ϵ_{opt} are set to 10^{-8}, which is the standard accuracy for testing LP codes.

If ϵ is close to the machine precision of the computer, the problem can be unacceptably ill conditioned. The parameter ϵ is therefore fixed equal to ϵ_l in the remaining iterations if it drops below this limit. It follows that in BARSUB $\Delta\epsilon = 0$ and $g_\epsilon = 0$ in these iterations. We know from Theorem 2.4 and the approximate bound given in §2 that the lower bound ϵ_l for ϵ must decrease with κ to ensure a given optimality. We therefore set $\epsilon_l = 10^{-12} * 1.5^{-(\kappa/10000)}$, which is found to ensure that the algorithm can find optimal solutions for $\epsilon_{\text{opt}} = 10^{-8}$.

4.4. Solving for the direction. In every iteration we need to solve the system

$$AHA^T \Delta x = g,$$

where $M = AHA^T$ is a positive definite matrix (see §3). It was observed that for large sparse problems, over 90% of the total time is used in this part of the algorithm. We find a Cholesky factorization of M:

$$M = LL^T,$$

where $\boldsymbol{L}$ is a lower triangular matrix. As noted in §3 $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\kappa]$, where $\boldsymbol{A}_i$ is $n \times \nu$, and $\boldsymbol{H}$ is a block diagonal matrix where every block is of size $\nu \times \nu$ (typically $\nu = 2$ or 3 is much smaller than m). We can find $\boldsymbol{M}$'s nonzero structure once and for all as follows.

Suppose A has the sparsity structure shown below and let

$$\boldsymbol{Q} = \boldsymbol{AH} = [\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_\kappa].$$

We have Fig. 2.

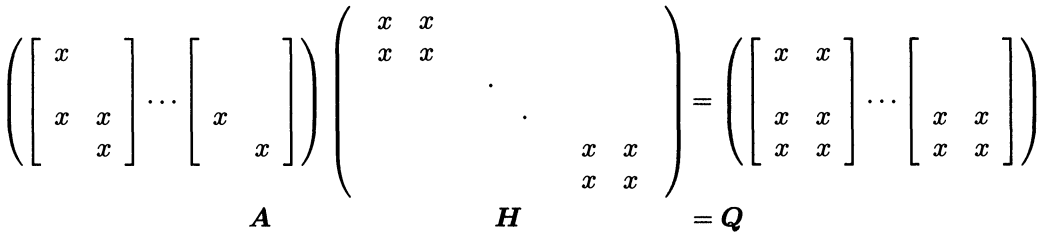


FIG. 2. *An example with $\nu = 2$ displaying the structure of $AH = Q$.*

Here $\boldsymbol{Q}_i$ is an $n \times \nu$ matrix, where every column has the same nonzero structure as illustrated in Fig. 2. Since the structure is given by the first column of $\boldsymbol{Q}_i$ we define

$$[\widehat{\boldsymbol{Q}}]_{ji} = [\boldsymbol{Q}_i]_{j1} \qquad \text{for } i = 1, \ldots, \kappa, \quad j = 1, \ldots, n$$

and let $\widehat{\boldsymbol{q}}_p$ be the pth row of $\widehat{\boldsymbol{Q}}$. It follows that

$$\operatorname{sym}(\boldsymbol{M}_{ij}) = \operatorname{sym}(\widehat{\boldsymbol{q}}_i \widehat{\boldsymbol{q}}_j^T) = 0 \qquad \text{always if } \operatorname{sym}((\widehat{\boldsymbol{q}}_i)_p (\widehat{\boldsymbol{q}}_j)_p) = 0, \qquad p = 1, \ldots, \kappa,$$

where sym() means that the element is symbolic (independent of numerical values). This structure of $\boldsymbol{M}$ is the most pessimistic but $\boldsymbol{M}$ is sparse if $\nu \ll m$ and $\boldsymbol{A}$ is sparse. We note that, except for finding the symbolic structure of $\boldsymbol{M}$, we can use exactly the same technique for solving (4.4) as in interior point methods for LP. A large scale implementation of an interior point method for LP is described in [1], from which the data structures and the fast Cholesky factorization are reused. Assuming that $\boldsymbol{M}$ is sparse we apply a minimum degree algorithm once to $\operatorname{sym}(\boldsymbol{M})$ in order to reduce the fill-in (number of nonzeros in $\boldsymbol{L}$). The code includes two different minimum degree algorithms, one programmed by E.D. Andersen using Multiple Elimination (see [20]), and one which is a slightly modified version of the algorithm described in [15]. The factorization used is a supernodal inner product Cholesky factorization as described in [15, pp. 149–156] and [3]. A supernode is a set of columns in $\boldsymbol{L}$ with the same nonzero structure under the diagonal, and they can therefore be handled as a dense submatrix [3, pp. 17–19]. The lower corner of $\boldsymbol{L}$ will normally be dense or nearly dense. Hence a dense Cholesky factorization is used for the remaining part of $\boldsymbol{L}$ if the density of a column is larger than 0.8, totally avoiding indirect addressing and therefore speeding up the algorithm significantly at the expense of a little extra storage [12]. The use of supernodes and dense matrix techniques makes it possible to unroll loops [3].

5. Computational results. All tests presented in this section were made on a CONVEX C3240 vector computer at Odense University. The code is compiled with

the CONVEX-C compiler using option "-O2" (full vectorization). In all numerical computations 64-bit floating-point arithmetic (implying between 15 and 16 digits accuracy) is used. Times are in CPU seconds and are measured with the ANSI-C procedure "clock." Computing time includes all processing except time for reading the data. Some of the test problems include one linear constraint. See Appendix A for handling linear constraints.

The test problems are finite element discretizations of problems from collapse load analysis. In our view these are good test problems because they are sparse, and they are not randomly generated (i.e., there is some model behind the problem formulation). Another nice feature of these problems is that they can easily be increased in size. Therefore it is possible to test the proposed method on very large problems.

TABLE 1
Results for sspN *with* $\nu = 3$.

NAME	κ	n	$\lvert A\rvert$	$\lvert AHA^T\rvert/2$	$\lvert L\rvert$
ssp4	25	15	224	97	101
ssp10	121	99	1760	1010	1807
ssp50	2601	2499	48800	31010	167591
ssp100	10201	9999	197600	127010	953095
ssp200	40401	39999	795200	514010	5038483
ssp300	90601	89999	1792800	1161010	12461749
ssp400	160801	159999	3190400	2068010	24937971

NAME	ITE	CPU	$\lVert r_i\rVert_2 \le \epsilon_l$	Dgap	D_{fea}	$\lVert Ay\rVert_2$
ssp4	18	0	0	4.2E-11	3.3E-09	2.5E-15
ssp10	19	2	0	2.8E-11	8.9E-09	7.0E-15
ssp50	18	103	0	2.7E-13	2.2E-09	4.2E-13
ssp100	21	649	0	1.4E-13	4.0E-09	9.0E-13
ssp200	25	4488	0	8.3E-15	5.8E-10	3.2E-12
ssp300	15	8344	0	1.8E-15	8.6E-11	2.9E-12
ssp400	15	20387	0	2.7E-16	1.3E-10	8.3E-12

In Tables 1–3 the numerical results are presented: $|AHA^T|/2$ is the number of nonzeros in AHA^T under and in the diagonal, $|L|$ is the number of nonzeros in the Cholesky factor, ITE is the number of iterations, CPU is cpu time used and $\|\boldsymbol{r}_i\|_2 \le \epsilon_l$ is the number of norms which is smaller than ϵ_l in the final solution. We have tested three different problems, all originating from collapse load analysis in the theory of rigid plastic materials. They are all sparse but differ in the number of zero norm terms at the minimum. The precise discretization of these problems as minimum sum of norms problems will be published separately. For each problem it is easy to vary the size, preserving the structure, by varying the mesh size of the grid. The first test problem, sspN (simply supported square plate with point load solved on an $N \times N$ grid), is described in [8]. In the optimal solution for these problems the active set (the set of terms with $\|\boldsymbol{r}_i\|_2 < \epsilon_l$) is empty. The results are seen in Table 1. The second test problem, lNa13 (Table 2), is a new discretization of the plane strain problem described in [7, 2] using divergence-free finite elements for the plastic flow. The size and accuracy of the discrete problem on an $N \times N$ grid is the same as on a $2N \times 2N$ grid in [2]. In the optimal solution up to more than 80% of the terms are active.

The third test problem, lNa20 (Table 3), is again a discretization of a model problem in plane strain, the so-called punch problem, where a hard die is punched into a half space of softer plastic material. In the optimal solution for this problem

TABLE 2
Results for lNa13 *with* $\nu = 2$.

NAME	κ	n	$\|A\|$	$\|AHA^T\|/2$	$\|L\|$
l3a13	49	52	1390	1142	1207
l12a13	625	640	21331	26406	58221
l21a13	1849	1876	64762	84384	271655
l30a13	3721	3760	131683	175086	736325
l39a13	6241	6292	222094	298512	1423399
l48a13	9409	9472	335995	454662	2561165
l57a13	13225	13300	473386	643536	3804247
l60a13	14641	14720	524403	713766	4342713
l78a13	24649	24752	885315	1211502	8062337
l84a13	28561	28672	1026499	1406502	10096365
l102a13	42025	42160	1512691	2078766	16633429
l120a13	58081	58240	2092843	2881926	24202413

NAME	ITE	CPU	$\|r_i\|_2 \leq \epsilon_l$	Dgap	D_{fea}	$\|Ay\|_2$
l3a13	26	2	0	8.4E-10	5.4E-10	9.3E-15
l12a13	33	68	179	1.6E-11	4.0E-10	1.0E-12
l21a13	45	391	553	1.1E-10	1.1E-14	1.7E-12
l30a13	52	1313	1756	2.7E-10	1.0E-12	3.1E-12
l39a13	58	3006	3416	5.2E-10	4.5E-14	1.5E-09
l48a13	91	9784	5718	8.0E-10	1.0E-17	8.3E-10
l57a13	97	15653	8727	7.7E-10	9.1E-14	1.3E-09
l60a13	80	15246	9843	8.6E-10	4.2E-15	7.4E-09
l78a13	134	52833	18061	1.0E-09	6.4E-09	3.4E-10
l84a13	131	69435	21211	1.2E-09	5.4E-11	7.2E-09
l102a13	188	189822	33529	8.0E-10	3.4E-14	3.3E-09
l120a13	176	284378	47602	1.0E-09	3.0E-09	2.1E-08

TABLE 3
Results for lNa20 *with* $\nu = 2$.

NAME	κ	n	$\|A\|$	$\|AHA^T\|/2$	$\|L\|$
l20a20	1681	1718	59054	76945	246012
l40a20	6561	6636	234145	315504	1511239
l60a20	14641	14754	525236	715653	4433304
l80a20	25921	26072	932327	1277402	8645485

NAME	ITE	CPU	$\|r_i\|_2 \leq \epsilon_l$	Dgap	D_{fea}	$\|Ay\|_2$
l20a20	59	438	1041	1.6E-10	1.6E-13	4.1E-13
l40a20	105	5704	5012	6.5E-10	2.7E-18	2.0E-11
l60a20	133	26847	13416	1.2E-09	4.5E-18	4.3E-12
l80a20	164	69460	24811	1.3E-09	2.0E-12	2.6E-11

up to more than 95% of the norms in the sum are active.

The three problems are different with respect to the number of active norms (number of $\|\boldsymbol{r}_i\|_2 \leq \epsilon_l$) which range from 0 in sspN to over 95% in l80a20. With these three types of test problems, the method is tried at problems which range from total smoothness to nearly total nonsmoothness in the optimal solution. We see that the duality gap, dual feasibility, and $\|\boldsymbol{Ay}\|_2$ for all problems are satisfactory, even for problems with a large number of active norms, where ill-conditioning can be expected.

We conclude that for this set of test problems our method is numerically stable.

The sspN problems do not have any active norms in the optimal solution. From our approximate bound in §2, the method should be expected to find primal and dual optimal solutions of very high accuracy measured in the duality gap. This is in fact the case, and the duality gap is closed to machine precision for the larger sspN problems. Furthermore, the number of iterations is nearly constant and even decreases for the larger problems. This could indicate that the method works as an unconstrained Newton method with quadratic convergence for this type of smooth problem.

Comparing Table 1 with Tables 2 and 3, it is seen that the number of iterations increases with the size of the problem and the number of active norms. This is what should be expected. However, the number of iterations is small. Also note that the $\boldsymbol{L}$ factor for all problems is sparse (for l120a13 the density of $\boldsymbol{L}$ is 1.4%).

The algorithm of [26], which is intended and implemented for smaller and dense problems, has also been applied to small test problems in plasticity [27, 14]. To our knowledge comparable numerical results, with our problem size and numbers of iterations, have not been published for any other method, except for the case where the convex problem is linearized, and an LP solver is used (see [7, 2]). The new method is much more efficient in terms of CPU time and memory than the LP method, as will be described in a forthcoming paper.

6. Conclusion. In this paper a complete duality theory has been established for the nonsmooth problem (P) and for its smooth approximation (P_ϵ) using a Lagrangian dual. This duality theory was used to establish a connection between the smooth problem (P_ϵ) and optimality for the original problem. A Newton barrier method was developed for the solution of (P_ϵ) which, for a strictly decreasing sequence of the barrier parameter μ, converges to a μ-optimal solution for the original problem (P). The duality theory provides, as in LP, an excellent stopping criterion for the new algorithm, which is one reason for the success of our method. Another reason is that we use second order estimates of the dual variables, which is observed to converge much faster than the normal first order estimates.

A large scale implementation for sparse problems of the method was described. With this implementation we obtain very promising results for large problems arising in plasticity which have not been achieved with any other algorithm.

Future work will include a study of the behavior of the algorithm on other types of problems, for example, the multifacility location problem [5]. It can also be noted that if $l = 1$ then (P) is a linear L_1 problem. For the L_1 problem $\boldsymbol{H}$ will actually be a diagonal matrix as in interior point methods for LP. Therefore we plan to compare our method with other methods for this type of problem; see [19].

Appendix A. Linear constraints. In this appendix we describe how linear equality constraints are handled. The problem is as follows:

$$(P_{\text{eq}}) \qquad \begin{aligned} &\min \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 \\ &\text{s.t.} \\ &\qquad (\boldsymbol{x}, \boldsymbol{r}) \in X^p_{\text{eq}}, \end{aligned}$$

where

$$X^p_{\text{eq}} = \left\{ (\boldsymbol{x}, \boldsymbol{r}) \in R^n \times R^{\kappa\nu} \mid \begin{bmatrix} \boldsymbol{A}^T \\ \boldsymbol{E}^T \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{b}_e \end{bmatrix} \right\} \tag{A.57}$$

and $\boldsymbol{E} \in R^{n\times l}$. In analogy with Lemma 2.1 this problem can be shown to have the following dual problem:

$$(D_{\text{eq}}) \qquad \begin{aligned} &\max \left(-\boldsymbol{b}^T\boldsymbol{y} - \boldsymbol{b}_e^T\boldsymbol{y}_e\right) \\ &\text{s.t.} \\ &\qquad (\boldsymbol{y}, \boldsymbol{y}_e) \in Y_{\text{eq}}^d, \end{aligned}$$

where

$$\text{(A.58)} \quad Y_{\text{eq}}^d = \left\{(\boldsymbol{y}, \boldsymbol{y}_e) \in R^{\kappa\nu} \times R^l \mid \boldsymbol{A}\boldsymbol{y} + \boldsymbol{E}\boldsymbol{y}_e = \boldsymbol{0}, \quad \|\boldsymbol{y}_i\|_2 \le 1, \quad i = 1, \ldots, \kappa\right\}.$$

As in §2, the smooth approximation to (P_{eq}) is given by

$$(P_{\text{eq}\epsilon}) \qquad \begin{aligned} &\min \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{1/2} \\ &\text{s.t.} \\ &\qquad (\boldsymbol{x}, \boldsymbol{r}) \in X_{\text{eq}}^p \end{aligned}$$

with dual problem

$$(D_{\text{eq}\epsilon}) \qquad \begin{aligned} &\max \left(-\boldsymbol{b}^T\boldsymbol{y} - \boldsymbol{b}_e^T\boldsymbol{y}_e + \epsilon \sum_{i=1}^{\kappa} (1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\right) \\ &\text{s.t.} \\ &\qquad (\boldsymbol{y}, \boldsymbol{y}_e) \in Y_{\text{eq}}^d. \end{aligned}$$

It is easily seen that Theorem 2.4 still holds. The only modification is that the algorithm is developed for the following barrier subproblem:

$$\text{(A.59)} \qquad \begin{aligned} &\min F_\mu(\boldsymbol{r}, \epsilon) = \sum_{i=1}^{\kappa} (\|\boldsymbol{r}_i\|_2^2 + \epsilon^2)^{\frac{1}{2}} - \mu \ln \epsilon \\ &\text{s.t.} \\ &\qquad (\boldsymbol{x}, \boldsymbol{r}) \in X_{\text{eq}}^p. \end{aligned}$$

The Newton search direction is determined by

$$\text{(A.60)} \qquad \frac{\partial L_Q}{\partial \Delta \boldsymbol{r}} = \boldsymbol{g}_r + \boldsymbol{H}_r \Delta \boldsymbol{r} + \boldsymbol{H}_{r\epsilon} \Delta\epsilon - \boldsymbol{y} = \boldsymbol{0},$$

$$\text{(A.61)} \qquad \frac{\partial L_Q}{\partial \Delta \boldsymbol{x}} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{E}\boldsymbol{y}_e = \boldsymbol{0},$$

$$\text{(A.62)} \qquad \frac{\partial L_Q}{\partial \Delta\epsilon} = g_\epsilon + \boldsymbol{H}_\epsilon \Delta\epsilon + \boldsymbol{H}_{r\epsilon}^T \Delta \boldsymbol{r} = \boldsymbol{0},$$

$$\text{(A.63)} \qquad \frac{\partial L_Q}{\partial \boldsymbol{y}} = \boldsymbol{A}^T \Delta \boldsymbol{x} - \Delta \boldsymbol{r} = \boldsymbol{0},$$

$$\text{(A.64)} \qquad \frac{\partial L_Q}{\partial \boldsymbol{y}_e} = \boldsymbol{E}^T \Delta \boldsymbol{x} = \boldsymbol{0}.$$

This system of equations can be rewritten into

$$\text{(A.65)} \qquad \begin{bmatrix} \boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T & \boldsymbol{E} \\ \boldsymbol{E}^T & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \boldsymbol{y}_e \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}(-\boldsymbol{g}_r + \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1} g_\epsilon) \\ \boldsymbol{0} \end{bmatrix}.$$

We assume that $\boldsymbol{E}$ is of full column rank and that the Cholesky factor $\boldsymbol{L}$ with $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T$ is known. Set $\boldsymbol{L}\boldsymbol{Q} = \boldsymbol{E}$ and $\boldsymbol{L}\boldsymbol{p} = \boldsymbol{A}(-\boldsymbol{g}_r + \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1} g_\epsilon)$. Then a solution to

(A.65) is given by

$$\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{y}_e = \boldsymbol{Q}^T\boldsymbol{p}, \tag{A.66}$$

$$\boldsymbol{L}^T\Delta\boldsymbol{x} = \boldsymbol{p} - \boldsymbol{Q}\boldsymbol{y}_e. \tag{A.67}$$

The full column rank of $\boldsymbol{E}$ ensures that $\boldsymbol{Q}^T\boldsymbol{Q}$ is nonsingular. Normally $\boldsymbol{Q}$ will be completely dense and it follows that this approach is expensive in computing time if l is large. This method is used in our implementation because the test problems include only one linear constraint. Another method would be to transform the problem as follows: Let $\boldsymbol{E}_i^T$ be row i in $\boldsymbol{E}^T$ and assume that $[\boldsymbol{E}^T]_{ij} \neq 0$. Then we can transform the problem by replacing x_j with

$$x_j = \frac{[\boldsymbol{b}_e]_i - \sum_{p \neq j} [\boldsymbol{E}^T]_{ip} x_p}{[\boldsymbol{E}^T]_{ij}}.$$

Appendix B. A generalization of Theorem 2.2. In this appendix we show how Theorem 2.2 can be generalized to the case where $\widehat{\boldsymbol{A}}$ is not of full column rank. If $\widehat{\boldsymbol{A}}$ is not of full column rank, then $\boldsymbol{y}^x$ in Theorem 2.2 is not uniquely defined by the necessary KKT conditions for the problem (11). In this case we replace (11) by

$$\begin{aligned} \min \quad & \sum_{i \in (\mathcal{K} \setminus J)} \|\boldsymbol{r}_i\|_2 \\ \text{s.t.} \quad & \\ & (\boldsymbol{x}, \boldsymbol{r}) \in X^p, \\ & \boldsymbol{r}_i = \widehat{\boldsymbol{A}}^T\boldsymbol{x} + \boldsymbol{D}\boldsymbol{x}_{\text{ar}} - \widehat{\boldsymbol{b}} = \boldsymbol{0} \quad \forall i \in J, \\ & \boldsymbol{0} \leq \boldsymbol{x}_{\text{ar}} \leq \boldsymbol{0}, \end{aligned} \tag{B.68}$$

where $\boldsymbol{x}_{\text{ar}}$ is a set of artificial variables and $\boldsymbol{D}$ is a diagonal matrix which is chosen such that $[\ \widehat{\boldsymbol{A}}^T \quad \boldsymbol{D}\ ]$ is of full row rank and such that

$$\widehat{\boldsymbol{A}}^T\boldsymbol{x} + \boldsymbol{D}\boldsymbol{x}_{\text{ar}} = \widehat{\boldsymbol{b}} \quad \Rightarrow \quad \boldsymbol{x}_{\text{ar}} = \boldsymbol{0}.$$

This corresponds to setting a Lagrange multiplier for a rank redundant constraint equal to 0. Then in Theorem 2.2 we can replace the problem (11) with (B.68) and $\widehat{\boldsymbol{A}}^T$ with the matrix $[\ \widehat{\boldsymbol{A}}^T \quad \boldsymbol{D}\ ]$ which is of full row rank and the theorem then follows. One way of obtaining $\boldsymbol{D}$ is to use a method similar to the method in [25].

Acknowledgments. I wish to thank E. Christiansen and E.D. Andersen for helpful comments and discussions. I also wish to thank M.L. Overton for suggesting the dual problem to (P).

REFERENCES

[1] K. Andersen, *An infeasible dual affine scaling method for linear programming*, COAL Bulletin, 22 (1993), pp. 19–27.

[2] K. Andersen and E. Christiansen, *Limit Analysis with the Dual Affine Scaling Algorithm*, Preprint 26, Dept. Of Math. and Computer Sci., Odense University, 1993, J. Comput. Applied Math., to appear.

[3] C. Ashcraft, R. Grimes, J. Lewis, B. Peyton, and H. Simon, *Progress in sparse matrix methods for large scale linear systems on vector supercomputers*, Internat. J. Supercomputer Appl., 1 (1987), pp. 10–30.

[4] M. BAZARAA, H. SHERALI, AND C. SHETTY, *Nonlinear Programming: Theory and Algorithms*, 2nd ed., John Wiley and Sons, New York, 1993.

[5] P. CALAMAI AND A. CONN, *A second-order method for solving the continuous multifacility location problem*, in Numerical Analysis: Proceedings of the Ninth Biennial Conference, Dundee, Scotland, Lecture Notes in Mathematics 912, G. Watson, ed., Berlin, 1982, Springer-Verlag, pp. 1–5.

[6] ———, *A projected Newton method for l_p norm location problems*, Math. Programming, 38 (1987), pp. 75–109.

[7] E. CHRISTIANSEN AND K. KORTANEK, *Computation of the collapse state in limit analysis using the lp primal affine scaling algorithm*, J. Comput. Appl. Math., 34 (1991), pp. 47–63.

[8] E. CHRISTIANSEN AND S. LARSEN, *Computations in limit analysis for plastic plates*, Internat. J. Numer. Methods Engrg., 19 (1983), pp. 169–184.

[9] U. ECKHARDT, *On an Optimization Problem Related to Minimal Surfaces with Obstacles*, in Optimization and Optimal Control, Lecture Notes in Mathematics 477, A. Dold and B. Eckmann, eds., Springer-Verlag, 1975, pp. 95–101.

[10] J. EYSTER, J. WHITE, AND W. WIERWILLE, *On solving multifacility location problems using a hyperboloid approximation procedure*, AIIE Trans., 5 (1973), pp. 1–6.

[11] A. FIACCO AND G. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968.

[12] J. FORREST AND J. TOMLIN, *Vector processing in simplex and interior methods for linear programming*, Ann. Oper. Res., 22 (1990), pp. 71–100.

[13] K. FRISCH, *The logarithmic potential method of convex programming*, Tech. report, University Institute of Economics, Oslo, Norway, 1955.

[14] V. GAUDRAT, *A Newton type algorithm for plastic limit analysis*, Comput. Methods Appl. Mech. Engrg., 88 (1991), pp. 207–224.

[15] A. GEORGE AND J.W.-H.LIU, *Computing Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[16] P. GILL, W. MURRAY, M. SAUNDERS, J. TOMLIN, AND M. WRIGHT, *On the projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method*, Math. Programming, 36 (1986), pp. 183–209.

[17] H. KUHN, *On a pair of dual nonlinear programs*, in Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1967, pp. 38–54.

[18] H. W. KUHN, *A note on Fermat's problem*, Math. Programming, 4 (1973), pp. 98–107.

[19] Y. LI, *A globally convergent method for l_p problems*, SIAM J. Optim., 3 (1993), pp. 609–629.

[20] J. LIU, *Modification of the minimum-degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.

[21] I. LUSTIG, R. MARSTEN, AND D. SHANNO, *Interior point methods for linear programming: Computational state of the art*, Technical report SOR 92-17, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1992.

[22] N. MEGIDDO, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods, N. Megiddo, ed., Springer-Verlag, 1989, pp. 131–158.

[23] S. MEHROTRA, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[24] M. MINOUX, *Mathematical Programming: Theory and Algorithms*, John Wiley and Sons, New York, 1986.

[25] B. MURTAGH AND M. SAUNDERS, *Large-scale linearly constrained optimization*, Math. Programming, 14 (1978), pp. 41–72.

[26] M. OVERTON, *A quadratically convergent method for minimizing a sum of Euclidian norms*, Math. Programming, 27 (1983), pp. 34–63.

[27] ———, *Numerical solution of a model problem from collapse load analysis*, in INRIA Conf. Comput. Methods Engrg. Appl. Sci., R. Glowinski and J. Lions, eds., North-Holland, Amsterdam, 1984, pp. 421–437.

[28] J. RENEGAR, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Programming, 40 (1988), pp. 59–73.

[29] J. ROSEN AND G. XUE, *On the convergence of Miehle's algorithm for the Euclidean multifacility location problem*, Oper. Res., 40 (1992), pp. 188–191.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 96–125, February 1996
© 1996 Society for Industrial and Applied Mathematics
007

AUGMENTED LAGRANGIAN-SQP-METHODS IN HILBERT SPACES AND APPLICATION TO CONTROL IN THE COEFFICIENTS PROBLEMS*

KAZUFUMI ITO† AND KARL KUNISCH‡

Abstract. Augmented Lagrangian methods for optimization problems in Hilbert spaces with second order updates of the Lagrangian are analyzed. The relationship and the differences to SQP-methods are pointed out. The convergence results depend on the second order sufficient optimality condition. Applications to parameter estimation problems are given and illustrated by numerical results. These illustrate the global behavior of the method.

Key words. augmented Lagrangians, SQP-methods, parameter estimation problems

AMS subject classifications. 65K10, 49D, 35R30

1. Introduction.

In this research we continue our efforts on developing stable and efficient techniques to solve parameter estimation problems formulated as nonlinear optimization problems. Such problems are ill posed in the sense of possible lack of continuous dependence of the minimizers with respect to perturbations in the problem data. Numerically this leads to instabilities of ad hoc algorithms. In [IK, IKK] we developed a method combining the regularized least squares and the output error formulation which allowed a stable estimation of the unknown coefficient, also in the presence of perturbed data, for fine discretization of the infinite dimensional problem and for a wide range of initial conditions. The mathematical framework was given by the augmented Lagrangian formulation with first order update of the Lagrange multiplier. The resulting algorithm has a linear rate of convergence.

In this paper we concentrate on second order methods. Section 2 is devoted to analyzing the augmented Lagrangian technique with second order update of the Lagrange multiplier in a general Hilbert space setting. Subsequently this method is compared to the augmented Lagrangian-SQP-technique. These two methods differ with respect to the solution of an auxiliary problem at each step of the iteration. Both methods are proved to be quadratically convergent for smooth optimization problems satisfying the second order sufficient optimality condition.

In §3 we concentrate on proving the second order sufficient optimality condition for a general class of parameter estimation problems with linear dynamics. These problems can be considered as control in the coefficient problems with a bilinear Lagrange functional. The lack of coercivity with respect to the unknown parameter is compensated by a regularization term. The second order sufficient optimality condition is shown to hold for observation norms that correspond to the norm of the state variable in the energy norm as well as with respect to weaker (L^2)-norms. The later may be advantageous in the presence of high noise levels.

Section 4 is devoted to calculating the operator representation of the algorithms described in §2 to parameter estimation problems associated with elliptic boundary

* Received by the editors April 19, 1993; accepted for publication (in revised form) May 16, 1994.
† Center for Research in Scientific Computing, North Carolina State University, Raleigh, NC 27695. This author's research was supported in part by National Science Foundation grant UINT-8521208.
‡ Institut für Mathematik, Technische Universität Berlin, D-10623, Berlin, Germany (kunisch@math.tu-berlin.de). This author's research was supported in part by Fonds zur Förderung der Wissenschaftlichen Forschung P-8146-PHY and the Bundesministerium für Wissenschaft und Forschung.

value problems. Once this is achieved the necessary discretization leading to implementable schemes is simple. The operator representation reveals a surprising simplicity of the proposed class of algorithms. Treating the state equation as a constraint retains the bilinear structure of the problem which is lost if the state variable is considered as a function of the unknown coefficients (controls). We shall also see that a modified form of an equation error step is contained in the algorithms that we propose. We remark that the potentially cumbersome numerical realization of the fourth order penalty term can be circumvented by an appropriate intermediate multiplier update. Section 5 is devoted to numerical case studies.

Let us put the present work into perspective with related research efforts. Section 2 on second order methods greatly benefits from the research by Bertsekas on finite dimensional optimization problems [B]. Concerning augmented Lagrangian-SQP-methods, again for finite dimensional problems, with appropriate updates of the Hessian, we mention the work by Byrd, Tapia, and Zhang [BTZ]. In [KK] a comparison of the numerical efficiency of the SQP-method and the augmented Lagrangian method with second order update is carried out. The SQP-method was applied to a one-dimensional parameter estimation problem in [KW]. There the resulting operator equation was reduced to an integral equation for the unknown parameter. In [KS] reduced SQP-methods were developed for parameter estimation problems. The shortcoming of SQP-methods for parameter estimation problems without augmentation is given by the fact that a globalization is required to produce a stable algorithm with a reasonably large radius of convergence.

2. Mixed augmented Lagrangian-SQP-methods. We consider the constrained minimization problem

$$\text{(P)} \qquad \min f(x) \text{ subject to } e(x) = 0,$$

where $f : X \to \mathbf{R}, e : X \to Y$, with X and Y real Hilbert spaces. Let x^* be a local solution of (P) and assume that f and e are twice continuously Fréchet differentiable in a convex neighborhood $V(x^*)$ of x^*. Derivatives as well as partial derivatives with respect to the variable x will be denoted by primes. It is assumed that f'' and e'' are Lipschitz continuous on $V(x^*)$. We do not distinguish by notation between the functional f' in the dual X^* of X and its Riesz representation in X.

The Lagrangian functional associated with (P) is denoted by $\mathcal{L} : X \times Y \to \mathbf{R}$ and is given by

$$\mathcal{L}(x, \lambda) = f(x) + \langle \lambda, e(x) \rangle_Y.$$

The space $X \times Y$ is endowed with the Hilbert space product norm. We shall require the regularity assumption

$$\text{(H1)} \qquad e'(x^*) \text{ is surjective.}$$

With (H1) holding there exists a Lagrange multiplier $\lambda^* \in Y$ such that the following first order necessary optimality condition is satisfied:

$$\text{(2.1)} \qquad \mathcal{L}'(x^*, \lambda^*) = 0, \quad e(x^*) = 0.$$

We also make use of the following second order optimality condition:

$$\text{(H2)} \qquad \begin{aligned} &\text{there exists } \kappa > 0 \text{ such that} \\ &\mathcal{L}''(x^*, \lambda^*)(h, h) \geq \kappa \, |h|_X^2 \quad \text{for all} \quad h \in \ker e'(x^*). \end{aligned}$$

Here $\mathcal{L}''(x^*, \lambda^*)$ denotes the bilinear form characterizing the second Fréchet derivative of $\mathcal{L}$ with respect to x at (x^*, λ^*). For any $c > 0$ the augmented Lagrangian functional $\mathcal{L}_c : X \times Y \to \mathbf{R}$ is defined by

$$\mathcal{L}_c(x, \lambda) = f(x) + \langle \lambda, e(x) \rangle_Y + \frac{c}{2} |e(x)|_Y^2 .$$

We note that the necessary optimality condition implies

$$\mathcal{L}_c'(x^*, \lambda^*) = 0, \quad e(x^*) = 0 \ \text{ for all } \ c \geq 0. \tag{2.2}$$

LEMMA 2.1. *Let* (H1) *and* (H2) *hold. Then there exists a neighborhood* $V(x^*, \lambda^*)$ *of* $(x^*, \lambda^*), \bar{c} > 0$ *and* $\bar{\sigma} > 0$ *such that*

$$\mathcal{L}_c''(x, \lambda)(h, h) \geq \bar{\sigma} |h|_X^2 \ \text{ for all } h \in X, \ (x, \lambda) \in V(x^*, \lambda^*) \ \text{ and } \ c \geq \bar{c}.$$

Proof. Due to (H1) and (H2) there exist $\bar{\sigma} > 0$ and $\bar{c} > 0$ such that

$$\mathcal{L}_c''(x^*, \lambda^*)(h, h) \geq 2\bar{\sigma} |h|_X^2 \ \text{ for all } \ h \in X \ \text{ and } \ c \geq \bar{c}$$

[PT]. Due to continuity of $(x, \lambda) \to \mathcal{L}_c''(x, \lambda)$ the conclusion of the lemma follows.

Lemma 2.1 implies in particular that $x \to \mathcal{L}_c(x, \lambda^*)$ can be bounded from below by a quadratic function. This fact is referred to as augmentability of $\mathcal{L}_c$.

LEMMA 2.2. *Let* (H1) *and* (H2) *hold. Then there exist* $\bar{\sigma} > 0, \bar{c} > 0$, *and a neighborhood* $\tilde{V}(x^*)$ *of* x^* *such that*

$$\mathcal{L}_c(x, \lambda^*) \geq \mathcal{L}_c(x^*, \lambda^*) + \bar{\sigma} |x - x^*|_X^2 \ \text{ for all } x \in \tilde{V}(x^*) \ \text{ and } c \geq \bar{c}. \tag{2.3}$$

Proof. Due to Taylor's theorem, Lemma 2.1, and (2.2), we find for $x \in V(x^*)$

$$\begin{aligned} \mathcal{L}_c(x, \lambda^*) &= \mathcal{L}_c(x^*, \lambda^*) + \frac{1}{2}\mathcal{L}_c''(x^*, \lambda^*)(x - x^*, x - x^*) + o(|x - x^*|_X^2) \\ &\geq \mathcal{L}_c(x^*, \lambda^*) + \frac{\bar{\sigma}}{2} |x - x^*|_X^2 + o(|x - x^*|_X^2). \end{aligned}$$

The claim follows from this estimate.

Without loss of generality we further assume that neighborhoods $V(x^*)$ and $V(\lambda^*)$ are chosen such that the regularity properties of f and e specified at the beginning of this section and in Lemma 2.2 hold for all $x \in V(x^*)$ and the conclusion of Lemma 2.1 holds for all $(x, \lambda) \in V(x^*) \times V(\lambda^*) = V(x^*, \lambda^*)$. Due to (H1) we can further assume that $e'(x)$ is surjective for all $x \in V(x^*)$.

To find x^* numerically we observe that (x^*, λ^*) is a solution to

$$\begin{aligned} &\mathcal{L}_c'(x, \lambda) = f'(x) + e'(x)^*(\lambda + ce(x)) = 0, \\ &e(x) = 0, \end{aligned} \tag{2.4}$$

where $e'(x)^*$ denotes the adjoint of $e'(x)$. To solve (2.4) for (x^*, λ^*) one can use Newton's method.

Given a current iterate (x, λ) the next iterate $(\hat{x}, \hat{\lambda})$ is the solution to

$$\begin{pmatrix} \mathcal{L}_c''(x, \lambda) & e'(x)^* \\ e'(x) & 0 \end{pmatrix} \begin{pmatrix} \hat{x} - x \\ \hat{\lambda} - \lambda \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_c'(x, \lambda) \\ e(x) \end{pmatrix}. \tag{2.5}$$

Alternatively, (2.4) can be used to define the λ-update only, whereas the x-update is calculated by a different technique. We shall see next that (2.4) can be solved for $\hat{\lambda}$ without recourse to $\hat{x}$.

For $(x, \lambda) \in V(x^*, \lambda^*)$ and $c \geq \bar{c}$ we define $B(x, \lambda) \in \mathcal{L}(Y)$ by

$$B(x, \lambda) = e'(x)\mathcal{L}_c''(x, \lambda)^{-1} e'(x)^*. \tag{2.6}$$

Here $\mathcal{L}(Y)$ denotes the space of bounded linear operators from Y into itself. Note that $B(x, \lambda)$ is invertible. In fact, there exists a constant $k > 0$ such that

$$\begin{aligned} \langle B(x, y)y, y\rangle_Y &= \langle \mathcal{L}_c''(x, \lambda)^{-1} e'(x)^* y, e'(x)^* y\rangle_X \\ &\geq k\, |e'(x)^* y|_X^2 \end{aligned}$$

for all $y \in Y$. Since $e'(x)^*$ is injective and has closed range there exists $\hat{k}$ such that

$$|e'(x)^* y|_X^2 \geq \hat{k}\, |y|_Y^2 \quad \text{for all } y \in Y,$$

and by the Lax–Milgram theorem continuous invertibility of $B(x, \lambda)$ follows, provided that $(x, \lambda) \in V(x^*, \lambda^*)$ and $c \geq \bar{c}$. Premultiplying the first equation in (2.5) by $e'(x)\mathcal{L}_c''(x, \lambda)^{-1}$ we obtain

$$\begin{aligned} \hat{\lambda} &= \lambda + B(x, \lambda)^{-1} \left[e(x) - e'(x)\mathcal{L}_c''(x, \lambda)^{-1} \mathcal{L}_c'(x, \lambda)\right], \\ \hat{x} &= x - \mathcal{L}_c''(x, \lambda)^{-1} \mathcal{L}_c'(x, \hat{\lambda}); \end{aligned} \tag{2.7}$$

compare [B, p. 134]. Whenever the dependence of $(\hat{x}, \hat{\lambda})$ on (x, λ, c) is important we shall write $(\hat{x}(x, \lambda, c), \hat{\lambda}(x, \lambda, c))$ in place of $(\hat{x}, \hat{\lambda})$. If, for fixed λ, $x = x(\lambda)$ is chosen as a local solution to

$$\min \mathcal{L}_c(x, \lambda) \text{ subject to } x \in X, \tag{2.8}$$

then $\mathcal{L}_c'(x(\lambda), \lambda) = 0$ and

$$\hat{\lambda}(x(\lambda), \lambda, c) = \lambda + B(x(\lambda), \lambda)^{-1} e(x(\lambda)). \tag{2.9}$$

We point out that (2.9) can be interpreted as a second order update to the Lagrange variable. To acknowledge this, let d_c denote the dual functional associated with $\mathcal{L}_c$, i.e.,

$$d_c(\lambda) = \min \mathcal{L}_c(x, \lambda) \text{ subject to } \{x : |x - x^*| \leq \epsilon\}$$

for some $\epsilon > 0$. Then the first and second derivatives of d_c with respect to λ satisfy

$$\nabla_\lambda d_c(\lambda) = e(x(\lambda))$$

and

$$\nabla_\lambda^2 d_c(\lambda) = -B(x(\lambda), \lambda)$$

and (2.9) presents a Newton step for maximizing d_c.

Returning to (2.5) we note that its "blockwise" solution given in (2.7) requires setting up and inverting $\mathcal{L}_c''(x, \lambda)$. Following an argument due to Bertsekas we next

argue that $\mathcal{L}_c''(x,\lambda)$ can be avoided during the iteration. One only requires $\mathcal{L}_0''(x,\lambda) = \mathcal{L}''(x,\lambda)$. In fact, we find

$$\mathcal{L}_c'(x,\lambda) = \mathcal{L}_0'(x,\lambda + ce(x))$$

and

$$\mathcal{L}_c''(x,\lambda) = \mathcal{L}_0''(x,\lambda + ce(x)) + c\,\langle e'(x)(\cdot), e'(x)(\cdot)\rangle\,.$$

Consequently, (2.5) can be expressed as

$$\begin{aligned} &\begin{pmatrix} \mathcal{L}_0''(x,\lambda+ce(x)) + ce'(x)^*e'(x) & e'(x)^* \\ e'(x) & 0 \end{pmatrix} \begin{pmatrix} \hat{x}-x \\ \hat{\lambda}-\lambda \end{pmatrix} \\ &= -\begin{pmatrix} \mathcal{L}_0'(x,\lambda+ce(x)) \\ e(x) \end{pmatrix}. \end{aligned} \tag{2.10}$$

Using the second equation $e'(x)(\hat{x}-x) = -e(x)$ in the first of (2.10) we arrive at

$$\mathcal{L}_0''(x,\lambda+ce(x))(\hat{x}-x) - ce'(x)^*e(x) + e'(x)^*(\hat{\lambda}-\lambda) = -\mathcal{L}_0'(x,\lambda+ce(x)),$$

and hence (2.5) is equivalent to

$$\begin{aligned} &\begin{pmatrix} \mathcal{L}_0''(x,\lambda+ce(x)) & e'(x)^* \\ e'(x) & 0 \end{pmatrix} \begin{pmatrix} \hat{x}-x \\ \hat{\lambda}-(\lambda+ce(x)) \end{pmatrix} \\ &= -\begin{pmatrix} \mathcal{L}_0'(x,\lambda+ce(x)) \\ e(x) \end{pmatrix}. \end{aligned} \tag{2.11}$$

Solving (2.5) is thus equivalent to (i) carrying out the first order multiplier iteration

$$\tilde{\lambda} = \lambda + ce(x)$$

(ii) and solving

$$\begin{pmatrix} \mathcal{L}_0''(x,\tilde{\lambda}) & e'(x)^* \\ e'(x) & 0 \end{pmatrix} \begin{pmatrix} \hat{x}-x \\ \hat{\lambda}-\tilde{\lambda} \end{pmatrix} = -\begin{pmatrix} \mathcal{L}_0'(x,\tilde{\lambda}) \\ e(x) \end{pmatrix}$$

for $(\hat{x},\hat{\lambda})$.

It will be convenient to introduce the matrix of operators

$$M(x,\lambda) = \begin{pmatrix} \mathcal{L}_0''(x,\lambda) & e'(x)^* \\ e'(x) & 0 \end{pmatrix}.$$

With (H1) and (H2) holding there exist a constant $\kappa > 0$ and a neighborhood $U(x^*,\lambda^*) \subset V(x^*,\lambda^*)$ of (x^*,λ^*) such that

$$\| M^{-1}(x,\lambda) \|_{\mathcal{L}(\mathcal{X}\times\mathcal{Y})} \leq \kappa \quad \text{for all } (x,\lambda) \in U(x^*,\lambda^*). \tag{2.12}$$

LEMMA 2.3. *Assume that* (H1) *and* (H2) *hold. Then there exists a constant* $K > 0$ *such that for any* $(x,\lambda) \in U(x^*,\lambda^*)$ *the solution* $(\hat{x},\hat{\lambda})$ *of*

$$M(x,\lambda)\begin{pmatrix} \hat{x}-x \\ \hat{\lambda}-\lambda \end{pmatrix} = -\begin{pmatrix} \mathcal{L}'(x,\lambda) \\ e(x) \end{pmatrix}$$

satisfies

$$\left|(\hat{x},\hat{\lambda})-(x^*,\lambda^*)\right|_{X\times Y}\le K\,|(x,\lambda)-(x^*,\lambda^*)|^2_{X\times Y}\,. \tag{2.13}$$

Proof. Note that

$$M(x,\lambda)\begin{pmatrix}\hat{x}-x\\ \hat{\lambda}-\lambda\end{pmatrix}=-\begin{pmatrix}\mathcal{L}'(x,\lambda)-\mathcal{L}'(x^*,\lambda^*)\\ e(x)-e(x^*)\end{pmatrix}$$

and consequently

$$M(x,\lambda)\begin{pmatrix}\hat{x}-x^*\\ \hat{\lambda}-\lambda^*\end{pmatrix}=-\begin{pmatrix}\mathcal{L}'(x,\lambda)-\mathcal{L}'(x^*,\lambda^*)\\ e(x)-e(x^*)\end{pmatrix}+M(x,\lambda)\begin{pmatrix}x-x^*\\ \lambda-\lambda^*\end{pmatrix}.$$

This equality further implies

$$M(x,\lambda)\begin{pmatrix}\hat{x}-x^*\\ \hat{\lambda}-\lambda^*\end{pmatrix}=\int_0^1\left[M(x,\lambda)-M(tx+(1-t)x^*,t\lambda+(1-t)\lambda^*)\right]\begin{pmatrix}x-x^*\\ \lambda-\lambda^*\end{pmatrix}dt.$$

The regularity properties of f and e imply that $(x,\lambda)\to M(x,\lambda)$ is Lipschitz continuous on $U(x^*,\lambda^*)$ for Lipschitz constant $\gamma>0$. Thus we obtain

$$\left|M(x,\lambda)\begin{pmatrix}\hat{x}-x^*\\ \hat{\lambda}-\lambda^*\end{pmatrix}\right|_{X\times Y}\le\frac{\gamma}{2}\,|(x,\lambda)-(x^*,\lambda^*)|^2_{X\times Y}\,,$$

and by (2.12)

$$\left|(\hat{x},\hat{\lambda})-(x^*,\lambda^*)\right|_{X\times Y}\le\frac{\gamma\kappa}{2}\,|(x,\lambda)-(x^*,\lambda^*)|^2_{X\times Y}\,,$$

which implies the claim.

We now describe three algorithms and analyze their convergence. All are based on iterations of (2.11). In the first one x in (2.11) is determined by solving

$$\min\mathcal{L}_c(x,\lambda_n)\text{ subject to }x\in V(x^*)$$

approximately. In the second one x is chosen as the solution of the auxiliary problem

$$\min\mathcal{L}_c(x,\lambda_n)\text{ subject to }x\in\overline{V(x^*)},$$

and in the third algorithm, x is taken as the solution $\hat{x}$ of the previous iteration of (2.11).

ALGORITHM 1.

(i) Choose $\lambda_0\in Y, c\in(\bar{c},\infty)$ and $\sigma\in(0,c-\bar{c}], n=0$.

(ii) Determine $\tilde{x}\in V(x^*)$ such that

$$\mathcal{L}_c(\tilde{x},\lambda_n)\le\mathcal{L}_c(x^*,\lambda_n)=f(x^*).$$

(iii) Set $\tilde{\lambda}=\lambda_n+\sigma e(\tilde{x})$.

(iv) Solve for $(\hat{x},\hat{\lambda})$:

$$M(\tilde{x},\tilde{\lambda})\begin{pmatrix}\hat{x}-\tilde{x}\\ \hat{\lambda}-\tilde{\lambda}\end{pmatrix}=-\begin{pmatrix}\mathcal{L}'_0(\tilde{x},\tilde{\lambda})\\ e(\tilde{x})\end{pmatrix}.$$

(v) Set $\lambda_{n+1} = \hat{\lambda}$, $n = n+1$ and goto (ii).

Remark 2.4. (i) From Theorem 2.5 below it will be seen that the choice for the initial guess λ_0 is essential for a good performance of Algorithm 1.

(ii) The proof of Theorem 2.5 reveals that at the nth iteration of Algorithm 1, the scalars c and σ can be replaced by the nth elements of sequences $\{c_n\}_{n=1}^{\infty}$ and $\{\sigma_n\}_{n=1}^{\infty}$, if only $\sigma_n \leq c_n - \bar{c}$ for all n and $\inf_n \sigma_n > 0$.

(iii) In the proof of Theorem 2.5 as well as in that of Theorem 2.6 we utilize (2.3) from Lemma 2.2. Conditions (H1) and (H2) are sufficient conditions for (2.3) to hold for $c \geq \bar{c} > 0$. If (2.3) can be shown to hold for all $c \geq 0$ then $\bar{c}$ can be chosen equal to 0 and $\sigma = c$ is admissible in (i) of Algorithm 1.

THEOREM 2.5. *Let* (H1) *and* (H2) *hold. If* $|\lambda_0 - \lambda^*|_Y$ *is sufficiently small, then Algorithm* 1 *is well defined and*

$$|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} \leq K\left(1 + \frac{1}{\sigma\bar{\sigma}}\right)|\lambda_n - \lambda^*|_Y^2 \tag{2.14}$$

for all $n = 0, 1, \ldots$. *Here* x_{n+1} *stands for* $\hat{x}$ *of step* (iv) *of Algorithm* 1 *and* K *is given by* (2.13).

Proof. Let $\hat{\eta}$ be the largest radius for a ball centered at (x^*, λ^*) and contained in $U(x^*, \lambda^*)$. We define

$$\eta = \min\left\{\frac{\hat{\eta}}{\sqrt{a}}, \frac{1}{Ka}\right\}, \tag{2.15}$$

where $a = 1 + \frac{1}{\sigma\bar{\sigma}}$. The proof is based on an induction argument. If

$$|\lambda_0 - \lambda^*| < \eta,$$

then the first iterate of Algorithm 1 is well defined, $|(x_1, \lambda_1) - (x^*, \lambda^*)|_{X\times Y} < \eta$, and (2.14) holds with $n = 0$. This will follow from the general arguments given below. Assuming that $|(x_n, \lambda_n) - (x^*, \lambda^*)|_{X\times Y} < \eta$, we show that Algorithm 1 is well defined for $n+1$, that $|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} < \eta$, and that (2.14) holds. Using (ii) of Algorithm 1 we have

$$\begin{aligned} f(x^*) \geq \mathcal{L}_c(\tilde{x}, \lambda_n) &= f(\tilde{x}) + \langle \lambda^*, e(\tilde{x})\rangle_Y + \langle \lambda_n - \lambda^*, e(\tilde{x})\rangle_Y \\ &\quad + \frac{1}{2}\bar{c}\,|e(\tilde{x})|_Y^2 + \frac{1}{2}(c - \bar{c})\,|e(\tilde{x})|_Y^2 \\ &\geq \mathcal{L}_{\bar{c}}(\tilde{x}, \lambda^*) + \langle \lambda_n - \lambda^*, e(\tilde{x})\rangle_Y + \frac{\sigma}{2}|e(\tilde{x})|_Y^2 \\ &= \mathcal{L}_{\bar{c}}(\tilde{x}, \lambda^*) + \frac{1}{2\sigma}\Big(|\lambda_n + \sigma e(\tilde{x}) - \lambda^*|_Y^2 - |\lambda_n - \lambda^*|_Y^2\Big). \end{aligned}$$

In view of Lemma 2.2 and the fact that $\tilde{x}$ is chosen in $V(x^*)(\subset \tilde{V}(x^*))$ we find

$$\bar{\sigma}\,|\tilde{x} - x^*|_X^2 + \frac{1}{2\sigma}\left|\tilde{\lambda} - \lambda^*\right|_Y^2 \leq \frac{1}{2\sigma}|\lambda_n - \lambda^*|_Y^2 \tag{2.16}$$

and consequently

$$\left|(\tilde{x}, \tilde{\lambda}) - (x^*, \lambda^*)\right|_{X\times Y}^2 \leq \left(1 + \frac{1}{2\sigma\bar{\sigma}}\right)|\lambda_n - \lambda^*|_Y^2. \tag{2.17}$$

This implies that $|(\tilde{x}, \tilde{\lambda}^*) - (x^*, \lambda^*)|_{X\times Y} < \hat{\eta}$ and hence Lemma 2.3 is applicable with $(x, \lambda) = (\tilde{x}, \tilde{\lambda})$ and (iv) of Algorithm 1 is well defined. Combining (2.13) with (2.17) we find

$$|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} \leq K\left(1 + \frac{1}{2\sigma\bar{\sigma}}\right)|\lambda_n - \lambda^*|_Y^2,$$

which is (2.14). By the definition of η we further obtain

$$|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} \leq |\lambda_n - \lambda^*|_Y < \eta.$$

This ends the proof.

The algorithm that we analyze next differs from Algorithm 1 in as far as the minimization of $\mathcal{L}(\cdot, \lambda_n)$ is carried out exactly.

ALGORITHM 2.

This coincides with Algorithm 1, except for (i) and (ii) which are replaced by:

(i') Choose $\lambda_0 \in Y, c \in (\bar{c}, \infty)$, and set $\sigma = c - \bar{c}$, $n = 0$.

(ii') Determine $\tilde{x}$ as a solution of

$$(\mathrm{P}_{\mathrm{aux}}) \qquad \min \mathcal{L}_c(x, \lambda_n) \text{ subject to } x \in \overline{V(x^*)}.$$

The existence of a solution to $(\mathrm{P}_{\mathrm{aux}})$ is guaranteed if, for example, the following conditions on f and e hold:

$$\text{(H3)} \qquad \begin{array}{ll} f : X \to \mathbf{R} & \text{is weakly lower semicontinuous,} \\ e : X \to Y & \text{maps weakly convergent sequences} \\ & \text{to weakly convergent sequences.} \end{array}$$

Under the conditions of Theorem 2.6 below it follows that the solutions $\tilde{x}$ of $(\mathrm{P}_{\mathrm{aux}})$ satisfy $\tilde{x} \in V(\tilde{x})$.

THEOREM 2.6. *Let* (H1)–(H3) *hold. If* $\frac{1}{c-\bar{c}}|\lambda_0 - \lambda^*|^2$ *is sufficiently small, then Algorithm* 2 *is well defined and*

$$(2.18) \qquad |(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} \leq \frac{\hat{K}}{c - \bar{c}}|\lambda_n - \lambda^*|_Y^2$$

for a constant $\hat{K}$ *independent of* c *and* $n = 0, 1, \ldots$.

Proof. Let $\hat{\eta}$ be defined as in the proof of Theorem 2.5 and let γ be a Lipschitz constant for f' and e' on $U(x^*, \lambda^*)$. Furthermore, let $E = e'(x^*)$ and note that $(EE^*)^{-1}E \in \mathcal{L}(X)$ as a consequence of (H1). We define

$$\bar{M} = 1 + 2(\bar{c}\gamma)^2 + 8\gamma^2(1 + \mu)^2 \parallel (EE^*)^{-1}E \parallel^2,$$

where $\mu = \left(1 + \frac{\bar{c}\gamma}{\sqrt{2\sigma\bar{\sigma}}}\right)|\lambda_0 - \lambda^*|_Y + |\lambda^*|_Y$, and we put

$$(2.19) \qquad \eta = \min\left(\hat{\eta}\sqrt{\frac{2\sigma\bar{\sigma}}{\bar{M}}}, \frac{2\sigma\bar{\sigma}}{K\bar{M}}\right),$$

where K is given in Lemma 2.3. Let us assume that

$$|\lambda_0 - \lambda^*|_Y < \eta.$$

The proof will be given by induction on n. The case $n = 0$ follows from the general arguments given below. For the induction step we assume that

$$|\lambda_i - \lambda^*|_Y \leq |\lambda_{i-1} - \lambda^*|_Y \quad \text{for } n = 1, \dots, n. \tag{2.20}$$

First note that (2.16) also holds for $(\tilde{x}, \tilde{\lambda})$ of Algorithm 2. This implies in particular that $|\tilde{x} - x^*| \leq \sqrt{\frac{\bar{M}}{2\sigma\bar{\sigma}}}\,|\lambda_0 - \lambda^*| < \hat{\eta}$ and hence $\tilde{x} \in V(x^*)$. The necessary optimality conditions for (P) and (P_{aux}) with $\tilde{x} \in V(x^*)$ are given by

$$f'(x^*) + E^*\lambda^* = 0$$

and

$$f'(\tilde{x}) + e'(\tilde{x})^*(\lambda_n + ce(\tilde{x})) = 0.$$

Subtracting these two equations and defining $\bar{\lambda} = \lambda_n + ce(\tilde{x})$ gives

$$E^*(\lambda^* - \bar{\lambda}) = f'(\tilde{x}) - f'(x^*) + (e'(\tilde{x})^* - e'(x^*)^*)(\bar{\lambda})$$

and consequently

$$\lambda^* - \bar{\lambda} = (EE^*)^{-1}E\left[f'(\tilde{x}) - f'(x^*) + (e'(\tilde{x})^* - e'(x^*)^*)(\bar{\lambda})\right].$$

We obtain

$$|\lambda^* - \bar{\lambda}|_Y \leq 2\gamma(1 + |\bar{\lambda}|_Y) \parallel (EE^*)^{-1}E \parallel |\tilde{x} - x^*|_X. \tag{2.21}$$

To estimate $\bar{\lambda}$ observe that due to (2.16) and (2.20)

$$\begin{aligned} |\bar{\lambda}|_Y &\leq \left|\tilde{\lambda} - \bar{\lambda}\right|_Y + \left|\tilde{\lambda}\right|_Y = \bar{c}\,|e(\tilde{x}) - e(x^*)|_Y + |\lambda^*|_Y + \left|\tilde{\lambda} - \lambda^*\right|_Y \\ &\leq \bar{c}\gamma\,|\tilde{x} - x^*|_Y + |\lambda^*|_Y + |\lambda_n - \lambda^*|_Y \leq \left(1 + \frac{\bar{c}\gamma}{\sqrt{2\sigma\bar{\sigma}}}\right)|\lambda_n - \lambda^*|_Y \\ &\quad + |\lambda^*|_Y \leq \mu. \end{aligned} \tag{2.22}$$

Using (2.20)–(2.22) and (2.16) we find

$$\begin{aligned} &|(\tilde{x}, \tilde{\lambda}) - (x^*, \lambda^*)|^2_{X\times Y} \leq |\tilde{x} - x^*|^2_X + 2|\tilde{\lambda} - \bar{\lambda}|^2_Y + 2|\bar{\lambda} - \lambda^*|^2_Y \\ &= |\tilde{x} - x^*|^2_X + 2(\bar{c}\gamma)^2|\tilde{x} - x^*|^2_X + 8\gamma^2(1+\mu)^2 \parallel (EE^*)^{-1}E \parallel^2 \ |\tilde{x} - x^*|^2_X \\ &= \frac{\bar{M}}{2\sigma\bar{\sigma}}\,|\lambda_n - \lambda^*|^2_X < \frac{\bar{M}}{2\sigma\bar{\sigma}}\eta^2 \leq \hat{\eta}^2. \end{aligned}$$

This implies that Lemma 2.3 is applicable with $(x, \lambda) = (\tilde{x}, \tilde{\lambda})$.

We find

$$|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} \leq \frac{K\bar{M}}{2\sigma\bar{\sigma}}\,|\lambda_n - \lambda^*|^2_X, \tag{2.23}$$

and (2.18) is proved with $\hat{K} = \frac{K\bar{M}}{2\bar{\sigma}}$.

From (2.20), (2.23), and the definition of η we also obtain

$$|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X\times Y} \leq |\lambda_n - \lambda^*|_Y < \eta.$$

This implies (2.20) for $i = n+1$ and the proof is finished.

Comparing Theorems 2.5 and 2.6 we observe that the region of convergence with respect to the initial guess λ_0 to λ^* can be increased by increasing c and by solving (P_{aux}) exactly.

In the third algorithm that we analyze here we consider the other extreme: We delete step 2 of Algorithms 1 and 2 and directly iterate (2.11).

ALGORITHM 3.

(i) Choose $(x_0, \lambda_0) \in X \times Y, c \geq 0$ and put $n = 0$.
(ii) Set $\tilde{\lambda} = \lambda_n + ce(x_n)$.
(iii) Solve for $(\hat{x}, \hat{\lambda})$:

$$M(x_n, \tilde{\lambda}) \begin{pmatrix} \hat{x} - x_n \\ \hat{\lambda} - \tilde{\lambda} \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_0'(x_n, \tilde{\lambda}) \\ e(x_n) \end{pmatrix}.$$

(iv) Set $(x_{n+1}, \lambda_{n+1}) = (\hat{x}, \hat{\lambda})$, $n = n+1$ and goto (ii).

THEOREM 2.7. *Let* (H1) *and* (H2) *hold. If* $c\,|(x_0, \lambda_0) - (x^*, \lambda^*)|_{X \times Y}$ *is sufficiently small, then Algorithm* 3 *is well defined and*

$$|(x_{n+1}, \lambda_{n+1}) - (x^*, \lambda^*)|_{X \times Y} \leq \tilde{K}\, |(x_n, \lambda_n) - (x^*, \lambda^*)|^2_{X \times Y}$$

for a constant $\tilde{K}$ *independent of* $n = 0, 1, 2, \ldots$, *but depending on* c.

Proof. Let $\hat{\eta}$ and γ be defined as in the proof of Theorem 2.6. We introduce

$$\eta = \min \left\{ \frac{\hat{\eta}}{\sqrt{a}}, \frac{1}{Ka} \right\}, \tag{2.24}$$

where $a = \max\left(2, 1 + 2c^2\gamma^2\right)$ and K is defined in Lemma 2.3.

Let us assume that

$$|(x_0, \lambda_0) - (x^*, \lambda^*)|_{X \times Y} < \eta.$$

Again we proceed by induction and the case $n = 0$ follows from the general arguments given below. Let us assume that $|(x_n, \lambda_n) - (x^*, \lambda^*)|_{X \times Y} < \eta$. Then

$$\begin{aligned} \left|(x_n, \tilde{\lambda}) - (x^*, \lambda^*)\right|^2_{X \times Y} &\leq |x_n - x^*|^2_X + 2c^2\, |e(x_n) - e(x^*)|^2_Y + 2\, |\lambda_n - \lambda^*|^2_Y \\ &\leq |x_n - x^*|^2_X + 2c^2\gamma^2\, |x_n - x^*|^2_X + 2\, |\lambda_n - \lambda^*|^2_Y \\ &\leq a\, |(x_n, \lambda_n) - (x^*, \lambda^*)|^2_{X \times Y} < \hat{\eta}^2, \end{aligned}$$

and thus Lemma 2.3 is applicable with $(x, \lambda) = (x_n, \tilde{\lambda})$.

3. Application to bilinear optimization problems arising in parameter identification. In this section we consider the regularized least squares problems

$$(P_\beta) \qquad \begin{aligned} &\min \tfrac{1}{2}|\mathcal{I}u - z|^2_{Z_2} + \tfrac{\beta}{2}|Nq|^2_{Z_1} \\ &e(q, u) = 0 \\ &q \in D, \end{aligned}$$

where

$(q,u) \in X_1 \times X_2 = X$,
$\beta \geq 0$ regularization parameter,
$z \in Z_2$ observation,
$\mathcal{I} \in \mathcal{L}(X_2, Z_2)$ observation operator,
$e : X \to Y$ equality constraint,
$N : X_1 \to Z_1$ bounded linear operator with closed range,
X_i, Y, Z_i Hilbert spaces,
$D \subset X_1$ closed convex set.

In the notation of §2 we have $x = (q,u)$ and

$$f(x) = |\mathcal{I}u - z|^2_{Z_2} + \frac{\beta}{2}|Nq|^2_{Z_1}.$$

Here N realizes regularization by a seminorm. For N = identity and $Z_1 = X_1$ we have regularization with a full norm. Finally e is assumed to be of the special form

$$e(q,u) = b(q,u) + l_1(q) + l_2(u) + f, \tag{C1}$$

where $f \in Y, l_1 \in \mathcal{L}(X_1, Y), l_2 \in \mathcal{L}(X_2, Y)$, and $b : X_1 \times X_2 \to Y$ is a bounded bilinear form, i.e., b is linear in each of its two variables separately and it satisfies

$$|b(q,u)|_Y \leq \| b \| \, |q|_{X_1} |u|_{X_2} \quad \text{for all} \quad (q,u) \in X,$$

where $\| b \| \in \mathbf{R}$.

To ascertain existence of a solution to (P_β) we make use of the fact that any element $q \in X_1$ can be uniquely decomposed as

$$q = q_1 + q_2 \in \ker N + (\ker N)^\perp$$

and that

$$|Nq|_{Z_1} \geq \kappa |q|_{X_1} \tag{3.1}$$

for some $\kappa > 0$ independent of $q \in (\ker N)^\perp$.

LEMMA 3.1. *Let us assume that*

$$\begin{aligned} &\mathcal{I} = \text{ identity and } X_2 = Z_2, \text{ or} \\ &\text{for any sequence } \{(q^n, u^n)\} \subset D \times X_2 \\ &\text{with } e(q^n, u^n) = 0 \text{ it follows that } \{u^n\} \text{ is bounded,} \end{aligned} \tag{3.2}$$

$$\begin{aligned} &\text{for any sequence } \{(q^n, u^n)\} \subset D \times X_2 \text{ with } e(q^n, u^n) = 0, u^n \rightharpoonup u, \\ &\text{and } \{(b(q_2^n, u^n) + l_2(q_2^n)\} \text{ bounded in } Y, \text{ it follows that} \\ &\{q_1^n\} \text{ is bounded in } X_1, \text{ and} \end{aligned} \tag{3.3}$$

$$\begin{aligned} &b \text{ maps weakly convergent sequences in } X \text{ into} \\ &\text{weakly convergent sequences in } Y. \end{aligned} \tag{3.4}$$

Then (P_β) *has a solution* $x^\beta = (q^\beta, u^\beta) \in X$ *for every* $\beta > 0$.

Proof. Let (q^n, u^n) be a minimizing sequence in X. By (3.2) the sequence $\{(q_2^n, u^n)\}_{n=1}^{\infty}$ is bounded. Here we put $q^n = q_1^n + q_2^n \in \ker N + (\ker N)^{\perp}$. Due to (3.3) and the special form of e it follows that $\{q_1^n\}_{n=1}^{\infty}$ is bounded as well and thus $\{(q^n, u^n)\}_{n=1}^{\infty}$ is a bounded sequence in X. Hence there exists a weakly convergent subsequence with weak limit (q^β, u^β). Due to (3.4) and weak lower semicontinuity of the norm, (q^β, u^β) is a solution of (P_β).

Remark 3.2. Condition (3.3) holds true, for example, if $b = 0$ and $l_2(q) \geq \kappa|q|$ for some $\kappa > 0$ independent of $q \in \ker N$ or if $l_2 = 0, u \neq 0$ and $b(q, v) \geq \kappa|q||v|$ for some $\kappa > 0$ independent of $(q, v) \in \ker N \times X_2$.

Henceforth it will be assumed that (P_β), $\beta > 0$, has a solution $x^\beta = (q^\beta, u^\beta)$. For $\beta \geq 0$ let $Q^\beta \times U^\beta = \{(q^\beta, u^\beta)$ is solution of $(\mathrm{P}_\beta)\}$.

LEMMA 3.3. *For $\beta > \beta_0 > 0$ we have*

(i) $\sup_{q^\beta \in Q^\beta} |N(q^\beta)| \leq \inf_{q^{\beta_0} \in Q^{\beta_0}} |N(q^{\beta_0})|$,

(ii) $\sup_{u^{\beta_0} \in U^{\beta_0}} |\mathcal{I}u^{\beta_0} - z| \leq \inf_{u^\beta \in U^\beta} |\mathcal{I}u^\beta - z|$.

If (P_0) *has a solution then* (i) *and* (ii) *hold for* $\beta > \beta_0 \geq 0$. *If, moreover,* (3.2)–(3.4) *hold then*

(iii) $\lim_{\beta \to 0^+} \inf_{q^\beta \in Q^\beta} |N(q^\beta)| = \lim_{\beta \to 0^+} \sup_{q^\beta \in Q^\beta} |N(q^\beta)| = \min_{q^0 \in Q^0} |N(q^0)|$,

(iv) $\sup_{u^\beta \in U^\beta} |\mathcal{I}u^\beta - z|^2 \leq |\mathcal{I}u^0 - z|^2 + \beta \left(\min_{q^0 \in Q^0} |Nq^0|^2 - \sup_{q^\beta \in Q^\beta} |Nq^\beta|^2 \right)$,

(v) $\lim_{\beta \to 0^+} \beta^{-1} \left(\sup_{u^\beta \in U^\beta} |\mathcal{I}u^\beta - z|^2 - |\mathcal{I}u^0 - z|^2 \right) = 0$.

The proof of this lemma follows with minor modifications from [CK].

The first and second Fréchet derivatives of e are given by

$$e_q(q,u)(h) = b(h,u) + l_1(h),\ e_u(q,u)(v) = b(q,v) + l_2(v),$$
$$e_{qu}(q,u)(h,v) = b(h,v),\ e_{qq}(q,u)(h,h) = e_{uu}(q,u)(v,v) = 0.$$

We require the assumption that for some $\beta^* \in (0, 1]$

(C2) $\qquad e_u(q^\beta, u^\beta) : X_2 \to Y$ is surjective for all $\beta \in [0, \beta^*]$.

We note that (C2) implies (H1) of §2.

LEMMA 3.4. *For $\beta \in (0, \beta^*)$ there exists a Lagrange multiplier $\lambda (= \lambda^\beta)$ of (P_β) and the necessary optimality conditions of first order are given by*

$$\langle \beta N^* N q^\beta + e_q^*(q^\beta, u^\beta)\lambda, q - q^\beta \rangle \geq 0 \text{ for all } q \in D, \tag{3.5}$$

and

$$\mathcal{I}^*(\mathcal{I}u^\beta - z) + e_u(q^\beta, u^\beta)^*\lambda = 0. \tag{3.6}$$

The proof of this lemma is quite simple and we directly turn to analyze (H2) for (P_β). We shall require the following condition.

(C3)

(i) The unregularized problem (P_0) has a solution. There exist constants K_1 and K_2 such that

(ii) $\| e_u^{-1}(q^\beta, u^\beta) \|_{\mathcal{L}(Y,X_2)} \le K_1$ for all $\beta \in [0, \beta^*]$, and
(iii) $\left| e_u^{-1}(q^\beta, u^\beta) e_q(q^\beta, u^\beta) h \right|_{X_2} \ge K_2 |h|_{X_1}$ for all $\beta \in [0, \beta^*]$ and $h \in \ker N$.
With (C3) (i), (ii) holding there exists a constant K_3 such that

$$\| e_u^{-1}(q^\beta, u^\beta) e_q(q^\beta, u^\beta) \|_{\mathcal{L}(X_1,X_2)} \le K_3 \text{ for all } \beta \in [0, \beta^*]. \tag{3.7}$$

We first consider the case where $\mathcal{I}$ is the identity operator.

PROPOSITION 3.5. *Assume that* (C1)–(C3) *hold with* $\mathcal{I}$ = *identity and* $X_2 = Z_2$, *and that*

$$|u^0 - z|_{X_2}^2 \le \frac{2\beta}{K_1^2 \| b \|} \left[\frac{\kappa^2 K_2^2}{2(K_2^2 + 2K_3)} - \frac{K_1^2}{2} \| b \| \left(\min_{q^0 \in Q^0} N(q^0) - \sup_{q^\beta \in Q^\beta} N(q^\beta) \right) \right]. \tag{3.8}$$

Then we have

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 &= |v|_{X_2}^2 + \beta |Nh|_{Z_1}^2 + \langle \lambda, b(h,v) \rangle_Y \\ &\ge \frac{1}{4} |v|_{X_2}^2 + \frac{\beta}{2} \frac{\kappa^2 K_2^2}{K_2^2 + 2K_3} |h|_{X_1}^2, \end{aligned}$$

for all $(h,v) \in \ker e'(q^\beta, u^\beta)$.

Proof. We recall that κ was defined in (3.1). Without loss of generality we may assume that $\frac{8\kappa^2}{K_2^2 + 2K_3} \le 1$. For $(h,v) \in \ker e'(q^\beta, u^\beta)$ we find due to (C2) that

$$v = e_u^{-1}(q^\beta, u^\beta) e_q(q^\beta, u^\beta) h. \tag{3.9}$$

We have the following estimate:

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 &= |v|_{Z_2}^2 + \beta |Nh|_{Z_1}^2 + \langle \lambda, b(h,v) \rangle_Y \\ &\ge |v|_{X_2}^2 + \beta |Nh|_{Z_1}^2 - |\lambda|_Y \| b \| |h|_{X_1} |v|_{X_2} \\ &\ge \frac{1}{2} |v|_{X_2}^2 + \beta |Nh|_{Z_1}^2 - \frac{1}{2} K_1^2 |u^\beta - z|_{X_2}^2 \| b \|^2 |h|_{X_1}^2 \\ &\ge \frac{1}{2} |v|_{X_2}^2 + \beta \kappa^2 |h_2|_{X_1}^2 - r(\beta) \left(|h_2|_{X_1}^2 + |h_2|_{X_1}^2 \right), \end{aligned} \tag{3.10}$$

where we used (C3)(ii), and put $h = h_1 + h_2 \in \ker N + (\ker N)^\perp$ and

$$r(\beta) = \frac{1}{2} K_1^2 |u^\beta - z|_Y^2 \| b \|^2.$$

From (3.10) we obtain

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 \ge \frac{1}{4} |v|_{X_2}^2 &+ \frac{\beta}{4} \frac{\mu}{1+\mu} \left| e_u^{-1} e_q h_1 \right|_{X_2}^2 \\ &- \frac{\beta}{4} \mu \left| e_u^{-1} e_q h_2 \right|_{X_2}^2 + \left(\beta \kappa^2 - r(\beta) \right) |h_2|_{X_1}^2 - r(\beta) |h_1|_{X_1}^2, \end{aligned}$$

where $\mu \in (0, \infty)$ is arbitrary and e_q, e_u are evaluated at (q^β, u^β). We further obtain

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 \le \frac{1}{4} |v|_{X_2}^2 &+ \left(\frac{\beta}{4} K_2^2 \frac{\mu}{1+\mu} - r(\beta) \right) |h_1|_{X_1}^2 \\ &+ \left[\beta \left(\kappa^2 - K_3 \frac{\mu}{4} \right) - r(\beta) \right] |h_2|_{X_2}^2, \end{aligned}$$

and for $\mu = \frac{8\kappa^2}{K_2^2+2K_2} \leq 1$,

$$\begin{aligned}
\mathcal{L}''(q^\beta, u^\beta)(h,v)^2 &\geq \frac{1}{4}|v|_{X_2}^2 + \left(\beta\frac{\kappa^2 K_2^2}{K_2^2+2K_3} - r(\beta)\right)|h|_{X_1}^2 \\
&= \frac{1}{4}|v|_{X_2}^2 + \beta\left[\frac{\kappa^2 K_2^2}{K_2^2+2K_3} - \frac{1}{2\beta}K_1^2 \parallel b \parallel^2 |u^\beta - z|^2\right]|h|_{X_1}^2 \\
&\geq \frac{1}{4}|v|_{X_2}^2 + \beta\Bigg[\frac{\kappa^2 K_2^2}{K_2^2+2K_3} - \frac{1}{2\beta}K_1^2 \parallel b \parallel^2 |u^0 - z|^2 \\
&\qquad + \frac{1}{2}K_1^2 \parallel b \parallel^2 \left(\min_{q^0\in Q^0} N(q^0) - \sup_{q^\beta \in Q^\beta} N(q^\beta)\right)\Bigg]|h|_{X_1}^2 \\
&\geq \frac{1}{4}|v|_{X_2}^2 + \frac{\beta}{2}\frac{\kappa^2 K_2^2}{K_2^2+2K_3}|h|_{X_1}^2,
\end{aligned}$$

if

$$|u^0 - z|_{X_2}^2 \leq \frac{2\beta}{K_1^2 \parallel b \parallel}\left[\frac{\kappa^2 K_2^2}{2(K_2^2+2K_3)} - \frac{1}{2}K_1^2 \parallel b \parallel \left(\min_{q^0\in Q^0} N(q^0) - \sup_{q^\beta\in Q^\beta} N(q^\beta)\right)\right].$$

This is the desired estimate.

As a consequence of the proof of Proposition 3.5 we find the following corollary.

COROLLARY 3.6. *Assume that* (C1)–(C3) *hold, that* $\mathcal{I} = N =$ *identity,* $Z_1 = X_1, Z_2 = X_2$, *and*

$$|u^0 - z|_{X_2}^2 \leq \frac{\beta}{K_1^2 \parallel b \parallel}\left[1 - K_1^2 \parallel b \parallel^2 \left(\min_{q^0\in Q^0} N(q^0) - \sup_{q^\beta\in Q^\beta} N(q^\beta)\right)\right].$$

Then

$$\mathcal{L}''(q^\beta, u^\beta)(h,v)^2 \geq \frac{\beta}{2}\left(|h|_{X_1}^2 + |v|_{X_2}^2\right)$$

for all $(h,v) \in X_1 \times X_2$.

Referring back to Remark 2.4 (iii) we observe that under the assumptions of Corollary 3.6, $\bar{c} = 0$ is admissible in the results of §2.

To treat more general output operators we require the following conditions:

(C4)

(i) range $\mathcal{I}$ is dense in Z_2,

(ii) there exists $|||b||| \in \mathbf{R}$ such that

$$\langle \lambda, b(h,v)\rangle_Y \leq |||b|||\ |h|_{X_1}|\mathcal{I}v|_{Z_2}|(\mathcal{I}^*)^{-1}\lambda|_{Z_2}$$

for all $\lambda \in$ range $\mathcal{I}^*$, $h \in X_1$ and $v \in X_2$,

(iii) $\lambda^\beta \in$ range $\mathcal{I}^*$ for all $\beta \in [0, \beta^*]$,

(iv) there exists $\tilde{K}_1$ such that

$$\parallel (\mathcal{I}^*)^{-1}\left(e_u^*(q^\beta, u^\beta)\right)^{-1}\mathcal{I}^* \parallel_{\mathcal{L}(Z_2)} \leq \tilde{K}_1 \quad \text{for all} \quad \beta \in [0, \beta^*].$$

(C5) There exists $\tilde{K}_2$ such that

$$|\mathcal{I}e_u^{-1}(q^\beta, u^\beta)e_q(q^\beta, u^\beta)h|_{Z_2} \geq \tilde{K}_2|h|_{X_1} \text{ for all } h \in \ker N \text{ and all } \beta \in [0, \beta^*].$$

We note that (C4) (i) implies that $\mathcal{I}^*$ is injective and that (C4)(iii) is implied by the assumption that

$$\text{range } \left(e_u^*(q^\beta, u^\beta)\right)^{-1} \mathcal{I}^* \subset \text{ range } \mathcal{I}^*.$$

In applications, (C4)(iii) is a regularity condition for the Lagrange multiplier; see Example 3.11.

PROPOSITION 3.7. *Assume that* (C1), (C2), (C3) (i), (ii), (C4), *and* (C5) *hold and that*

(3.11)
$$|\mathcal{I}u^0 - z|_{Z_2}^2 \leq \frac{2\beta}{\tilde{K}_1^2 |||b|||} \left[\frac{\kappa^2 \tilde{K}_2^2}{2(\tilde{K}_2^2 + 2\tilde{K}_3)} - \frac{\tilde{K}_1^2}{2} |||b||| \left(\min_{q^0 \in Q^0} N(q^0) - \sup_{q^\beta \in Q^\beta} N(q^\beta) \right) \right].$$

Then

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 &\geq |\mathcal{I}v|_{Z_2}^2 + \frac{\beta}{2} \frac{\kappa^2 \tilde{K}_2^2}{\tilde{K}_2^2 + 2\tilde{K}_3} |h|_{X_1}^2 \\ &\geq \frac{1}{4} |\mathcal{I}v|_{Z_2}^2 + \beta \min \left(\frac{\kappa^2 \tilde{K}_2^2}{\tilde{K}_2^2 + 2\tilde{K}_3}, \frac{1}{K_3} \right) \left(|h|_{X_1}^2 + |v|_{X_2}^2 \right) \end{aligned}$$

holds for all $(h,v) \in \ker e'(q^\beta, u^\beta)$, *where* $\tilde{K}_3 = K_3 \cdot \| \mathcal{I} \|_{\mathcal{L}(X_2, Z_2)}$.

Proof. Arguing as in (3.10), we find by (C4)

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 &= |\mathcal{I}v|_{Z_2}^2 + \beta |Nh|_{Z_1}^2 + \langle \lambda, b(h,v) \rangle_Y \\ &\geq |\mathcal{I}v|_{Z_2}^2 + \beta |Nh|_{Z_1}^2 - |||b||| \, |h|_{X_1} |\mathcal{I}v|_{Z_2} |(\mathcal{I}^*)^{-1} \lambda|_{Z_2} \\ &\geq \frac{1}{2} |\mathcal{I}v|_{Z_2}^2 + \beta \kappa^2 |h_2|_{X_1}^2 \\ &\quad - \frac{1}{2} |||b|||^2 \, |h|_{X_1}^2 |(\mathcal{I}^*)^{-1} e_u^*(q^\beta, u^\beta)^{-1} \mathcal{I}^* (\mathcal{I}u^\beta - z)|^2 \\ &\geq \frac{1}{2} |\mathcal{I}v|_{Z_2}^2 + \beta \kappa^2 |h_2|_{X_1}^2 - \frac{1}{2} |||b|||^2 \tilde{K}_1^2 |\mathcal{I}u^\beta - z|_{Z_2}^2 |h|_{X_1}^2 \\ &= \frac{1}{2} |\mathcal{I}v|_{Z_2}^2 + \beta \kappa^2 |h_2|_{X_1}^2 - \tilde{r}(\beta) |h|_{X_1}^2, \end{aligned} \tag{3.12}$$

where $\tilde{r}(\beta) = \frac{1}{2} \tilde{K}_1^2 |\mathcal{I}u^\beta - z|_{Z_2}^2 |||b|||^2$.

From (3.12) we have

$$\begin{aligned} \mathcal{L}''(q^\beta, u^\beta)(h,v)^2 &\geq \frac{1}{4} |\mathcal{I}v|_{Z_2}^2 + \frac{\beta}{4} \frac{\mu}{1+\mu} |\mathcal{I} e_u^{-1} e_q h_1|_{Z_2}^2 \\ &\quad - \frac{\beta}{4} \mu |\mathcal{I} e_u^{-1} e_q h_2|_{Z_2}^2 + \left(\beta \kappa^2 - \tilde{r}(\beta) \right) |h_2|_{X_1}^2 - \tilde{r}(\beta) |h_1|_{X_1}^2 \\ &\geq \frac{1}{4} |\mathcal{I}v|_{Z_2}^2 + \left(\frac{\beta}{4} \tilde{K}_2^2 \frac{\mu}{1+\mu} - \tilde{r}(\beta) \right) |h_1|_{X_1}^2 \\ &\quad + \left[\beta \left(\kappa^2 - K_3 \frac{\mu}{4} \| \mathcal{I} \|_{\mathcal{L}(X_2, Z_2)} \right) - \tilde{r}(\beta) \right] |h_2|_{X_1}^2. \end{aligned}$$

Putting $\mu = \frac{8\kappa^2}{\tilde{K}_2^2 - 2\tilde{K}_3}$ (and again assuming that $\mu \leq 1$) we find

$$\mathcal{L}''(q^\beta, u^\beta)(h,v)^2 \geq \frac{1}{4}|\mathcal{I}v|_{Z_2}^2 + \left(\beta \frac{\kappa^2 \tilde{K}_2^2}{\tilde{K}_2^2 + 2\tilde{K}_3} - \tilde{r}(\beta)\right)|h|_{X_1}^2.$$

As in the proof of Proposition 3.5 this estimate implies

$$\mathcal{L}''(q^\beta, u^\beta)(h,v)^2 \geq \frac{1}{4}|\mathcal{I}v|_{Z_2}^2 + \frac{\beta}{2}\frac{\kappa^2 \tilde{K}_2^2}{\tilde{K}_2^2 + 2\tilde{K}3}|h|_{X_1}^2$$

for all β satisfying (3.11).

By (3.9) and (3.7) (which is a consequence of (C3)(i)) we obtain

$$\mathcal{L}''(q^\beta, u^\beta)(h,v)^2 \geq \frac{1}{4}|\mathcal{I}v|_{Z_2}^2 + \beta \min\left(\frac{\kappa^2 \tilde{K}_2^2}{\tilde{K}_2 + 2\tilde{K}_3}, \frac{1}{K_3}\right)\left(|h|_{X_1}^2 + |v|_{X_2}^2\right)$$

as desired.

COROLLARY 3.8. (i) *Under the assumptions of Proposition* 3.5,

$$\mathcal{L}(q,u) - \mathcal{L}(q^\beta, u^\beta) \geq \frac{1}{4}|u - u^\beta|_{X_2} + \frac{\beta}{2}\frac{\kappa^2 K_2^2}{K_2^2 + 2K_3}|q - q^\beta|_{X_1}^2$$

for all $(q,u) \in X$.

(ii) *Under the assumptions of Corollary* 3.6,

$$\mathcal{L}(q,u) - \mathcal{L}(q^\beta, u^\beta) \geq \frac{1}{4}|\mathcal{I}(u - u^\beta)|_{Z_2}^2 + \beta \min\left(\frac{\kappa^2 \tilde{K}_2^2}{\tilde{K}_2^2 + 2\tilde{K}_3}, \frac{1}{K_3}\right) \cdot \left(|q - q^\beta|_{X_1}^2 + |u - u^\beta|_{X_2}^2\right)$$

for all $(q,u) \in X$.

Examples 3.9 and 3.11, which follow, illustrate the applicability of Lemma 3.1, Proposition 3.5, and Proposition 3.7. These examples concern themselves with the estimation of the diffusion coefficient in elliptic boundary value problems.

Example 3.9. (H^1−output). We consider

$$\begin{aligned} &\min \tfrac{1}{2}|u - z|_{H_0^1}^2 + \tfrac{\beta}{2}\left(|\nabla q|_{L^2}^2 + \textstyle\sum_{1 \leq i,j \leq 2} |D_{x_i} D_{x_j} q|_{L^2}^2\right), \\ &e(q,u) = (-\Delta)^{-1}(-\nabla \cdot (q \nabla u) + pu + f) = 0, \end{aligned}$$

where the function spaces are considered over a domain $\Omega \subset \mathbf{R}^2$ with smooth boundary or over a convex domain, and Δ denotes the Laplacian from H_0^1 to H^{-1}. Furthermore, $p \in L^\infty(\Omega)$ with $p(x) \geq 0$ a.e. is fixed.

We put

$$X_1 \times X_2 = H^2 \times H_0^1,$$

$$Z_2 = Y = H_0^1, \ \mathcal{I} = \text{identity}, \ z \in H_0^1,$$

$$Nq = (q_{x_1}, q_{x_2}, q_{x_1 x_1}, q_{x_1 x_2}, q_{x_2 x_2}),$$

$$Z_1 = \otimes_{i=1}^5 L^2,$$

$$D = \{q \in H^2 : q(x) \geq \alpha > 0\},$$

$$b(q,u) = (-\Delta)^{-1}(-\nabla \cdot (q\nabla u)),$$

$$l_1 = 0,\ l_2(u) = (-\Delta)^{-1}(pu), 0 \neq f \in H^{-1}.$$

Clearly, $l_2 \in \mathcal{L}(H_0^1)$ and

$$|b(q,u)|_{H_0^1} = |\nabla \Delta^{-1} \nabla \cdot (q\nabla u)|_{L^2} \leq |q\nabla u|_{L^2} \leq k_1 |q|_{H^2} |u|_{H_0^1},$$

where $k_1 = \| b \|$ is the embedding constant from H^2 into L^∞. Thus b is a bounded bilinear form. The kernel of N is the set of constant functions and (3.1) follows from well-known results on Sobolev spaces [M]. Turning to the conditions of Lemma 3.1, we note that (3.2) is clearly satisfied and, using Remark 3.2, we find that (3.3) holds if $u \neq 0$ (which is equivalent to $f \neq 0$). Finally (3.4) follows from the compact embedding of H^2 into L^∞. To verify (C2) and (C3) let us calculate

$$e_u(q,u)v = (-\Delta)^{-1}(-\nabla \cdot (q\nabla u) + pv),$$

and

$$e_q(q,u)h = \Delta^{-1}(\nabla \cdot (h\nabla u)).$$

For $q \in D$ it follows that $e_u(q,u)$ is surjective and thus (C2) is satisfied. Turning to (C3), the unregularized problem (P_0) has a solution, for example, if z is attainable, i.e., if there exists $(q,u) \in D \times X_2$, with $e(q,u) = 0$ and $u = z$, or if D is bounded. The Lax–Milgram lemma implies that $\| e_u^{-1}(q,u) \|_{\mathcal{L}(H_0^1)} \leq k_1$ for some constant independent of $(q,u) \in D \times X_2$. This implies (C3) (ii). To verify (C3) (iii) we first show that $\{q^\beta : 0 \leq \beta \leq \beta^*\}$ is bounded in H^2, provided the signal-to-noise ratio satisfies

$$\frac{|z|_{H_0^1}}{|z - u_0|_{H_0^1}} > 1, \tag{3.13}$$

and that $\beta^* > 0$ is sufficiently small. With (3.13) holding, there exist $k_2 > 0$ and $\beta^* > 0$ such that

$$|u^\beta|_{H_0^1} \geq |z|_{H_0^1} - |u^\beta - z|_{H_0^1} \geq |z|_{H_0^1} - |u^0 - z|_{H_0^1} - \left[\beta \left(\min_{q^0 \in Q^0} |Nq^0|^2 - \sup_{q^\beta \in Q^\beta} |Nq^\beta|^2 \right)\right]^{\frac{1}{2}} \geq k_2 \text{ for all } \beta \in [0, \beta^*]. \tag{3.14}$$

Here we used Lemma 3.3 (iv).

As before we put $q^\beta = q_1^\beta + q_2^\beta \in \ker N + (\ker N)^\perp$. Due to Lemma 3.3 (i) we know that $\{q_2^\beta\}_{\beta \geq 0}$ is bounded in H^2. Moreover we find

$$\left|q_1^\beta\right| |u^\beta|_{H_0^1} = \left|(-\Delta)^{-1}\left(\nabla \cdot (q_1^\beta \nabla u^\beta)\right)\right|_{H_0^1} \leq \left|(-\Delta)^{-1} \nabla \cdot (q_2^\beta \nabla u^\beta)\right|_{H_0^1} + \left|(-\Delta)^{-1}(u^\beta + f)\right|_{H_0^1},$$

and using (3.14) this implies that $\{q_1^\beta\}$ is bounded in $\mathbf{R}$. Thus $\{|q^\beta|_{H^2} : 0 \leq \beta \leq \beta^*\}$ is bounded. It follows that

$$\| e_u(q^\beta, u^\beta) \|_{\mathcal{L}(H_0^1)} \leq k_3$$

for some constant k_3 independent of $\beta \in [0, \beta^*]$. For $h \in \ker N$ we have

$$\left|e_u^{-1}(q^\beta, u^\beta) e_q(q^\beta, u^\beta) h\right|_{H_0^1} \geq k_3^{-1} |h|_{\mathbf{R}} |u^\beta|_{H_0^1} \geq k_3^{-1} k_2 |h|_{\mathbf{R}},$$

and thus (C3) (iii) is implied by (3.13).

Remark 3.10. The choice of the parameter space in Examples 3.9 and 3.11 is dictated by the requirement that it be contained in the space of L^∞-functions. In fact, H^2 can be replaced by $H^{3/2+\epsilon}$ for any $\epsilon > 0$. In this case the numerical realization of the regularization term can be based on the fast Fourier transform. If in Examples 3.9 and 3.11 the domain Ω is one dimensional, then one chooses $X_1 = H^1, Z_1 = L^2$, and $Nq = q_x$. Since H^1 embeds continuously into L^∞ in dimension 1, conditions (C1)–(C5) can be verified similarly as in Examples 3.9 and 3.11.

Example 3.11. (L^2-output). This example is similar to Example 3.9, with the exception that the H^1-output is replaced by the L^2-output. Thus we analyze

$$\begin{aligned} &\min \tfrac{1}{2} |\mathcal{I}u - z|_{L^2}^2 + \tfrac{\beta}{2} |Nq|^2, \\ &e(q, u) = 0, \end{aligned}$$

with $Z_2 = L_2$, $\mathcal{I}$ the embedding of H_0^1 into L^2 and $z \in L^2$. The spaces X_1, X_2, Y, and Z_1, as well as the set D and the operators e, l_1, l_2 have the same meaning as in Example 3.9. The conditions of Lemma 3.1 are again satisfied and so are (C1), (C2), and (C3) (ii). Turning to (C4) we first note that the range of $\mathcal{I}$ is dense in L^2 and that $\mathcal{I}^* = (-\Delta)^{-1}$ with range $\mathcal{I}^* = H^2 \cap H_0^1$. To verify (C4) (ii) we estimate

$$\begin{aligned} \langle \lambda, b(h, v) \rangle_{H_0^1} &= \langle \nabla \lambda, \nabla(-\Delta)^{-1}(\nabla \cdot (h \nabla v)) \rangle_{L^2} = -\langle \nabla \lambda, h \nabla v \rangle_{L^2} \\ &= \langle \nabla(h \nabla \lambda), v \rangle_{L^2} \leq k_4 |v|_{L^2} |h|_{H^2} |\lambda|_{H^2} \end{aligned}$$

for some constant k_4 independent of $(\lambda, h, v) e (H^2 \cap H_0^1) \times H^2 \times H_0^1$. Since $|\lambda|_{H^2} \leq k_5 |\Delta \lambda|_{L^2}$ for a constant k_5 independent of $\lambda \in H^2 \cap H_0^1$ [LU], we find

$$\langle \lambda, b(h, v) \rangle_{H_0^1} \leq k_4 k_5 |h|_{H^2} |\mathcal{I}v|_{L^2} |(\mathcal{I}^*)^{-1} \lambda|_{L^2}$$

as desired. Let us put $A(q)v = -\nabla \cdot (q \nabla v) + pv$. Then for $v \in L^2$ we find

$$\left(e_u^*(q^\beta, u^\beta)\right)^{-1} \mathcal{I}^* v = A(q)^{-1} \Delta \Delta^{-1} v \in H^2 \cap H_0^1,$$

which implies (C4) (iii). Next we argue the existence of $\beta^* > 0$ such that $\{|q^\beta|_{H^2} : 0 \leq \beta \leq \beta^*\}$ is bounded, provided that

$$\frac{|z|_{L^2}}{|z - k_0|_{L^2}} > 1, \tag{3.15}$$

where u_0 is a solution of (P_0) which is assumed to exist. In fact, (3.15) implies the existence of $\beta^* > 0$ and of $k_6 > 0$ such that

$$\begin{aligned} \left|\mathcal{I}u^\beta\right|_{L^2} &\geq |z|_{L^2} - |\mathcal{I}(u^\beta - z)|_{L^2} \\ &\geq |z|_{L^2} - \left|\mathcal{I}(u^0 - z)\right|_{L^2} - \left[\beta \left(\min_{q^0 \in Q^0} |Nq^0|^2 - \sup_{q^\beta \in Q^\beta} |Nq^\beta|^2 \right)\right]^{\frac{1}{2}} \\ &\geq k_6, \end{aligned}$$

for all $\beta \in [0, \beta^*]$. Again $\{q_2^\beta\}_{\beta>0}$ is bounded in H^2 by Lemma 3.3 and it suffices to argue boundedness of $\{q_1^\beta : 0 \le \beta \le \beta^*\}$ in $\mathbf{R}$. We find

$$\begin{aligned} (3.16) \qquad \left|q_1^\beta\right| |u^\beta|_{L^2} &= \left|(-\Delta)^{-1}\nabla \cdot (q_1^\beta \nabla u^\beta)\right|_{L^2} \\ &\le \left|(-\Delta)^{-1}\nabla \cdot \left(q_2^\beta \nabla u^\beta\right)\right|_{L^2} + \left|(-\Delta)^{-1}\left(u^\beta + f\right)\right|_{L^2}. \end{aligned}$$

Since $\{|q_2^\beta|_{H^2} : \beta > 0\}$ is bounded, it follows that $\{\| (-\Delta)^{-1} A(q_2^\beta) \|_{\mathcal{L}(L^2)} : \beta > 0\}$ is bounded. Together with the boundedness of $\{|u^\beta|_{L^2} : \beta > 0\}$ it follows that the right-hand side of (3.16) can be bounded by a constant k_7. Together with (3.15) this implies that $\{|q_1^\beta| : 0 \le \beta \le \beta^*\}$ and consequently $\{|q^\beta|_{H^2} : 0 \le \beta \le \beta^*\}$ are bounded. Therefore there exists a constant $\tilde{K}_1$ such that

$$\| (\mathcal{I}^*)^{-1} \left(e_u^*(q^\beta, u^\beta)\right)^{-1} \mathcal{I}^* \|_{\mathcal{L}(L^2)} = \| \Delta A(q^\beta)^{-1} \|_{\mathcal{L}(L^2)} \le \tilde{K}_1$$

for all $\beta \in [0, \beta^*]$. This proves (C4) (iv). To verify (C5) let us first observe that there exists a constant $k_7 > 0$ such that

$$\| A^{-1}(q^\beta)\Delta \|_{\mathcal{L}(L^2)} \ge k_7 \text{ for all } \beta \in [0, \beta^*]. \tag{3.17}$$

Moreover, for $h \in \ker N$ we find

$$\begin{aligned} \left|\mathcal{I} e_u^{-1}(q^\beta, u^\beta) e_q(q^\beta, u^\beta) h\right|_{L^2} &= |h| \left|A^{-1}(q^\beta)\Delta u^\beta\right|_{L^2} \\ &\ge k_7 |u^\beta|_{L^2} |h| \ge k_6 k_7 |h|, \end{aligned}$$

and (C5) is proved.

Remark 3.12. Which of the two observations, the H^1- or the L^2-observation, is preferable for estimation of q from u in the equation $e(q, u) = 0$? In the case that z is attainable, so that $u^0 = z$, Propositions 3.5 and 3.7 suggest that the H^1-observation should be preferred since it gives a better convexity estimate at the minimizer. The advantage of the L^2- over the H^1-observation will lie in the case of nonattainability, since we can expect a wider range of applicability of (3.8) than of (3.11). This is not easy to quantify since, in general, one has no estimate on $N(q^\beta) \to N(q^0)$ as $\beta \to o^+$ uniformly as u^0 varies in a neighborhood of z. However, we point out that in the case of full norm regularization, i.e., when $N(q)$ is replaced by $|q|_{H^2}^2$, $\| b \|$ and $|||b|||$ depend only on embedding constants, K_1 depends only on α, $\tilde{K}_1$ depends on α and on $|q^0|_{H^2}$, $K_3 = \tilde{K}_3$ depends on $|q^0|_{H^2}$ and embedding constants, and $K_2, \tilde{K}_2$ can be taken equal to 1. In particular, these constants are independent of $\beta \ge 0$.

4. Representation of the augmented Lagrangian-SQP-methods in the bilinear case. In this section we specialize the algorithms of §2 to bilinear optimization problems of the type (P_β) that were discussed in §3. We mainly concentrate on (2.5). This system contains all relevant operators.

(i) Operator representation of (2.5) in the bilinear case. We use the notation of §3 and introduce the following additional operators:

$$B(u) : X_1 \to Y,\ A(q) : X_2 \to Y,\ \Lambda : X_1 \to Y,\ E : X_1 \to Y,$$

with

$$B(u)h = b(h, u) + l_1(h),\ A(q)v = b(q, v) + l_2(v),$$

and

$$\langle \Lambda h, v\rangle_Y = \langle \lambda, b(h,v)\rangle_Y,\ \langle Eh, v\rangle_Y = \langle e, b(h,v)\rangle_Y$$

for all $h \in X_1, v \in X_2$, and with $q \in X_1, u \in X_2, \lambda \in Y, e \in Y$.

Furthermore, just as in §2, we put

$$\tilde{\lambda} = \lambda + ce(q,u).$$

It is not difficult to calculate the Riesz representation of $\mathcal{L}_c'(q,u)$ in $X_1 \times X_2$. Denoting it by the same symbol as $\mathcal{L}_c'(q,u)$ we find

$$\mathcal{L}_c'(q,u) = \begin{pmatrix} \beta N^*Nq + B^*(u)\lambda \\ \mathcal{I}^*(\mathcal{I}u - z) + A^*(q)\lambda \end{pmatrix} + c \begin{pmatrix} B^*(u)e(q,u) \\ A^*(q)e(q,u) \end{pmatrix}.$$

Using the variable $\tilde{\lambda}$ for the first order update of λ, this expression can be simplified to

$$\mathcal{L}_c'(q,u) = \begin{pmatrix} \beta N^*Nq + B^*(u)\tilde{\lambda} \\ \mathcal{I}^*(\mathcal{I}u - z) + A^*(q)\tilde{\lambda} \end{pmatrix}.$$

For the matrix-operator representation of $\mathcal{L}_c''(q,u)$ one obtains

$$\mathcal{L}_c''(q,u) = \begin{pmatrix} \beta N^*N & \Lambda^* \\ \Lambda & \mathcal{I}^*\mathcal{I} \end{pmatrix} + c \begin{pmatrix} B^*(u) \\ A^*(q) \end{pmatrix} (B(u)\ A(q)) + c \begin{pmatrix} 0 & E^* \\ E & 0 \end{pmatrix},$$

where $E = E(e(q,u))$, and with

$$\tilde{\Lambda} = \Lambda + cE,$$

we find

$$\mathcal{L}_c''(q,u) = \begin{pmatrix} \beta N^*N & \tilde{\Lambda}^* \\ \tilde{\Lambda} & \mathcal{I}^*\mathcal{I} \end{pmatrix} + c \begin{pmatrix} B^*(u) \\ A^*(q) \end{pmatrix} (B(u)\ A(q)). \tag{4.1}$$

The Riesz representation of (2.5) can therefore be expressed as

$$\begin{pmatrix} \begin{pmatrix} \beta N^*N + cB^*B & \tilde{\Lambda}^* + cB^*A \\ \tilde{\Lambda} + cA^*B & \mathcal{I}^*\mathcal{I} + cA^*A \end{pmatrix} & \binom{B^*}{A^*} \\ B \qquad\qquad A & 0 \end{pmatrix} \begin{pmatrix} \hat{q} - q \\ \hat{u} - u \\ \hat{\lambda} - \lambda \end{pmatrix}$$

$$= -\begin{pmatrix} \beta N^*Nq + B^*\tilde{\lambda} \\ \mathcal{I}^*(\mathcal{I}u - z) + A^*\tilde{\lambda} \\ e(q,u) \end{pmatrix}, \tag{4.2}$$

where $A = A(q)$ and $B = B(u)$.

Remark 4.1. (i) The system matrix in (4.2) is characterized by the observation operator $\mathcal{I}$, the regularization operator N, the linearization operators $A(q)$ and $B(u)$, and the operator $\tilde{\Lambda}$ describing the Lagrangian term. For $c = 0$ the left-hand side of (4.2) gives the representation of $M(q,u,\lambda)$ used in Algorithms 1–3. We note that the 2×2 block representing $\mathcal{L}_c''(q,u)$ is independent of (q,u) if $c = 0$.

(ii) From (4.2) it is obvious that changing the observation operator $\mathcal{I}$ implies only minor changes in the numerical realizations of Algorithms 1–3.

(iii) In §2 we showed that (2.5) can be rearranged in such a way that the Hessian of the augmented Lagrangian $\mathcal{L}_c''(x,\lambda)$ is replaced by the Hessian $\mathcal{L}_0''(x,\tilde{\lambda})$. In view of the relative simplicity of (2.5) it may appear that for bilinear systems this rearrangement becomes unnecessary. Below we shall see that for bilinear systems, as well, replacing (2.5) by (2.11) can be advantageous.

(iv) For numerical realizations of (4.1) matrix representations of $(\mathcal{I}, N, A(q), B(u), \tilde{\Lambda})$ need to be calculated. The representation of the adjoint operators then follows easily. In this respect, Lemma 4.2 holds.

LEMMA 4.2. *Let C be a bounded linear operator between real Hilbert spaces X and Y. Let $X^m = \text{span}\,\{\Phi_i\}_{i=1}^n$ and $Y^n = \text{span}\,\{\Psi_i\}_{i=1}^n$ be finite dimensional subspaces of X and Y, respectively. Then the matrix representation $[C]$ of the restriction of C to X^m and the matrix representation $[C^*]$ of C^* restricted to Y^n are given by*

$$\begin{aligned}[C] &= M_2^{-1}S \text{ and } [C^*] = M_1^{-1}S^T, \text{ where}\\ (M_1)_{i,j} &= \langle \Phi_i, \Phi_j\rangle_X,\ (M_2)_{i,j} = \langle \Psi_i, \Psi_j\rangle_Y, \text{ and}\\ (S)_{i,j} &= \langle \Psi_i, C\Phi_j\rangle.\end{aligned}$$

The verification of this lemma is straightforward. We use the lemma frequently to calculate the operator representations in the examples that follow below. We also note that the terms cA^*A and cB^*B in (4.2) lead to discretized operators of the form $cM_1^{-1}\ S^T\ M_2^{-1}\ S$ (in the notation of Lemma 4.2). While the inversion of M_1 can always be avoided by multiplying the complete row of (4.2) by M_1, the inversion of M_2 (or at the least the action of M_2^{-1} on certain elements) is unavoidable. It is here that an advantage of (2.11) over (2.5) becomes obvious, since the terms cA^*A and cB^*B disappear.

(ii) Estimation of the diffusion coefficient with H^1- or L^2-output. We return to the problem of estimating the diffusion coefficient q in

$$\begin{aligned}&-\text{div}\,(q\ \text{grad}\ u) + pu - f = 0 \text{ in } \Omega,\\ &u|\partial\Omega = 0\end{aligned} \tag{4.3}$$

from data z in H_0^1 or in L^2, respectively. First the H^1-output case is considered and the assumptions of Example 3.9 are assumed to hold. For $q \in H^2$, let $-\tilde{D}^2 q$ denote the element of $(H^2)^*$ defined by

$$h \to \langle \nabla q, \nabla h\rangle_{L^2} + \sum_{1\le i,j\le 2} \langle D_{x_i}D_{x_j}q, D_{x_i}D_{x_j}h\rangle_{L^2}.$$

Furthermore, let $\mathcal{N} : (H^2)^* \to H^2$ denote the solution operator to the variational equation

$$\langle u, \varphi\rangle_{H^2} = \langle f, \varphi\rangle_{(H^2)^*, H^2} \text{ for all } \varphi \in H^2,$$

with $f \in (H^2)^*$. Employing this notation the system matrix in (4.2) for $c = 0$ is given by

$$\begin{pmatrix} -\beta\mathcal{N}\tilde{D}^2 & \mathcal{N}(\nabla\lambda\cdot\nabla\bullet) & \mathcal{N}(\nabla u\cdot\nabla\bullet)\\ -(-\Delta)^{-1}\nabla(\bullet\nabla\lambda) & I & (-\Delta)^{-1}\left[-\nabla(q\nabla\bullet) + p\bullet\right]\\ -(-\Delta)^{-1}\nabla(\bullet\nabla u) & (-\Delta)^{-1}\left[\nabla(q\nabla\bullet) + p\bullet\right] & 0\end{pmatrix}$$

and the right-hand side of (4.2) becomes

$$\begin{pmatrix} -\beta\mathcal{N}\tilde{D}^2 + \mathcal{N}(\nabla\lambda\cdot\nabla u)\\ u - z + (-\Delta)^{-1}(-\nabla\cdot(q\nabla\lambda) + p\lambda)\\ e(q,u)\end{pmatrix}.$$

The resulting equation is an operator equation from $H^2 \times H_0^1 \times H_0^1$ into itself. Multiplying the first row by $\mathcal{N}^{-1}$ and the second and third by $(-\Delta)$ we obtain

$$\begin{pmatrix} -\beta\tilde{D}^2 & \nabla\lambda\cdot\nabla\bullet & \bar{B}^*(x) \\ -\nabla(\bullet\nabla\lambda) & -\Delta & \bar{A}(z) \\ \bar{B}(u) & \bar{A}(q) & 0 \end{pmatrix} \begin{pmatrix} \hat{q}-q \\ \hat{u}-u \\ \hat{\lambda}-\lambda \end{pmatrix}$$

$$= -\begin{pmatrix} -\beta\tilde{\Delta}^2 q + \bar{B}^*\lambda \\ (-\Delta)(u-z) + \bar{A}^*(q)\lambda \\ -\nabla\cdot(q\nabla u) + pu + f \end{pmatrix}, \tag{4.4}$$

where the system matrix acts as follows:

$$H^2 \times H_0^1 \times H_0^1 \to (H^2)^* \times H^{-1} \times H^{-1}$$

and

$$\bar{A}(q)u = -\nabla\cdot(q\nabla u) + pu,\ \ \bar{B}(u)h = -\nabla\cdot(h\nabla u).$$

The operator $\bar{A}(q) : H_0^1 \to H^{-1}$ is selfadjoint, and the adjoint of $\bar{B}(u) : H^2 \to H^{-1}$ is given by

$$\bar{B}^* w = \nabla u \cdot \nabla w.$$

If $c > 0$, then λ must be replaced by $\tilde{\lambda}$ in (4.4), and the following matrix must be added to the upper left 2×2 submatrix of the system matrix in (4.4):

$$c \begin{bmatrix} \bar{B}(u)^* \\ \bar{A}(q)^* \end{bmatrix} (-\Delta)^{-1} \begin{bmatrix} \bar{B}(u) & \bar{A}(q) \end{bmatrix}. \tag{4.5}$$

If, as in Example 3.11, the H_0^1-output is replaced by the L^2-output norm then the only change that results for the system (4.4) is the replacement of the circled $-\Delta$ in the second row by the identity operator.

(iii) Estimation of the potential. We also carried out numerical experiments for the estimation of the coefficient characterizing the potential in elliptic boundary value problems. Specifically we considered:

$$\begin{aligned} &\min \tfrac{1}{2}\,|u-z|^2_{H_0^1} + \tfrac{\beta}{2}\,|\nabla q|^2_{L^2}\,, \\ &e(q,u) = (-\Delta)^{-1}(-\Delta u + qu + f) = 0, \end{aligned} \tag{4.6}$$

where the function spaces are considered over a domain $\Omega \subset \mathbf{R}^n$, with Lipschitzian boundary. The estimation of the potential is more ill posed than that of the diffusion coefficient. In the notation of §3 we put

$$\begin{aligned} X_1 \times X_2 = H^1 \times H_0^1,\ Z_2 = Y = H_0^1,\ \mathcal{I} = \text{identity},\ z \in H_0^1, \\ Nq = \nabla q,\ Z_1 = \otimes_{i=1}^n L^2, \\ D = \{q \in H^1 : q(x) \geq 0\}, \\ b(q,u) = (-\Delta)^{-1}(qu),\ l_1 = 0,\ l_2(u) = u,\ 0 \neq f \in H^{-1}. \end{aligned}$$

For $q \in H^1$, let $-\tilde{D}^2 q \in (H^1)^*$ denote the functional

$$h \to \langle \nabla q, \nabla h \rangle_{L^2} \text{ for all } h \in H^1.$$

For $c = 0$, (4.2) can be expressed for the present problem as

$$\begin{pmatrix} -\beta\tilde{D}^2 & \lambda & u \\ \lambda & -\Delta & (-\Delta+q) \\ u & (-\Delta+q) & 0 \end{pmatrix} \begin{pmatrix} \hat{q}-q \\ \hat{u}-u \\ \hat{\lambda}-\lambda \end{pmatrix}$$

$$= -\begin{pmatrix} -\beta\tilde{D}^2 q + \lambda u \\ -\Delta(u-z) \quad -\Delta\lambda + q\lambda \\ -\Delta u + qu + f \end{pmatrix}, \tag{4.7}$$

with the system matrix acting as follows:

$$H^1 \times H_0^1 \times H_0^1 \to (H^1)^* \times H^{-1} \times H^{-1}.$$

If $c > 0$, then λ must be replaced by $\tilde{\lambda}$ in (4.8) and the matrix corresponding to (4.6), given by

$$\begin{pmatrix} u(-\Delta)^{-1}(u\bullet) & u(-\Delta)^{-1}(-\Delta+q) \\ (-\Delta+p)(-\Delta)^{-1}(u\bullet) & (-\Delta+p)(-\Delta)^{-1}(-\Delta+q) \end{pmatrix}, \tag{4.8}$$

must be added to the upper left 2×2 submatrix of the system matrix in (4.7). In the case of L^2-ouput the circled $-\Delta$ operators must be replaced by the identity operator.

Let us close this section with a final comment on the relative merit of the rearrangement of (2.5) as (2.11). The advantage of the system matrices (4.4) and (4.7) that arise for $c = 0$ over those where (4.5) and (4.8) are added to the system matrix when $c > 0$ lies in the fact that $(-\Delta)^{-1}$ is not required. However, we should note that for $c = 0$ the left upper 2×2 blocks in (4.5) and (4.7) are singular! (Choose, for example, the q-coordinate as constant and the u-coordinate zero.) Thus to solve (4.4) or (4.7) the method of first eliminating the (q, u)-variable for the λ-variable may not be well defined, since it requires the inversion of the upper left 2×2 block. If the 2×2 blocks (4.5) and (4.8) are added, then for λ sufficiently small (see §3) the resulting systems are special cases of (2.5) and can be solved by first eliminating the (q, u)-variable; see (2.7).

If the seminorm regularization term is replaced by a full norm, then the upper left 2×2 blocks in (4.4) and (4.7) are positive definite for λ sufficiently small (compare Corollary 3.6), without adding the 2×2 matrices (4.5) and (4.8).

5. Numerical experiments. This section is devoted to computational aspects of augmented Lagrangian-SQP-methods for parameter estimation in elliptic boundary value problems. These problems are well known to be ill posed. Moreover, one cannot assume availability of a good initial guess for the unknown parameter. To account for ill-posedness, a seminorm regularization term as discussed in §§3 and 4 is used. The lack of a good initial guess for the unknown parameter does not represent a significant obstacle for the problems under consideration. First, the minimization is carried out with respect to the two variables q and u, and z (or an interpolation of pointwise data) serves as a good initial guess for the u-variable. Second, concerning the unknown parameter variable q, we shall see that each of the variants of the algorithm that we employ contains an equation error step. This equation error part of the algorithm provides a very good approximation of the unknown optimal parameter variable already in the first iteration of the algorithm. Here the bilinear structure of the problem is essential.

All algorithms that we implemented are variants of the following model algorithm.

MODEL ALGORITHM.

(i) Choose $\lambda_1 = 0$, $\bar{c} \geq 0$, $c \in (\bar{c}, \infty)$, $\sigma \in (0, c - \bar{c}]$, $n_1 \in \mathbf{N}$, $\beta > 0$.
Phase 1.

For $n = 1, \ldots, n_1$, do:
(ii) solve numerically for x_n

$$\min \mathcal{L}_c(x, \lambda_n) \text{ over } X,$$

(iii) set $\lambda_{n+1} = \lambda_n + \sigma e(x_n)$, end do.
Phase 2.

Given (x_{n_1}, λ_{n_1}), for $n = n_1 + 1$ until stopping criterion is reached do:
(iv) set $\tilde{\lambda} = \lambda_n + \sigma e(x_n)$,
(v) solve for (x, λ) :

$$M(x_n, \tilde{\lambda}) \begin{pmatrix} x - x_n \\ \lambda - \tilde{\lambda} \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_0'(x_n, \tilde{\lambda}) \\ e(x_n) \end{pmatrix},$$

and set $(x_{n+1}, \lambda_{n+1}) = (x, \lambda)$.

Remark 5.1. (a) The necessary optimality conditions for the minimization problem in (ii) reads as follows:

$$\beta N^* N q + B^*(u)\lambda_n + c\, B^*(u)\, e(q, u) = 0, \tag{5.1}$$

$$\mathcal{I}^* \mathcal{I}(u - z) + A^*(q)\lambda_n + cA^*(q)e(q, u) = 0, \tag{5.2}$$

with $A(q)$ and $B(u)$ defined in §4 (i).

(b) Under appropriate assumptions Phase 1 (which serves the purpose of a stable startup algorithm) is linearly convergent [IK], whereas Phase 2 is quadratically convergent. For the problems that we tested, only the cases $n_1 = 1$ and $n_1 = 0$ (no initialization) turned out to be relevant. For $n_1 = 1$ we carried out the minimization step (ii) in Phase 1 by solving (5.1) and (5.2) only once replacing u by (an interpolation of) z in (5.1), i.e., we solved

$$\beta N^* N q + c\, B^*(z) e(q, z) = 0 \text{ for } q_1, \text{ and} \tag{5.1'}$$

$$\mathcal{I}^* \mathcal{I}(u - z) + cA^*(q_1)e(q_1, u) = 0 \text{ for } u_1. \tag{5.2'}$$

We note that (5.1') is a regularized equation error step. In fact, for $\beta = 0$, (5.1') becomes the equation error equation $e(q, z) = 0$.

(c) If we skip the initialization phase, then the first time that the system in (v) is solved it also contains the equation-error equation in the special case that $\beta = \sigma = 0$, and for the choice $x_0 = (q_0, z)$ with q_0 arbitrary. In fact we find for (v)

$$\begin{pmatrix} 0 & 0 & B^*(z) \\ 0 & \mathcal{I}^*\mathcal{I} & A^*(q_0) \\ B(z) & A(q_0) & 0 \end{pmatrix} \begin{pmatrix} q - q_0 \\ u - z \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ e(q_0, z) \end{pmatrix}. \tag{5.3}$$

The last equation in (5.3) is found to be equivalent to

$$e(q, z) + A(q)(u - z) = 0,$$

which is independent of q_0. If the equation-error equation $e(q,z)=0$ has a solution q_1, then $(q_1,z,0)$ is a solution of (5.3). Of course, (5.3) is not appropriate for numerical computations. It indicates, however, why we can expect to obtain convergence even for a bad initial guess of the q-variable.

(d) In view of (3.6) in Lemma 3.4, $\lambda=0$ is an effective choice for the initial guess for the Lagrange multipler.

(e) Phase 2 corresponds to Algorithm 3 of §2. Appropriate modifications can be made to realize the auxiliary optimization problems corresponding to steps (ii) and (ii') of Algorithms 1 and 2 (e.g., by iterating the necessary conditions (5.1) and (5.2)). However, they do not significantly influence the behavior of the algorithms for the examples presented below.

(f) Concerning the choice of the regularization parameter β we proceeded empirically in the examples given below. For a systematic choice of β we refer to [IK] and the references given there.

The numerical realization of the model algorithm is straightforward. Following a Galerkin approach one chooses subspaces Q^N and H^N for the approximation of the parameter space X_1 and the state space X_2. In our calculations elements in Y and Z_2 were also approximated by elements in H^N and elements in Z_1 by elements from Q^N. Then one calculates matrix representations for $(\mathcal{I},N,A(q),B(u),\tilde{\Lambda})$, and approximations to $e(q,u),f$ and z. Upon assembling these quantities in the system matrix, appropriate modifications (e.g., multiplication by mass matrices) must be made to obtain the simplest possible linear algebra systems; compare (4.2) to (4.5) in this respect.

In this paper we give one- and two-dimensional test examples. Let us turn to one-dimensional examples first. We set $\Omega=(0,1)$ and the spaces Q^N and $H^N\subset H_0^1$ are chosen as piecewise constant functions and as linear spline functions with respect to the grid $\{\frac{i}{N}\}_{i=0}^N$, respectively. While Q^N is not a subspace of $H^1(=X_1)$, all required matrix representations will be well defined except for that of the regularized term $q\to\frac{\beta}{2}|q_x|^2$, where a discrete derivative formula is used. For the estimation of the diffusion coefficient as well as of the potential H^1-seminorm regularization $q\to\frac{\beta}{2}|q_x|^2$ is used; compare Remark 3.10. The observation operator is the identity operator.

First we turn to the estimation of q in

$$\begin{aligned}&-(qu_x)_x=f \text{ in } (0,1),\\ &u(0)=u(1)=0\end{aligned}$$

with f given. Unless specified otherwise the unperturbed data z^* are given by

$$z^*(x)=e^{-x}\sin(2\pi x) \text{ for } x\in(0,1), \tag{5.4}$$

and the perturbed data are a linear interpolation of $z(0)=0, z(1)=0$, and

$$z\left(\frac{i}{N}\right)=z^*\left(\frac{i}{N}\right)+\delta_i,\ i=1,\dots,N-1, \tag{5.5}$$

where δ_i are uniformly distributed random numbers in $[-\delta,\delta]$. The unknown coefficient is chosen to be

$$q^*(x)=1.5+.5\arctan(1000(x-.4))$$

and $f^*=-(q^*z_x^*)_x$. For these specifications we now describe a series of computer runs. Unless specifically stated otherwise we choose $N=64$ and put

$$e^*(n)=|q_n-q^*|_{L^2}+|u_n-z^*|_{L^2} \text{ for } n=1,\dots,$$

$$e(n) = |q_n - q_{n-1}|_{L^2} + |u_n - u_{n-1}|_{L^2} \quad \text{for } n = 2, \ldots,$$

where $(q_n, u_n) = x_n$.

RUN 1. This is the augmented Lagrangian algorithm with first order update used in Phase 1, with $n_1 = 7$. We put

$$\delta = 0, \quad \beta = 10^{-6}, \quad c = 1,$$

and found the results shown in Table 1.

TABLE 1

n	$10^{-3} \cdot e(n)$	$10^{-3} \cdot e^*(n)$	$\frac{e(n-1)}{e(n)}$
1		0.1332	
2	0.0522	0.1601	
3	0.0266	0.1826	0.5107
4	0.0162	0.1969	0.6089
5	0.0098	0.2056	0.6048
6	0.0057	0.2107	0.5841
7	0.0033	0.2137	0.5740

For problems with fast convergence and small value for δ, the column e^* is not very significant, since q^* is not the solution of the discretized problem! We include here the results for e^* to allow comparison with later runs.

RUN 2. This is the augmented Lagrangian-SQP-algorithm with first order startup $n_1 = 1$ of the model algorithm. We put

$$\delta = 0, \quad \beta = 10^{-6}, \quad c = 1$$

and found the results shown in Table 2.

TABLE 2

n	$10^{-3} \cdot e(n)$	$10^{-3} \cdot e^*(n)$	$10^{-14} \cdot \frac{e(n)}{e(n-1)^2}$
1		0.0657	
2	0.21795724187	0.2189	
3	0.00050819426	0.2178	0.00000000000011
4	0.00000006374	0.2178	0.00000000000247
5	0.00000000002	0.2178	0.00000006770840
6	0.00000000001	0.2178	0.18753175007366
7	0.00000000002	0.2178	1.05210368751625
8	0.00000000001	0.2178	0.21724093557351

We observe that $\frac{e(n)}{e(n-1)^2}$ is not bounded. This should be attributed to the fact that the initialization phase is almost a one-step algorithm since $\delta = 0$. In view of (5.1'), the only errors are due to $\beta > 0$ and numerical error. We should also note that the algorithm does not converge with $c = 0$. The algorithm, however, gave very similar results, if c was taken to be any value in [1,1000].

Concerning the solution of the discretized system in (v) of the model algorithm there is essentially no difference with respect to accuracy and computing time whether or not one uses block inversion (in an analogous manner as (2.7) is obtained from (2.5)). We also tested the implementation in which steps (iv) and (v) are replaced by (2.5). This does not change the accuracy but the computing time is increased by a factor of 1.9.

RUN 3. For this run the specifications are given by

$$\delta = .01, \quad \beta = 10^{-3}, \quad c = 1,$$

and the results are shown in Table 3.

TABLE 3

n	$e(n)$	$e^*(n)$	$10^{-6} \cdot \frac{e(n)}{e(n-1)^2}$
1		0.2354	
2	0.11121370645386	0.1112	
3	0.01797890751182	0.1025	0.00000145
4	0.00639561197134	0.1019	0.00001978
5	0.00064624291258	0.1017	0.00001579
6	0.00000039769060	0.1017	0.00000095
7	0.00000000004792	0.1017	0.00030298
8	0.00000000000001	0.1017	2.69728345

The behavior of $\frac{e(n)}{e(n-1)^2}$ is distinctly different here from RUN 1. This is due to the fact that $\delta > 0$, and thus the algorithm is much further from being a one-step algorithm.

RUN 4. Now we test the behavior of the model algorithm, if Phase 1 is deleted. The specifications are

$$\delta = 0, \quad \beta = 10^{-5}, \quad c = 10.$$

The results are shown in Table 4.

TABLE 4

n	$e(n)$	$e^*(n)$	$10^{-2} \cdot \frac{e(n)}{e(n-1)^2}$
1		0.268510	
2	0.26851036898427	0.330837	
3	0.21336410505937	0.348882	0.030
4	0.64389381573612	0.150978	0.141
5	0.30729604016148	0.039138	0.007
6	0.13525351307961	0.005875	0.014
7	0.03959112582818	0.000223	0.021
8	0.00582535926526	0.000218	0.037
9	0.00005958566351	0.000218	0.018
10	0.00000000673218	0.000218	0.019
11	0.00000000000004	0.000218	8.930

Comparing the result of RUN 1 we note that it takes more iterations before quadratic convergence is obtained. It also takes eight iterations before a constant distance to q^* is obtained. For $c = 1$ (or smaller) the algorithm does not converge. For large c the convergence is faster than for $c = 10$; compare RUN 7.

RUN 5. We carried out several runs comparing the seminorm regularization term $\beta|q_x|^2_{L^2}$ to full norm regularization $\beta|q|^2_{H^1}$. The results are in favor of seminorm regularization, but the difference is not significant. For z^* as in (5.4), $q^* = 5 + .5 \arctan(1000(x - .4))$, and

$$\delta = .01, \quad \beta = 10^{-3}, \quad c = 1000,$$

we obtain

$$e^*(8) = .1631 \text{ for seminorm regularization and}$$
$$e^*(8) = .1745 \text{ for fullnorm regularization.}$$

RUN 6. We compared the behavior of Phase 2 (as used in RUN 4), to the algorithm that arises from iterating (2.5), which does not employ a first order update involving $\tilde{\lambda}$. The algorithm converges for a wide range of small c-values ($c \in [10^{-6}, 1]$) with almost identical results to those in RUN 4, but the computing time is almost twice as long. This is probably due to the necessity of inverting the Laplacian in $\mathcal{L}_c''$ as explained in §3.

RUN 7. The best algorithms from the class of algorithms that we discussed in this paper are the model algorithm with one loop of the Phase 1 or just Phase 2 alone. We give a comparison of these two algorithms with

$$\delta = .01, \quad \beta = 10^{-3}, \quad c = \begin{cases} 1 & \text{if } n_1 = 1, \\ 1000 & \text{if } n_1 = 0. \end{cases}$$

The computing times shown in Table 5 are almost identical.

TABLE 5

n	$n_1 = 1, e(n)$	$n_1 = 0, e(n)$	$n_1 = 1, e^*(n)$	$n_1 = 0, e^*(n)$
1			0.11121	0.23736
2	0.11121370645386	0.23736093213904	0.01251	0.22567
3	0.01797890751182	0.11366221652847	0.10192	0.13401
4	0.00639561197134	0.11311072424741	0.10172	0.10283
5	0.00064624291258	0.05575026800434	0.10172	0.10172
6	0.00000039769060	0.00285011982705	0.10172	0.10172
7	0.00000000004792	0.00000153927526	0.10172	0.10172
8	0.00000000000001	0.00000000000058	0.10172	0.10172

Remark 5.1. We made numerous runs comparing the L^2- to the H_0^1-output norm, both for the model algorithm with $n_1 = 1$ and for the first order augmented Lagrangian algorithm of Phase 1. Generally the H^1-output norm was superior to the L^2-output norm. This is in contrast to the calculations in [E], where the L^2-output norm was found to be superior to the H^1-output in the presence of noise for the augmented Lagrangian algorithm. We have no definite conclusion at this moment, but expect that the type of discretization plays some role. The results in [E] are based on a discretization by linear splines for both the state and the parameter variable.

We also carried out experiments for the estimation of q in

$$\begin{aligned} -u_{xx} + qu &= f \text{ in } (0,1), \\ u(0) = u(1) &= 0. \end{aligned} \tag{5.6}$$

The unperturbed and the perturbed observations were chosen as in (5.4), (5.5), and $q^*(x) = 1.5 + .5 \arctan(30(x - .4))$. All other specifications are those chosen above for the estimation of the diffusion coefficient. In many aspects the model algorithm and its variations behaved similarly for the problem of estimating q in (5.6) as for estimating the diffusion coefficient and therefore we only point out some differences that we observed.

(i) Estimating the potential is a more ill posed problem than estimating the diffusion coefficient and therefore only a smaller noise level can be admitted for the potential than for the diffusion estimation problem.

(ii) In the model algorithm with $n_1 = 1$ one can set c very small or even zero for the higher iterates $n = 2, \ldots$, without affecting quadratic convergence.

(iii) In the model algorithm with $n_1 = 0$, it is necessary to choose $c > 0$. Convergence is achieved for a much wider range of c values if the L^2-output criterion is used $\left(c \in [10^{-10}, 1]\right)$, then if an H_0^1-output criterion is used.

(iv) The L^2-output criterion is preferable to the H_0^1-output criterion in the presence of noise ($\delta > 0$).

We close by giving results for the two-dimensional problem of estimating q in

$$\begin{aligned} &-\text{div}\,(q\ \text{grad}\ u) = 1 \text{ in } \Omega, \\ &u|\partial\Omega = 0, \end{aligned} \tag{5.7}$$

where $\Omega = [0,1] \times [0,1]$. The unknown coefficient was chosen to be

$$q^*(x,y) = 1 + \frac{1}{2}\left(x^2 + y^2\right).$$

For this choice of q^* the solution $u(q^*)$ of (5.7) was calculated numerically on a grid twice as fine as the grid that was used to solve the parameter estimation problem. The data z were obtained by adding uniformly distributed random numbers in $[-.01, .01]$ to the solution $u(q^*)$ at the nodal points of the discretization for the estimation problem. This results in a maximum relative error at the nodal points of 17%.

The discretization is carried out by central finite differences on a uniform grid with meshsize $\frac{1}{15}$ for the state variable u and by step functions on the same grid for the coefficients. The model algorithm with $n_1 = 1$, $c = 1$, and $\beta = 10^{-3}$ was employed. The numerical results which show quadratic convergence are given in Table 6.

TABLE 6

n	$e(n)$	$e^*(n)$	$10^{-2} \cdot \frac{e(n)}{e(n-1)^2}$
1		0.05318	
2	0.01972181576474	0.03950	
3	0.00993833288902	0.03141	0.256
4	0.01203661826238	0.02103	1.219
5	0.01151282001372	0.01217	0.795
6	0.00571637843059	0.00968	0.431
7	0.00095779140699	0.00953	0.293
8	0.00002823314280	0.00952	0.308
9	0.00000001925428	0.00952	0.242
10	0.00000000000001	0.00952	0.251

REFERENCES

[B] D.P. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[BTZ] R.H. BYRD, R.A. TAPIA, AND Y. ZHANG, *An* SQP*-augmented Lagrangian* BFGS *algorithm for constrained optimization*, Rice University, Houston, TX, preprint.

[CK] F. COLONIUS AND K. KUNISCH, *Output least squares stability in elliptic systems*, Appl. Math. Optim., 19 (1989), pp. 33-63.

[E] W. EGARTNER, *Augmentierte Lagrange Verfahren und deren Anwendung auf Inverse Probleme mit* H^1*- und* L^2 *-Beobachtungsnorm*, Master's Thesis, Technical University of Graz, 1991.

[IK] K. ITO AND K. KUNISCH, *The augmented Lagrangian method for parameter estimation in elliptic systems*, SIAM J. Control Optim., 28 (1990), pp. 113–136.

[IKK] K. Ito, M. Kroller, and K. Kunisch, *A numerical study of the augmented Lagrangian method for the estimation of parameters in elliptic systems*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 884–910.

[KK] A. Kölbl and Ch. Kredler, *The second order update for the augmented Lagrangian avoiding explicit inverses*, T.U. München, 1990, preprint.

[KW] C.T. Kelley and S.J. Wright, *Sequential quadratic programming for certain parameter identification problems*, Math. Programming, 51 (1991), pp. 281–305.

[KS] K. Kunisch and E. W. Sachs, *Reduced* SQP*-methods for parameter identification problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1793–1820.

[LU] O. Ladyzhenskaya and N. Ural'tseva, *Linear and Quasilinear Elliptic Equations*, Academic Press, New York, 1968.

[M] V.G. Maz'ja, *Sobolev Spaces*, Springer-Verlag, Berlin, 1985.

[PT] V.T. Poljak and N.Y. Tret'yakov, *The method of penalty estimates for conditional extremum problems*, Zh. Vychisl. Mat. i Mat. Fiz., 13 (1973), pp. 34–46.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 126–163, February 1996
© 1996 Society for Industrial and Applied Mathematics
008

AN INFINITE-DIMENSIONAL CONVERGENCE THEORY FOR REDUCED SQP METHODS IN HILBERT SPACE*

F.-S. KUPFER†

Abstract. We develop a general convergence theory for a class of reduced successive quadratic programming (SQP) methods for infinite-dimensional equality constrained optimization problems in Hilbert space. The methods under consideration approximate second order information of the Lagrangian restricted to the null space of h', the derivative of the constraints. In particular, a sufficient condition for two-step superlinear convergence will be given, and a general result for secant methods will be established which can serve as a tool to prove superlinear convergence in an infinite-dimensional framework. This result is applied to prove local two-step superlinear convergence of reduced SQP methods under the assumption that exact reduced second order information is known up to a compact perturbation. To test our convergence theory we consider optimal control problems, and we formulate a reduced quasi-Newton algorithm which presents a new approach to an efficient solution of these problems. Our main motivation for studying reduced SQP is that these methods allow us to exploit special structure of the null space of h' which usually leads to more reasonable storage requirements. Such a structure is often the result of a natural separability of variables which appears frequently in infinite-dimensional applications, as for example in optimal control and in parameter identification problems. This special structure suggests the use of a representation of the null space of h' which is nonstandard in the reduced SQP context.

Key words. reduced successive quadratic programming, quasi-Newton methods, infinite-dimensional optimization, equality constraints, null space parametrization, two-step superlinear convergence, compact operators, optimal control

AMS subject classifications. 65J15, 47H17, 49D15

1. Introduction.

In this paper we study a class of quasi-Newton methods to solve infinite-dimensional minimization problems with equality constraints. Let X be a real Hilbert space, Z be a Banach space, and $f : X \to \mathbb{R}$, $h : X \to Z$ be twice Fréchet-differentiable functions. Then the problem is formulated as follows:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0\,. \end{array} \tag{1.1}$$

The Lagrangian function associated with problem (1.1) is the function

$$L(x,l) = f(x) - l(h(x))\,, \quad x \in X\,,\ l \in Z^*\,, \tag{1.2}$$

where l is called the Lagrange multiplier and Z^* denotes the dual space of Z. We use $\hat{x}$ to denote a local solution of problem (1.1) and $\hat{l}$ for its associated Lagrange multiplier, i.e., the $\hat{l}$ that satisfies $\nabla_x L(\hat{x},\hat{l}) = 0$. For normed linear spaces E and F we let $\mathcal{L}(E,F)$ denote the space of linear and bounded operators from E into F, and we set $\mathcal{L}(E) := \mathcal{L}(E,E)$.

As far as finite-dimensional problems are concerned, successive quadratic programming (SQP) methods are among the most successful methods for solving (1.1). At each iteration of an SQP method a quadratic approximation of the Lagrangian is minimized subject to linearized constraints. In addition, an approximation of the

* Received by the editors March 20, 1991; accepted for publication (in revised form) August 4, 1994.

† Universität Trier, FB IV–Mathematik, 54286 Trier, Germany. This research was supported by the Deutsche Forschungsgemeinschaft.

Lagrange multiplier is calculated. Let x_k denote the current iterate. Then the SQP x-step d_k is computed from the solution of the following quadratic subproblem:

$$\begin{array}{ll} \text{minimize} & \langle \nabla f(x_k), d\rangle + \frac{1}{2}\langle d, M_k d\rangle \\ \text{subject to} & h'(x_k)d + h(x_k) = 0\,. \end{array} \tag{1.3}$$

The new iterate is $x_{k+1} = x_k + d_k$. Since the computation of second derivatives may be too expensive, it is reasonable to approximate $L''_x(x_k, l_k)$ by a self-adjoint operator $M_k \in \mathcal{L}(X)$. Secant updates are commonly used for M_k in finite-dimensional implementations. For a study of various SQP methods for the solution of finite-dimensional problems we refer the reader to Han [22], Powell [37], and Tapia [40]–[42].

In infinite-dimensional spaces there has been some progress on the SQP Newton method, which uses exact second order information in (1.3). Machielsen [34] has implemented this method successfully for the solution of optimal control problems with state constraints. Convergence properties are not investigated. Alt [1] applies the method to problem (1.1) in reflexive Banach spaces. Moreover, as in [34], the feasible set is restricted by the additional constraint $g(x) \in K$ for a closed convex cone K. It is shown that the method is locally and quadratically convergent. Hwang and Kelley [25] and Kelley and Wright [29] apply the SQP Newton method to parameter identification problems, and Sachs [38] studies SQP methods with quasi-Newton updates for optimal control problems. To our knowledge, theoretical convergence results for SQP quasi-Newton methods in infinite-dimensional spaces are not available up to now.

One problem with SQP algorithms which use quasi-Newton updates is that the use of positive definite secant updates is not in line with the second order sufficiency condition. Alternative formulations of SQP methods have been proposed in order to circumvent the lack of positive definiteness of the Hessian of the Lagrangian at the solution. We are interested in one of these approaches which leads to a class of methods known in the finite-dimensional case as reduced SQP (or projected Hessian) methods. In contrast to full SQP methods, reduced SQP methods approximate the Hessian restricted to the null space of the Jacobian of the constraints. This is a more natural approach since only the reduced Hessian enters into the optimality conditions for problem (1.1). But there is another advantage of reduced methods which becomes even more evident in infinite-dimensional applications. Updating reduced second derivative information offers a reduction in dimension which can lead to enormous savings in storage for discretized implementations of the infinite-dimensional algorithms. This attractive property is our main motivation for studying reduced methods in a Hilbert space framework. Before describing the effect of reduction in the context of optimal control problems, we give a description and motivation of reduced SQP methods, and present some finite-dimensional convergence results.

Let x_k be the current iterate in an SQP method. Suppose that $h'(x_k)$ is surjective and that (1.3) has a unique solution $d_k \in X$. In order to obtain a Hilbert space setting for reduced methods we assume that the null space of $h'(x_k)$ can be represented in the following way:

$$\mathcal{N}(h'(x_k)) := \{s \in X : h'(x_k)s = 0\} = \{T_k w : w \in W\} =: \mathcal{R}(T_k)\,,$$

where $W \neq \{0\}$ is a real Hilbert space and $T_k \in \mathcal{L}(W, X)$ is injective. For applications in optimal control we demonstrate at the end of this section that the structure of $\mathcal{N}(h'(x_k))$ allows a very natural definition of the operator T_k. In the finite-dimensional case, T_k is any basis for the intersection of the kernels of the gradients of the constraints.

Now, let $q_k \in X$ be any solution of the linearized constraint $h'(x_k)q = -h(x_k)$. Then an easy calculation shows that the SQP step can be written in the following way:

$$d_k = q_k + T_k w_k \,, \tag{1.4}$$

where $w_k = w_k(q_k) \in W$ is the solution of the following unconstrained problem in W

$$\text{minimize} \quad \langle \nabla f(x_k), T_k w \rangle + \langle T_k w, M_k q_k \rangle + \frac{1}{2} \langle T_k w, M_k T_k w \rangle \,.$$

In particular, let $R_k \in \mathcal{L}(Z, X)$ be any right-inverse of $h'(x_k)$, i.e., $h'(x_k)R_k = I_Z$. (Note that a right-inverse exists, since $h'(x_k)$ is assumed to be surjective.) With the choice $q_k = -R_k h(x_k)$ in (1.4) we obtain the following closed form expression for the SQP step (compare Gabay [14] for the finite-dimensional case):

$$d_k = -T_k [T_k^* M_k T_k]^{-1} T_k^* (\nabla f(x_k) - M_k R_k h(x_k)) - R_k h(x_k) \,. \tag{1.5}$$

Here T_k^* denotes the adjoint operator of T_k. By formula (1.5) one can interpret the direction d_k as a combination of a restoration step $-R_k h(x_k)$, i.e., a right-inverse Newton step for solving the equality constraint $h(x) = 0$, and a minimization step belonging to $\mathcal{N}(h'(x_k))$. This terminology is used by Gilbert [17].

To motivate reduced SQP methods we consider for a moment linear constraints $h(x) = Ax + b$, where $A \in \mathcal{L}(X, Z)$ is surjective and $b \in Z$. In this case all iterates x_k generated from an SQP method are feasible except possibly the starting point. Consequently, in (1.5) the expression $M_k R_k h(x_k)$ vanishes so that it is reasonable to replace $T_k^* M_k T_k$ by an operator that approximates the reduced Hessian $T_k^* L_x''(x_k, l_k) T_k$ directly. If the idea to approximate only the reduced Hessian while discarding the $M_k R_k h(x_k)$ term is applied to the case of general constraints, one is led to the reduced SQP step:

$$p_k = -T_k B_k^{-1} T_k^* \nabla f(x_k) - R_k h(x_k) \,. \tag{1.6}$$

Here $B_k \in \mathcal{L}(W)$ is interpreted as an approximation to reduced Hessian $T_k^* L_x''(x_k, l_k)$ T_k. By a reduced SQP method we mean the iteration $x_{k+1} = x_k + p_k$ with p_k from (1.6).

Obviously, there is a difference between reduced and full SQP methods concerning the role of the right-inverse and the null space parametrization. For a given operator B_k the reduced SQP step (1.6) generally varies with different choices of R_k and T_k. In contrast, if M_k is given, then the full SQP step is the unique solution of a quadratic subproblem and (1.5) is just one representation of this solution. In a reduced method the right-inverse and the parametrization of $\mathcal{N}(h'(x_k))$ are optional parameters yielding different algorithms while the representation (1.5) of the full SQP step is invariant to changes in R_k and T_k, at least theoretically.

For finite-dimensional problems, Gabay [14] introduced the reduced SQP step in the general formulation (1.6). He gives a different development which shows that a particular choice of M_k in a full SQP iteration yields a reduced step. This is an interesting interpretation of reduced SQP, and we want to demonstrate that the derivation in [14] is also suitable in an infinite-dimensional framework. It is easily verified that the operator S_k defined by

$$S_k = T_k^{-1}(I - R_k h'(x_k))$$

satisfies $\mathcal{N}(S_k) = \mathcal{R}(R_k)$. Furthermore, $S_k T_k = I_W$, and the following representation of the identity holds

$$R_k h'(x_k) + T_k S_k = I_X .$$

(Compare Gabay [13], Proposition 2.1, Gilbert [17], Proposition 3.1, for the finite-dimensional case.) Obviously, it is reasonable in an SQP method to keep the approximations to the Hessian positive definite at least on the (changing) null space of h'. This property can be achieved by a simple strategy if one takes into account that S_k is a left-inverse of T_k. Let $B_k \in \mathcal{L}(W)$ be self-adjoint and positive definite. Then the operator $M_k = S_k^* B_k S_k$ satisfies $T_k^* M_k T_k = B_k$ and, under the second order optimality condition for (1.1), locally B_k can be maintained positive definite for example by an appropriate reduced version of the BFGS update. Exactly this choice of M_k in (1.5) yields the reduced step (1.6), since $M_k R_k = 0$. In this way a reduced method can be interpreted as a special case of SQP.

We would now like to state some finite-dimensional convergence results for reduced SQP methods and for some of their modifications. If the choice of null space basis or right-inverse changes too much from one iterate to the next, superlinear convergence of these methods can be impeded. Therefore it is always assumed in the convergence analyses that $T_k = T(x_k)$ and $R_k = R(x_k)$ are given by functions R and T that are sufficiently smooth in some neighborhood of the solution. Since only reduced second order information is approximated, one cannot generally expect a reduced SQP method to be q-superlinearly convergent. (See the examples by Byrd [4] and Yuan [43].) However, under the assumption of convergence, a two-step q-superlinear convergence can be achieved, if the approximation of the reduced Hessian is consistent along the direction $B_k^{-1} T(x_k)^T \nabla f(x_k)$. More precisely, if the condition

$$\lim_{k \to \infty} \frac{\| [B_k - T(\hat{x})^T \nabla_x^2 L(\hat{x}, \hat{l}) T(\hat{x})] B_k^{-1} T(x_k)^T \nabla f(x_k) \|}{\| x_{k+1} - x_k \|} = 0 \tag{1.7}$$

holds, then the sequence generated from a reduced SQP method is two-step superlinearly convergent. That is,

$$\lim_{k \to \infty} \frac{\| x_{k+1} - \hat{x} \|}{\| x_{k-1} - \hat{x} \|} = 0 .$$

This result was given by Powell [37] for a particular method and Gabay [14] extended it to general reduced SQP methods. The conditions in [37] and [14] were originally formulated for the full SQP iteration, but the criterion (1.7) is a direct consequence if one considers reduced methods as a special case of SQP. In §3 we show that the infinite-dimensional version of (1.7) is sufficient for two-step superlinear convergence also in Hilbert space.

Several modifications of the reduced SQP iteration have been proposed in order to obtain (one-step) superlinear convergence, i.e., $\lim_{k \to \infty} \| x_{k+1} - \hat{x} \| / \| x_k - \hat{x} \| = 0$. From the analysis in [14] it follows that superlinear convergence could be established if $\lim_{k \to \infty} \| h(x_k) \| / \| x_k - \hat{x} \| = 0$. To achieve this condition Gabay [14] modifies the reduced SQP iteration to $x_{k+1} = x_k + p_k + s_k$ with p_k from (1.6) and $s_k = -R(x_k) h(x_k + p_k)$. The additional step s_k is only incorporated if it actually improves feasibility, where the improvement is measured by a simple criterion. He shows that (1.7) is sufficient for superlinear convergence of the sequence $\{x_k\}_{k=0}^{\infty}$. Furthermore, he introduces a globalization technique and proves a global convergence result. Similarly,

Byrd [5] considers the particular step $s_k = -h'(x_k)^T[h'(x_k)h'(x_k)^T]^{-1}h(x_k + p_k)$ which is added to p_k at each iteration. He shows that the sequence $\{x_k + p_k\}_{k=0}^{\infty}$ converges superlinearly provided the matrices B_k satisfy (1.7). We must note that the modifications in [14] and [5] are studied more generally in the framework of full SQP with the goal of obtaining superlinearly convergent sequences when only the reduced Hessian approximation is sufficiently accurate.

Gilbert [17] gives a modification where the reduced gradient $T(x_k - R(x_k)h(x_k))^T \nabla f(x_k - R(x_k)h(x_k))$ is calculated after the restoration. Unfortunately, this strategy requires two linearizations of the constraints per iteration: at x_k to determine $R(x_k)$ and at $x_k - R(x_k)h(x_k)$ to obtain $T(x_k - R(x_k)h(x_k))$. For this reason he suggests the following algorithm which avoids the linearization at x_k:

$$\begin{aligned} z_k &= x_k - R(z_{k-1})h(x_k), \\ x_{k+1} &= z_k - T(z_k)B_k^{-1}T(z_k)^T\nabla f(z_k). \end{aligned} \tag{1.8}$$

Assuming convergence of $\{x_k\}_{k=0}^{\infty}$ and $\{z_k\}_{k=0}^{\infty}$ a necessary and sufficient condition for superlinear convergence of the sequence $\{x_k\}_{k=0}^{\infty}$ is given. This characterization is similar to the sufficient condition (1.7). Again the examples by Byrd [4] and Yuan [43] show that the sequence $\{z_k\}_{k=0}^{\infty}$ does not usually converge superlinearly. Furthermore, Gilbert [17] adds a strategy to globalize the algorithm (1.8) and proves a global convergence result. In the global method the step is determined along an arc-shaped search path in order to decrease an exact penalty function with the help of an Armijo-like criterion. Conditions are given so that asymptotically the step-size is equal to one, i.e., the "Maratos effect" can be avoided. Gilbert [16] obtains similar convergence results for a variant of the method (1.8) with a supplementary restoration phase. The BFGS formula is used to update the approximation to the reduced Hessian but not at each iteration (a criterion is proposed when to skip the update). The use of general quasi-Newton type restoration steps in (1.8) is proposed by Gilbert [15] .

It was already mentioned that the class of reduced SQP methods allows some freedom in the definition of the right-inverse and the null space basis. This flexibility can be crucial in some applications such as, for example, in optimal control and parameter identification problems where T_k can be defined in a very natural way. Indeed, it is exactly this natural choice of T_k that makes reduced methods so effective and attractive for these problems. However, frequently in constrained optimization only a particular choice of matrices R_k and T_k is considered, namely, $R_k = h'(x_k)^T[h'(x_k)h'(x_k)^T]^{-1}$ the Moore–Penrose pseudo-inverse of $h'(x_k)$ and T_k is an orthonormal basis of $\mathcal{N}(h'(x_k))$ for the Euclidian scalar product. In this case $\mathcal{R}(R_k)$ is orthogonal to $\mathcal{N}(h'(x_k))$ and the restoration step may be regarded as a minimum norm Newton step on the equation $h(x) = 0$. Reduced SQP methods within this framework have been studied by Coleman and Conn [9], Nocedal and Overton [36], Byrd [5], Byrd and Nocedal [7], and Zhang and Zhu [44]. We summarize the corresponding convergence results.

Coleman and Conn [9] propose the following algorithm:

$$\begin{aligned} x_{k+1} &= x_k + t_k - h'(x_k)^T[h'(x_k)h'(x_k)^T]^{-1}h(x_k + t_k), \quad \text{where} \\ t_k &= -T(x_k)B_k^{-1}T(x_k)^T\nabla f(x_k). \end{aligned} \tag{1.9}$$

In contrast to the reduced step (1.6) the constraints in the restoration step in (1.9) are evaluated at $x_k + t_k$, after the tangent step t_k. In [9] the matrix approximation B_k in (1.9) is updated at each iteration by a projected version of the DFP or BFGS formula. Following the Broyden–Dennis–Moré convergence theory for unconstrained

optimization, Coleman and Conn show that their method is locally convergent and that the convergence is two-step superlinear. One cannot expect faster convergence for $\{x_k\}_{k=0}^{\infty}$. However, Byrd [5] shows that the sequence $\{x_k + t_k\}_{k=0}^{\infty}$ from (1.9) in the quasi- Newton version [9] converges superlinearly. This convergence behavior was already indicated in the context of algorithm (1.8). Indeed, it is easily checked that if $x_0 = z_0$ in (1.9) and provided (1.8) is started with $x_1 = z_0 - T(z_0)B_0^{-1}T(z_0)^T\nabla f(z_0)$, then the sequences $\{x_k\}_{k=0}^{\infty}$ and $\{x_k + t_k\}_{k=0}^{\infty}$ produced by (1.9) coincide with the sequences $\{z_k\}_{k=0}^{\infty}$ and $\{x_{k+1}\}_{k=0}^{\infty}$ generated from (1.8), respectively.

Byrd [5] also proves under certain conditions that if exact information is used in (1.9), i.e.,

$$B_k = T_k^T\nabla_x^2 L(x_k, l_k)T_k \quad \text{with} \quad l_k = [h'(x_k)h'(x_k)^T]^{-1}h'(x_k)\nabla f(x_k)\,,$$

then both sequences $\{x_k + t_k\}_{k=0}^{\infty}$ and $\{x_k\}_{k=0}^{\infty}$ converge to the solution with an r-order of at least $\frac{1}{2}(\sqrt{5}+1)$. We also note that Fontecilla [12] proposes a method that is related to the algorithm of Coleman and Conn [9] and has similar convergence properties. However, Fontecilla's method uses a full Hessian approximation.

The main difference between the algorithms by Coleman and Conn [9] and Nocedal and Overton [36] is the update rule. In [9] the new approximation B_{k+1} is defined so that $B_{k+1}s_k = v_k$, where $s_k = -B_k^{-1}T(x_k)^T\nabla f(x_k)$ and v_k is a secant approximation to $T(x_k)^T\nabla_x^2 L(x_k, l_k)T(x_k)s_k$. The particular definition of v_k in [9] ensures that $\|s_k\|$ is not small compared to $\|v_k\|$ which yields stability in the DFP and BFGS formula. However, the drawback of this choice of v_k is that the gradient of the Lagrangian must be computed twice at each step of (1.9). Nocedal and Overton [36] propose a more complex updating strategy that requires only one evaluation of the derivatives of f and h per iteration. The approximating matrix is updated by a standard formula such as DFP or BFGS if and only if a certain update criterion is satisfied. This criterion guarantees that the update is skipped if $\|s_k\|$ is considered to be too small. Various definitions for s_k and v_k are suggested for the update rule, all requiring only one set of gradient evaluations per iteration. It is shown in [36] that if the update criterion is defined appropriately and if the starting values are sufficiently accurate, then the iterates produced by the algorithm converge two-step superlinearly to the solution.

Byrd and Nocedal [7] develop some practical convergence results for the reduced SQP method using the updating procedure of [9] and a backtracking line search. The ℓ_1 merit function and a differentiable merit function proposed by Fletcher are considered for the line search. A global convergence result and a local r-linear convergence result are proved under certain assumptions. Moreover, conditions are given under which the consistency property (1.7) is satisfied. Then, due to the difficulties associated with the Maratos effect, the algorithm is modified for the ℓ_1 merit function. Two modifications are studied, and it is shown that they allow for unit steplengths near the solution which implies two-step superlinear convergence. This convergence result is also established for Fletcher's merit function. In contrast to the above mentioned analyses the convergence theory in [7] only assumes of the matrices B_k that the initial one is positive definite, and does not assume that the iterates converge. Zhang and Zhu [44] obtain a similar global convergence result. They employ the trust region approach to globalize the reduced SQP method. An algorithm is proposed and it is proved to be two-step superlinearly convergent under (1.7). However, it is not shown that the condition (1.7) holds for the method. Instead it is indicated that if the updating procedure from Nocedal and Overton [36] is used in the algorithm, then

the superlinear convergence of the sequence can be proved in a way that is similar to the proof in [36].

The assumption that T_k is given by a continuous function is somewhat problematic to ensure; see, e.g., Byrd and Schnabel [8]. An orthonormal basis T_k of $\mathcal{N}(h'(x_k))$ can be obtained by forming a QR factorization of $h'(x_k)^T$. Several procedures which are based on QR decomposition techniques and which guarantee that T varies smoothly have been suggested by Coleman and Sorensen [10] and Gill et al. [18]. However, the derivation of reduced SQP methods shows that there is no reason to calculate an orthonormal basis or to take a restoration step orthogonal to $\mathcal{N}(h'(x_k))$. Indeed, this orthogonal framework is unsuitable if the null space has a special structure that allows for a natural definition of a right-inverse and of a null space basis. Such a structure always occurs when the variables can be separated as for example in optimal control problems or parameter identification problems. Consider for awhile optimal control problems where one usually distinguishes between the control variables $u \in U$ and the state variables $y \in Y$. Then the iterates $x_k = (y_k, u_k)$ belong to a product space $X = Y \times U$ and $(\xi, \nu) \in X$ lies in $\mathcal{N}(h'(y_k, u_k))$ if and only if

$$h'_y(y_k, u_k)\xi = -h'_u(y_k, u_k)\nu\,,$$

where $h'_y(y_k, u_k) \in \mathcal{L}(Y, Z)$ and $h'_u(y_k, u_k) \in \mathcal{L}(U, Z)$ are the partial Fréchet-derivatives of h in (y_k, u_k). Provided that $h'_y(y_k, u_k)$ is bijective, the elements of the null space can be represented as follows:

$$(\xi, \nu) \in \mathcal{N}(h'(y_k, u_k)) \quad \Leftrightarrow \quad (\xi, \nu) = (-h'_y(y_k, u_k)^{-1} h'_u(y_k, u_k)\nu, \nu)\,.$$

This suggests the following natural choice which defines what could be called the separability framework:

$$\begin{aligned} W &= U, \\ T(y_k, u_k) &= (-h'_y(y_k, u_k)^{-1} h'_u(y_k, u_k),\ I_U)\,, \quad \text{and} \\ R(y_k, u_k) &= (h'_y(y_k, u_k)^{-1},\ \mathcal{O})\,, \end{aligned}$$

where the second component in $R(y_k, u_k)$ denotes the null operator in $\mathcal{L}(Z, U)$. This choice is also adopted by Kunisch and Sachs [30], where reduced SQP is applied to parameter identification problems, and the practical relevance of the separability framework is already mentioned by Gilbert [17]. We also note that Hoyer [24] applies this framework to develop quadratically convergent variants of reduced methods using exact second order information. He considers finite-dimensional problems with separable variables which, by the implicit function theorem, are transformed into unconstrained minimization problems of reduced dimension. If Newton's method is applied to the reduced problem, at each iteration a nonlinear subproblem must be solved. The algorithms in [24] are derived by replacing these subproblems by a suitable linearization. This yields an interesting interpretation of reduced SQP methods that use the separability framework.

The example of separability shows that one should not concentrate only on the Penrose pseudo-inverse and on a parametrization of the null space through an orthonormal basis, but that it is sensible to allow for an arbitrary choice of R and T in a reduced SQP method. Furthermore, in the separability approach the advantage of reduction becomes quite obvious. At each iteration of a full SQP method we must store and update an operator $M_k \in \mathcal{L}(Y \times U)$. In contrast, in a reduced method the approximating operators belong to $\mathcal{L}(U)$. Let $Y_N \subset Y$ and $U_M \subset U$ be finite-dimensional discretizations of the spaces Y and U, respectively. If the dimension of Y_N

is considerably higher than that of U_M, the savings in storage can be enormous when a reduced SQP method is used. This situation occurs for example in certain parabolic boundary control problems (see Kupfer [31], Kupfer and Sachs [32, 33]), where the control space $U = U([0,T])$ is a function space over an interval while the state space $Y = Y([0,T] \times \Omega)$ is a function space over $[0,T] \times \Omega$ for a domain $\Omega \subset \mathbb{R}^n$. If one discretizes the time-dependent space U by a d_t-dimensional space and the space- and time-dependent Y by a $d_t n d_x$-dimensional space, where d_x denotes the discretization in one space component, then the matrix to be stored for a full SQP update is of the size $\mathbb{R}^{d_t(1+nd_x) \times d_t(1+nd_x)}$ as opposed to the size $\mathbb{R}^{d_t \times d_t}$ in a reduced update. This shows that a full method is not to be recommended in this case. It is the main advantage of the reduced SQP method that it allows for the use of the structure of the null space.

In §3 we establish a convergence theory for general reduced SQP methods in Hilbert space. Indeed, the discussion of the preceding paragraph shows that there are practical problems, which are usually formulated in infinite-dimensional spaces, where the application of a reduced method is not only suitable, but also offers enormous advantage over methods that approximate full Hessian information. Moreover, for an efficient treatment of these problems by a reduced algorithm it is absolutely necessary to allow for a general framework that is not restricted by the Penrose pseudo-inverse or a particular null space parametrization. On the other hand it is well known that the speed of convergence of iterative methods applied to a discretized problem may depend strongly on their rate of convergence for the underlying infinite-dimensional problem, and, although the algorithm in infinite- dimensional spaces is only conceptual and not implementable, it can serve as a tool to predict the convergence behavior of finite-dimensional implementations. It is for these reasons that we study the convergence behavior of infinite-dimensional reduced SQP algorithms in the general setting (1.6), and we think that this convergence theory may give some insight into the numerical performance of reduced SQP in infinite-dimensional applications studied recently [30, 31, 33].

The convergence analysis is organized in the following manner. We first establish some properties of the iteration function corresponding to the reduced SQP method. These properties are basic both for the derivation of a sufficient condition for two-step superlinear convergence and for the proof of a two-step contraction property of the iterates. The technique of proof based on properties of the iteration function is somewhat unusual. However, this approach has turned out to be rather convenient if the convergence behavior of modifications of the standard method is analyzed. This will not be discussed any further in this paper, and the interested reader is referred to Kupfer [31]. In the second part of the convergence analysis we consider the updates of the convex Broyden class, $B_{k+1} = B_k + V(B_k, s_k, v_k, \alpha_k)$, and we give a condition for v_k and s_k, which ensures that the corresponding reduced secant method converges to the solution two-step q-linearly. Finally, we will show that the same condition and an additional compactness assumption yield two-step superlinear convergence. The main tool in our proof of two-step superlinear convergence will be a general result for the updates of the convex Broyden class. To our knowledge the formulation of this result is new in Hilbert space. However, it follows almost directly from Griewank's [20] convergence analysis, and it can be regarded as an extension of a result given by Byrd and Nocedal [6] for the finite-dimensional BFGS-update. To be more precise, we will give a condition for v_k and s_k which ensures that

$$\lim_{k\to\infty} \frac{\|(B_k - \hat{D})s_k\|}{\|s_k\|} = 0$$

holds, provided that the sequences $\{\|B_k\|\}_{k=0}^{\infty}$ and $\{\|B_k^{-1}\|\}_{k=0}^{\infty}$ are bounded and that $B_0 - \hat{D}$ is compact. Here $\hat{D}$ is a self-adjoint and positive definite operator. This statement is independent of the algorithmic context of the update and relevant to both constrained and unconstrained optimization. We note that this result can be applied to prove superlinear convergence of general quasi-Newton methods without using Frobenius norm estimates. This can be of interest for infinite- dimensional convergence analyses.

In the next section we give definitions and state the assumptions that will be used to prove the convergence results. One of the assumptions to be checked is that the operators $R(x)$ and $T(x)$ are continuously Fréchet-differentiable (with respect to x) in some neighborhood of the solution. In our application to optimal control problems this assumption can be guaranteed, if standard smoothness properties hold for the constraint function h. This is proved in §4 where the separability approach is used to formulate an appropriate reduced SQP algorithm for certain optimal control problems. The main goal of §4 is to verify all the assumptions needed in the convergence analysis by making use of known results from optimal control theory.

2. Definitions and assumptions. The purpose of this section is to give definitions and assumptions that will be used to prove the convergence results.

For the following notation and definitions let $E = (E, \|\cdot\|_E)$ and $F = (F, \|\cdot\|_F)$ be normed linear spaces, and let $H = (H, \|\cdot\|_H)$ be a Hilbert space where the norm in H is induced by the scalar product $\langle\cdot,\cdot\rangle_H$. We always drop the subscript in the norm and in the scalar product if its use is clear from the context. The dual space of E is denoted by E^*, and Hilbert spaces will be identified with their duals, i.e., we let $H = H^*$. The space of linear and bounded operators from E into F (with domain E) endowed with the usual operator norm is denoted by $\mathcal{L}(E,F)$. For simplicity, $\mathcal{L}(E,E)$ will be abbreviated to $\mathcal{L}(E)$. The set $\mathcal{COM}(H)$ denotes the space of linear and compact operators in H. For $A \in \mathcal{L}(E,F)$ we use the symbols $\mathcal{R}(A)$ and $\mathcal{N}(A)$ for the range space and the null space of A, respectively, and the operator $A^* \in \mathcal{L}(F^*,E^*)$ denotes the adjoint of A. We call an operator $Q \in \mathcal{L}(H)$ positive if $\langle h, Qh\rangle \geq 0$ for all $h \in H$, and we say that Q is positive definite if there is a constant $c > 0$ such that

$$\langle h, Qh\rangle \geq c\|h\|^2 \quad \text{for all } h \in H.$$

As usual ∇ will denote the gradient operator. In this section and in §3 all derivatives will be Fréchet-derivatives, unless stated otherwise.

The Lagrangian function will be defined as in (1.2). Recall that $\hat{x} \in X$ is a local solution of problem (1.1) with associated multiplier $\hat{l}$. Let $D := \{x \in X : \|x - \hat{x}\| < \rho\}$ be a ball with radius ρ around $\hat{x}$. A mapping $\varphi : D \to F$ is said to be Lipschitz continuous at $\hat{x}$ in D, written $\varphi \in Lip(D)$, if there is a constant $M > 0$ such that

$$\|\varphi(x) - \varphi(\hat{x})\| \leq M\|x - \hat{x}\| \quad \text{for all } x \in D.$$

M is called Lipschitz constant of φ.

To define a reduced SQP method in D we suppose that there exist a real Hilbert space $W \neq \{0\}$ and mappings $T : D \to \mathcal{L}(W,X)$ and $R : D \to \mathcal{L}(Z,X)$ such that

$$\mathcal{N}(h'(x)) = \mathcal{R}(T(x)) \quad \text{and} \quad h'(x)R(x) = I_Z \quad \text{for all } x \in D.$$

We note that the existence of a right-inverse $R(x)$ implies that $h'(x)$ is surjective. Throughout §3 we suppose that $T(x)$ is injective for $x \in D$, and the following hypotheses will be assumed:

(A1) f and h are twice continuously differentiable in D and $f'' \in \mathrm{Lip}(D)$, $h'' \in \mathrm{Lip}(D)$.

(A2) T and R are continuously differentiable in D and $T' \in \mathrm{Lip}(D), R' \in \mathrm{Lip}(D)$.

(A3) There exists $m > 0$ such that

$$\langle s, L_x''(\hat{x}, \hat{l})s\rangle \geq m\|s\|^2 \quad \text{for all } s \in \mathcal{N}(h'(\hat{x})).$$

Note that, since T is a mapping from D into $\mathcal{L}(W, X)$, it is required in (A2) that

$$\|T'(x) - T'(\hat{x})\|_{\mathcal{L}(X,\mathcal{L}(W,X))} \leq M\|x - \hat{x}\| \quad \text{for all } x \in D,$$

and an analogous estimate must hold for R'. The assumption (A3) (which is equivalent to the positive definiteness of $T(\hat{x})^* L_x''(\hat{x}, \hat{l})T(\hat{x})$) is the well-known second order sufficiency condition from constrained optimization.

The assumptions (A1) and (A3) are standard in the finite-dimensional convergence theory of reduced SQP methods, and also the smoothness condition (A2) for the mappings T and R is usually assumed when reduced SQP is studied in the general framework, compare Gabay [14] and Gilbert [17]. For a particular implementation (A2) can be weakened to the Lipschitz continuity of T and R on D; see, for example, the algorithms by Coleman and Conn [9] and Nocedal and Overton [36]. In §4 we will see that (A2) can be satisfied if the minimization problem has a special structure.

Since $T(x)^{-1} \in \mathcal{L}(\mathcal{N}(h'(x)), W)$, we can define the following operator for $x \in D$

$$S(x) = T(x)^{-1}(I_X - R(x)h'(x)).$$

Obviously, $S(x) \in \mathcal{L}(X, W)$ satisfies

$$\mathcal{N}(S(x)) = \mathcal{R}(R(x)) \quad \text{and} \quad S(x)T(x) = I_W \quad \text{and} \quad T(x)S(x) + R(x)h'(x) = I_X.$$

Moreover, the Lipschitz continuity of T, h', and R on D implies that S is Lipschitz continuous on a possibly smaller set $\bar{D} \subset D$. Hence we can assume that the radius of D is sufficiently small so that $S \in \mathrm{Lip}(D)$.

In the first part of §3 we establish convergence results that hold for the reduced SQP method under certain assumptions on the sequence $\{B_k\}_{k=0}^{\infty}$. We do not consider a specific update, and we use the following algorithmic formulation for one iteration of the method.

ALGORITHM 2.1 (Reduced SQP Method)

Given $x_k \in D$ and $B_k \in \mathcal{L}(W), B_k$ bijective. Let

$$x_{k+1} = x_k - T(x_k)B_k^{-1}T(x_k)^*\nabla f(x_k) - R(x_k)h(x_k).$$

Obtain B_{k+1} from some update rule.

The superlinear convergence results will be proved for reduced secant methods where the sequence $\{B_k\}_{k=0}^{\infty}$ in Algorithm 2.1 is generated by any update of the convex Broyden family:

$$
\begin{aligned}
&B_{k+1} = B_k + V(B_k, s_k, v_k, \alpha_k), \qquad \alpha_k \in [0,1], \\
&\text{where} \quad V(B,s,v,\alpha) = \frac{v \otimes v}{\langle v, s\rangle} - \frac{(Bs)\otimes(Bs)}{\langle s, Bs\rangle} + \alpha\langle s, Bs\rangle u \otimes u \qquad (2.1) \\
&\text{with} \qquad u = \frac{v}{\langle v, s\rangle} - \frac{Bs}{\langle s, Bs\rangle}.
\end{aligned}
$$

Here, for $w_1, w_2 \in W$ the operator $w_1 \otimes w_2 \in \mathcal{L}(W)$ is defined as

$$(w_1 \otimes w_2)(w) = \langle w_2, w\rangle w_1 \quad \text{for } w \in W.$$

For the sake of completeness we present one step of the reduced secant method in its entirety instead of merely adding an appropriate update strategy to Algorithm 2.1.

ALGORITHM 2.2 (Reduced Secant Method)

Given $x_k \in D$ and $B_k \in \mathcal{L}(W)$, B_k bijective.

(1) Solve $B_k w_k = -T(x_k)^* \nabla f(x_k)$ for $w_k \in W$.

(2) Set $x_{k+1} = x_k + T(x_k)w_k - R(x_k)h(x_k)$.

(3) Select $s_k \in W, v_k \in W, \alpha_k \in [0,1]$.

(4) Set $B_{k+1} = B_k + V(B_k, s_k, v_k, \alpha_k)$, if it is defined.
Else set $B_{k+1} = B_k$.

The first question to be addressed concerns the definitions that can be used for v_k and s_k. Possible candidates are proposed by Nocedal and Overton [36] for the orthogonal framework and similar choices are conceivable for reduced methods in their general formulation. However, in infinite-dimensional applications it may happen that v_k or s_k cannot be calculated explicitly from a given formula, but an approximation must be used instead. We allow for these perturbations by not considering particular definitions for v_k and s_k. It will be shown instead that a two-step superlinear convergence can be achieved by a reduced secant method if a certain condition is satisfied by the sequences $\{v_k\}_{k=0}^{\infty}$ and $\{s_k\}_{k=0}^{\infty}$. In particular, we will show that this condition is fulfilled if the standard choice $s_k = -B_k^{-1}T(x_k)^*\nabla f(x_k)$ is used, together with one of the following definitions for v_k:

$$
\begin{aligned}
v_k &= T(x_k)^*[\nabla_x L(x_k + T(x_k)w_k, l_k) - \nabla f(x_k)], \\
v_k &= T(x_k + T(x_k)w_k)^* \nabla f(x_k + T(x_k)w_k) - T(x_k)^* \nabla f(x_k).
\end{aligned}
$$

Here l_k is some approximation to the Lagrange multiplier. The first formula for v_k is that proposed and analyzed by Coleman and Conn [9].

Some additional notation is needed. We use p_k to represent the step $x_{k+1} - x_k$ and $e_k = x_k - \hat{x}$ to denote the current error. For convenience, we will assume that finite convergence does not occur, i.e., $x_k \neq \hat{x}$ for all k. We use the standard notation

$$\sigma(x,y) := \max\{\|x - \hat{x}\|, \|y - \hat{x}\|\} \quad \text{for } x, y \in X,$$

and for $\gamma > 0$ we define the sets

$$\mathcal{B}(\gamma) := \{Q \in \mathcal{L}(W)\,;\, Q \text{ bijective and } \|Q\| \le \gamma\,,\ \|Q^{-1}\| \le \gamma\}.$$

Finally, we let $\hat{G} = L''_x(\hat{x}, \hat{l})$ and $\hat{H} = T(\hat{x})^* \hat{G} T(\hat{x})$.

3. Local convergence analysis. This section is devoted to the study of the convergence of the sequences generated by the reduced SQP algorithms. As a first step we show that the infinite-dimensional analog of (1.7) ensures two-step superlinear convergence of the iterates produced by Algorithm 2.1. The results and estimates established for the proof of this sufficient criterion will be used further to show a contraction property for the iterates that ensures two-step linear convergence, provided $\|B_k - \hat{H}\|$ is sufficiently small for all k.

One iteration of Algorithm 2.1 can be written as follows:

$$x_{k+1} = x_k + \Psi(x_k, B_k) = \Phi_k(x_k).$$

Here we introduced the iteration function

$$\Phi_k(x) := x + \Psi(x, B_k), \quad x \in D, k \in \mathbb{N}_0,$$

where Ψ is defined for $x \in D$ and for $B \in \mathcal{L}(W), B$ bijective, as

$$\Psi(x, B) := -T(x) B^{-1} T(x)^* \nabla f(x) - R(x) h(x).$$

Stoer and Tapia [39] use a similar setting to study stability properties of full SQP methods in $\mathbb{R}^n$. The differentiability properties of the iteration function are explored in [39] and, as a by-product, a short proof is obtained for Powell's [37] sufficient condition for two-step superlinear convergence of an SQP method. This proof makes use of an orthonormal basis $T_k = T(x_k)$ of $\mathcal{N}(h'(x_k))$, where $T(x)$ is a continuously differentiable matrix function obtained from a particular QR decomposition of $h'(x)^T$. Under the smoothness assumption (A2) we can use a similar approach as in [39] to prove a convergence result for reduced SQP methods in Hilbert space without restricting ourselves to a particular choice of T.

In this section we sometimes drop the subscripts k, and we denote the current iterates by x_c, B_c and the preceding and following iterates by x_-, B_- and x_+, B_+, respectively. With this notation two consecutive steps of Algorithm 2.1 can be expressed as follows:

$$\begin{aligned} x_c &= x_- + \Psi(x_-, B_-), \\ x_+ &= x_c + \Psi(x_c, B_c). \end{aligned} \tag{3.1}$$

Furthermore, it is convenient to define the iteration function more generally as

$$\Phi(x, B) := x + \Psi(x, B) \quad \text{for } x \in D \text{ and } B \in \mathcal{L}(W), B \text{ bijective.}$$

We use $\Psi'(\cdot, B)$ and $\Phi'(\cdot, B)$ to denote the derivative of $\Psi(\cdot, B)$ and $\Phi(\cdot, B)$, respectively.

The Kuhn–Tucker conditions for problem (1.1) imply that $\hat{x}$ is a fixed point of $\Phi(\cdot, B)$ for all $B \in \mathcal{L}(W), B$ bijective. We consider the error in iteration $k+1$ of Algorithm 2.1:

$$e_{k+1} = \Phi_k(x_k) - \Phi_k(\hat{x}).$$

To obtain an estimate for the ratio $\|e_{k+1}\|/\|e_{k-1}\|$ we show that Φ_k is differentiable in D. Then we can write

$$[I - \Phi_k'(\hat{x})]e_{k+1} = \Phi_k(x_k) - \Phi_k(\hat{x}) - \Phi_k'(\hat{x})e_k - \Phi_k'(\hat{x})p_k \tag{3.2}$$

and the mean value theorem can be applied to Φ_k. Furthermore, an explicit formula of $\Psi'(\hat{x}, B_k)$ will be helpful, since it can be used to analyze the invertibility properties of the operator on the left side of (3.2). For this purpose we calculate the derivative of the reduced gradient in the proof of Lemma 3.1 below. Evaluation of this derivative at $\hat{x}$ yields a formula for $\Psi'(\hat{x}, B)$ which is independent of $T'(\hat{x})$.

LEMMA 3.1. *Assume* (A1) *and* (A2), *and let* $B \in \mathcal{L}(W)$, B *bijective. Then* $\Psi(\cdot, B)$ *is continuously differentiable in* D *and* $\Psi'(\cdot, B) \in \mathrm{Lip}(D)$. *In particular,*

$$\Psi'(\hat{x}, B) = -T(\hat{x})B^{-1}T(\hat{x})^*\hat{G} - R(\hat{x})h'(\hat{x}). \tag{3.3}$$

Proof. For $x \in D$ let $g(x) = T(x)^*\nabla f(x)$. The differentiability of g and Ψ follows from (A1), (A2), and the definition of the Fréchet-derivative. Furthermore,

$$\begin{aligned}\Psi'(x, B)\xi = &-(T'(x)\xi)B^{-1}g(x) - T(x)B^{-1}g'(x)\xi \\ &-(R'(x)\xi)h(x) - R(x)h'(x)\xi, \qquad x \in D, \xi \in X.\end{aligned} \tag{3.4}$$

This shows that $\Psi'(\cdot, B)$ is continuous in D and $\Psi'(\cdot, B) \in \mathrm{Lip}(D)$.

We calculate the derivative of g. The surjectivity of $h'(x)$ implies that

$$[h'(x)h'(x)^*]^{-1} \in \mathcal{L}(Z, Z^*) \quad \text{for } x \in D.$$

Hence, for $x \in D$ we can define

$$\begin{aligned}P(x) &= I - h'(x)^*[h'(x)h'(x)^*]^{-1}h'(x) \quad \text{and} \\ l(x) &= [h'(x)h'(x)^*]^{-1}h'(x)\nabla f(x).\end{aligned} \tag{3.5}$$

Note that $P(x)d$ is the orthogonal projection of d onto $\mathcal{N}(h'(x))$ for $d \in X$, and $l(x) \in Z^*$ is often called the least squares multiplier estimate.

Consider the mapping $\varphi : D \to \mathcal{L}(Z^*, W)$ defined as

$$\varphi(x) = T(x)^*h'(x)^*, \quad x \in D.$$

Since $\varphi(x)$ is the null operator for $x \in D$, we obtain from the definition of the Fréchet-derivative

$$0 = \varphi'(x)\xi = (T'(x)\xi)^*h'(x)^* + T(x)^*(h''(x)\xi)^*, \quad \xi \in X,\ x \in D.$$

Consequently,

$$(T'(x)\xi)^*h'(x)^* = -T(x)^*(h''(x)\xi)^*, \quad \xi \in X,\ x \in D.$$

Together with (3.5) this shows

$$\begin{aligned}(T'(x)\xi)^*\nabla f(x) &= (T'(x)\xi)^*[P(x)\nabla f(x) + h'(x)^*[h'(x)h'(x)^*]^{-1}h'(x)\nabla f(x)] \\ &= (T'(x)\xi)^*P(x)\nabla f(x) - T(x)^*(h''(x)\xi)^*l(x), \quad \xi \in X,\ x \in D.\end{aligned}$$

This identity and the formula

$$L_x''(x, l)\xi = f''(x)\xi - (h''(x)\xi)^*l, \quad \xi \in X,\ x \in D,\ l \in Z^*$$

lead to the following expression for the derivative of the reduced gradient

$$\begin{aligned} g'(x)\xi &= (T'(x)\xi)^*\nabla f(x) + T(x)^* f''(x)\xi \\ &= (T'(x)\xi)^* P(x)\nabla f(x) + T(x)^* L_x''(x, l(x))\xi, \quad \xi \in X,\ x \in D. \end{aligned} \tag{3.6}$$

Recall from the optimality conditions for (1.1) that $\hat{l} = l(\hat{x})$ and $P(\hat{x})\nabla f(\hat{x}) = 0$. This implies $g'(\hat{x}) = T(\hat{x})^*\hat{G}$. Now the formula (3.3) for $\Psi'(\hat{x}, B)$ follows from (3.4), since $g(\hat{x}) = 0$ and $h(\hat{x}) = 0$. □

Remark. The formula for the Fréchet-differential (3.4) shows that the Lipschitz constant of $\Psi'(\cdot, B)$ depends on $\|B^{-1}\|$.

Remark. Results on the derivative of $T(x)^T\nabla f(x)$ that are similar to (3.6) have been given by Goodman [19] and Nocedal and Overton [36].

The next lemma will be applied to obtain a representation of the error e_{k+1} by eliminating the operator $I - \Phi_k'(\hat{x})$ on the left side of (3.2).

LEMMA 3.2. *Let* (A1) *and* (A2) *hold, and assume that* $\hat{H}^{-1} \in \mathcal{L}(W)$. *Let* $B \in \mathcal{L}(W)$ *be a bijective operator. Then* $\Psi'(\hat{x}, B)$ *is bijective, and the inverse operator is given by*

$$\Psi'(\hat{x}, B)^{-1} = T(\hat{x})\hat{H}^{-1}[T(\hat{x})^*\hat{G} - BS(\hat{x})] - I_X. \tag{3.7}$$

Proof. Set $\Gamma(\hat{x}, B) = T(\hat{x})\hat{H}^{-1}[T(\hat{x})^*\hat{G} - BS(\hat{x})] - I$.

Recall that $S(\hat{x})T(\hat{x}) = I_W, S(\hat{x})R(\hat{x}) = 0$, and, by definition, $\hat{H} = T(\hat{x})^*\hat{G}T(\hat{x})$. Then it follows directly from (3.3) that

$$\begin{aligned} \Gamma(\hat{x}, B)\Psi'(\hat{x}, B) &= -T(\hat{x})\hat{H}^{-1}T(\hat{x})^*\hat{G}T(\hat{x})B^{-1}T(\hat{x})^*\hat{G} - T(\hat{x})\hat{H}^{-1}T(\hat{x})^*\hat{G}R(\hat{x})h'(\hat{x}) \\ &\quad + T(\hat{x})\hat{H}^{-1}T(\hat{x})^*\hat{G} + T(\hat{x})B^{-1}T(\hat{x})^*\hat{G} + R(\hat{x})h'(\hat{x}) \\ &= T(\hat{x})\hat{H}^{-1}T(\hat{x})^*\hat{G}[I - R(\hat{x})h'(\hat{x})] + R(\hat{x})h'(\hat{x}) \\ &= T(\hat{x})S(\hat{x}) + R(\hat{x})h'(\hat{x}) = I. \end{aligned}$$

A similar calculation using that $h'(\hat{x})T(\hat{x}) = 0$ shows that

$$\Psi'(\hat{x}, B)\Gamma(\hat{x}, B) = T(\hat{x})S(\hat{x}) + R(\hat{x})h'(\hat{x}) = I.$$

This implies that $\Psi'(\hat{x}, B)$ is bijective, and $\Gamma(\hat{x}, B) \in \mathcal{L}(X)$ is the inverse of $\Psi'(\hat{x}, B)$. □

Now, (3.2) is justified by Lemma 3.1. Moreover, the existence of the inverse operators and the definition of Φ_k imply that

$$\Psi'(\hat{x}, B_k)^{-1}\Phi_k'(\hat{x}) = \Psi'(\hat{x}, B_k)^{-1}(I + \Psi'(\hat{x}, B_k)) = \Psi'(\hat{x}, B_k)^{-1} + I,$$

so that with $[I - \Phi_k'(\hat{x})]^{-1} = -\Psi'(\hat{x}, B_k)^{-1}$ the identity (3.2) can be rewritten as

$$e_{k+1} = -\Psi'(\hat{x}, B_k)^{-1}[\Phi_k(x_k) - \Phi_k(\hat{x}) - \Phi_k'(\hat{x})e_k] + (I + \Psi'(\hat{x}, B_k)^{-1})p_k. \tag{3.8}$$

The mean value theorem (see, e.g., [21, p. 255]) applied to the term in square brackets in (3.8) yields a bound that depends on the iteration index k. But, if the sequence $\{\|B_k^{-1}\|\}_{k=0}^{\infty}$ is bounded, then the Lipschitz constants of Φ_k' can be chosen uniformly in k. More precisely, let the assumptions of Lemma 3.2 hold. For $\gamma > 0$ we let $M_\gamma > 1$ be a constant such that for all $B \in \mathcal{B}(\gamma)$ the following estimates hold:

$$\left.\begin{aligned} \|\Psi(x, B) - \Psi(\hat{x}, B)\| &\le M_\gamma\|x - \hat{x}\| \\ \|\Phi(x, B) - \Phi(\hat{x}, B)\| &\le M_\gamma\|x - \hat{x}\| \\ \|\Psi'(x, B) - \Psi'(\hat{x}, B)\| &\le M_\gamma\|x - \hat{x}\| \end{aligned}\right\} \quad \text{for all } x \in D, \tag{3.9}$$

$$\|\Phi'(\hat{x}, B)\| \leq M_\gamma, \ \|\Psi'(\hat{x}, B)\| \leq M_\gamma, \ \|\Psi'(\hat{x}, B)^{-1}\| \leq M_\gamma.$$

One can see from (3.3), (3.4), and (3.7) that such a choice of M_γ is certainly possible, where M_γ only depends on γ and on the radius of D, further on quantities evaluated at $\hat{x}$, and on the Lipschitz constants of f, h, T, R, including their derivatives.

To analyze the convergence behavior of Algorithm 2.1 we examine the second part of the error in (3.8) given by $(I + \Psi'(\hat{x}, B_k)^{-1})p_k$. This can be done conveniently because the inverse of $\Psi'(\hat{x}, B_k)$ is known explicitly. From (3.7) we get the following rather lengthy expression:

$$\begin{aligned} I + \Psi'(\hat{x}, B_k)^{-1} &= T(\hat{x})\hat{H}^{-1}(T(\hat{x})^*\hat{G}[T(\hat{x})S(\hat{x}) + R(\hat{x})h'(\hat{x})] - B_k S(\hat{x})) \\ &= -T(\hat{x})\hat{H}^{-1}([B_k - \hat{H}]S(x_k) + [B_k - \hat{H}][S(\hat{x}) - S(x_k)] \\ &\qquad -T(\hat{x})^*\hat{G}R(\hat{x})h'(\hat{x})). \end{aligned} \tag{3.10}$$

This identity leads us to the following important observation: If it is possible to show for Algorithm 2.1 that

$$\|R(\hat{x})h'(\hat{x})p_k\| \leq \eta_k \|e_k\|$$

for some sequence $\{\eta_k\}_{k=0}^\infty$ satisfying $\lim_{k\to\infty} \eta_k = 0$, then the convergence of $\{x_k\}_{k=0}^\infty$ to $\hat{x}$ and the condition

$$\lim_{k\to\infty} \frac{\|(B_k - \hat{H})S(x_k)p_k\|}{\|p_k\|} = 0$$

would imply (one-step) superlinear convergence.

Unfortunately, generally one can only achieve a weaker estimate that can be regarded as the source for the two-step convergence properties of reduced SQP methods. For x_c and x_+ given by (3.1) we prove the following result.

LEMMA 3.3. *Suppose that $\hat{H}$ is bijective, and let* (A1), (A2) *hold. Then, for each $\gamma > 0$ there exists a positive constant $c = c(\gamma)$ such that for all $x_- \in D$, $B_- \in \mathcal{B}(\gamma)$ with $x_c = \Phi(x_-, B_-) \in D$ and $B_c \in \mathcal{B}(\gamma)$ the estimate*

$$\|R(\hat{x})h'(\hat{x})(x_+ - x_c)\| \leq c\|x_- - \hat{x}\|^2 \tag{3.11}$$

is satisfied.

Proof. For $\gamma > 0$ and M_γ from (3.9) choose

$$c > \frac{M_\gamma^2(1 + M_\gamma)}{2}\|R(\hat{x})h'(\hat{x})\|. \tag{3.12}$$

Let $x_- \in D$ and $B_-, B_c \in \mathcal{B}(\gamma)$ such that x_c is in D. Then x_+ is well defined and by (3.9)

$$\|x_c - \hat{x}\| = \|\Phi(x_-, B_-) - \Phi(\hat{x}, B_-)\| \leq M_\gamma\|x_- - \hat{x}\|. \tag{3.13}$$

From (3.3) and the definitions of $R(\hat{x})$ and $T(\hat{x})$ one obtains

$$h'(\hat{x})\Psi'(\hat{x}, B_c) = -h'(\hat{x}) \quad \text{and} \quad h'(\hat{x})\Phi'(\hat{x}, B_-) = 0,$$

and it follows that

$$R(\hat{x})h'(\hat{x})\Psi'(\hat{x}, B_c)\Phi'(\hat{x}, B_-) = -R(\hat{x})h'(\hat{x})\Phi'(\hat{x}, B_-) = 0. \tag{3.14}$$

With $x_c - \hat{x} = \Phi(x_-, B_-) - \Phi(\hat{x}, B_-)$ the step can be written as

$$\begin{aligned} x_+ - x_c &= \Phi(x_c, B_c) - \hat{x} - (x_c - \hat{x}) \\ &= \Phi(x_c, B_c) - \hat{x} - (I + \Psi'(\hat{x}, B_c))(x_c - \hat{x}) + \Psi'(\hat{x}, B_c)(x_c - \hat{x}) \\ &= \Phi(x_c, B_c) - \Phi(\hat{x}, B_c) - \Phi'(\hat{x}, B_c)(x_c - \hat{x}) \\ &\quad + \Psi'(\hat{x}, B_c)(\Phi(x_-, B_-) - \Phi(\hat{x}, B_-)). \end{aligned}$$

If we multiply both sides of this equation by $R(\hat{x})h'(\hat{x})$ and use the representation (3.14) of the null operator, then we have

$$\begin{aligned} R(\hat{x})h'(\hat{x})(x_+ - x_c) &= R(\hat{x})h'(\hat{x})[\Phi(x_c, B_c) - \Phi(\hat{x}, B_c) - \Phi'(\hat{x}, B_c)(x_c - \hat{x})] \\ &\quad + R(\hat{x})h'(\hat{x})\Psi'(\hat{x}, B_c)[\Phi(x_-, B_-) - \Phi(\hat{x}, B_-) - \Phi'(\hat{x}, B_-)(x_- - \hat{x})]. \end{aligned}$$

Taking norms, using the mean value theorem and (3.9), (3.13), one finally gets

$$\begin{aligned} \|R(\hat{x})h'(\hat{x})(x_+ - x_c)\| &\le \tfrac{M_\gamma}{2}\|R(\hat{x})h'(\hat{x})\|(\|x_c - \hat{x}\|^2 + \|\Psi'(\hat{x}, B_c)\|\|x_- - \hat{x}\|^2) \\ &\le c\|x_- - \hat{x}\|^2 \end{aligned}$$

with $c > 0$ chosen from (3.12). □

Now we are in the position to give a sufficient condition for two-step superlinear convergence of reduced SQP methods in Hilbert space. For the finite-dimensional case this result is a consequence of a sufficient condition established for full SQP methods by Powell [37] and Gabay [14].

THEOREM 3.4. *Assume* (A1), (A2), *and let* $\hat{H}$ *be bijective. Suppose further that for some* $\gamma > 0$

$$\|B_k\| \le \gamma \quad \textit{and} \quad \|B_k^{-1}\| \le \gamma$$

for all $k \in \mathbb{N}_0$. *If the sequence* $\{x_k\}_{k=0}^{\infty}$ *generated by Algorithm* 2.1 *converges to* $\hat{x}$ *and*

$$\lim_{k\to\infty} \frac{\|(B_k - \hat{H})B_k^{-1}T(x_k)^*\nabla f(x_k)\|}{\|x_{k+1} - x_k\|} = 0, \tag{3.15}$$

then

$$\lim_{k\to\infty} \frac{\|x_{k+1} - \hat{x}\|}{\|x_{k-1} - \hat{x}\|} = 0.$$

Proof. Since $x_k \to \hat{x}$ and $x_k \neq \hat{x}$, we can assume that $p_k \neq 0$ and $x_k \in D$ for all k. Due to the boundedness of $\{\|B_k\|\}_{k=0}^{\infty}$ and $\{\|B_k^{-1}\|\}_{k=0}^{\infty}$ we can use (3.11) and the estimates established in the proof of Lemma 3.3.

From (3.9) one obtains for all $k \in \mathbb{N}_0$

$$\|p_k\| = \|\Psi(x_k, B_k) - \Psi(\hat{x}, B_k)\| \le M_\gamma\|e_k\|. \tag{3.16}$$

From (3.8), (3.10), the mean value theorem, and (3.9) we get

$$\begin{aligned} \|e_{k+1}\| \le \tfrac{M_\gamma^2}{2}\|e_k\|^2 &+ \|T(\hat{x})\hat{H}^{-1}\|[\|T(\hat{x})^*\hat{G}\|\|R(\hat{x})h'(\hat{x})p_k\| \\ &+ \|[B_k - \hat{H}]S(x_k)p_k\| + \|B_k - \hat{H}\|\|S(x_k) - S(\hat{x})\|\|p_k\|]. \end{aligned}$$

Since $S \in \mathrm{Lip}(D)$ and $S(x_k)p_k = -B_k^{-1}T(x_k)^*\nabla f(x_k)$, the assertion follows immediately from (3.11), (3.13), (3.16), and from the assumptions of the theorem. □

Remark. The result in Theorem 3.4 is valid for reduced SQP methods in the general formulation of Algorithm 2.1. In particular, it applies to every choice of the right-inverse and the null space parametrization, provided the assumption (A2) can be satisfied.

To apply Theorem 3.4 one needs for the sequence $\{x_k\}_{k=0}^{\infty}$ to converge. In the next lemma we give a two-step contraction property for the iterates from Algorithm 2.1 which ensures two-step linear convergence, provided that $\|B_k - \hat{H}\|$ is sufficiently small for all k.

LEMMA 3.5. *Let $\hat{H}$ be bijective, and suppose that* (A1), (A2) *hold. For each $r > 0$ and $\gamma > 0$ there are positive constants ε_1 and δ_1, depending on r and γ, such that if $x_- \in X$ and $B_-, B_c \in \mathcal{B}(\gamma)$ satisfy*

$$\|x_- - \hat{x}\| < \varepsilon_1 \quad \text{and} \quad \|B_c - \hat{H}\| < \delta_1,$$

then it follows that

$$\|\Phi(x_c, B_c) - \hat{x}\| \leq r\|x_- - \hat{x}\|,$$

where x_c is defined by $x_c = \Phi(x_-, B_-)$.

Proof. Let $r > 0$ and $\gamma > 0$ be given. Choose

$$c_1 \geq \|T(\hat{x})\hat{H}^{-1}\| \max\{\|T(\hat{x})^*\hat{G}\|, \|S(\hat{x})\|\}$$

and define

$$\varepsilon_1 = \min\left\{\frac{\rho}{M_\gamma}, \frac{r}{M_\gamma^4 + 2c_1c(\gamma)}\right\}, \qquad \delta_1 = \frac{r}{2c_1M_\gamma^2}$$

with $M_\gamma > 1$ from (3.9) and $c(\gamma)$ from Lemma 3.3.

Assume that $\|x_- - \hat{x}\| < \varepsilon_1$ and $B_-, B_c \in \mathcal{B}(\gamma)$. Then one obtains from (3.9) that $x_c = \Phi(x_-, B_-) \in D$, and it follows from (3.8), (3.10), (3.9), and Lemma 3.3 that

$$\begin{aligned} &\|\Phi(x_c, B_c) - \hat{x}\| \\ &\leq \tfrac{M_\gamma^2}{2}\|x_c - \hat{x}\|^2 + \|T(\hat{x})\hat{H}^{-1}\| \\ &\quad \cdot (\|B_c - \hat{H}\|\|S(\hat{x})\|\|\Psi(x_c, B_c)\| + \|T(\hat{x})^*\hat{G}\|\|R(\hat{x})h'(\hat{x})\Psi(x_c, B_c)\|) \\ &\leq (\tfrac{1}{2}(M_\gamma^4 + 2c_1c(\gamma))\|x_- - \hat{x}\| + c_1M_\gamma^2\|B_c - \hat{H}\|)\|x_- - \hat{x}\|. \end{aligned} \tag{3.17}$$

Now the assertion follows from $\|B_c - \hat{H}\| < \delta_1$ and the definition of ε_1. □

Obviously, one can deduce by Lemma 3.5 that the sequence produced by Algorithm 2.1 converges two-step q-linearly to the solution, provided that B_k is sufficiently close to $\hat{H}$ at each iteration. Both this convergence result and the sufficient criterion for two-step superlinear convergence given in Theorem 3.4 impose certain conditions on the approximating operators B_k. In the remaining part of this section we show that these conditions can be satisfied when reduced secant updates are used appropriately. But first we want to consider a version of Algorithm 2.1 where the calculation of the sequence $\{B_k\}_{k=0}^{\infty}$ is based on exact second order information. In this case one expects at least two-step q-quadratic convergence. We think that this result is worth proving because it has received only little attention in the literature.

COROLLARY 3.6. *Assume that $\hat{H}$ is bijective, and let* (A1), (A2) *hold. Suppose that Algorithm* 2.1 *is used with $B_k = T(x_k)^*L_x''(x_k, l_k)T(x_k)$, where $l_k \in Z^*$ is chosen so that*

$$\|l_k - \hat{l}\| \leq \kappa_l\|x_k - \hat{x}\| \tag{3.18}$$

is satisfied for some constant κ_l and for all $k \in \mathbb{N}_0$. If x_0 is sufficiently close to $\hat{x}$, then the sequence $\{x_k\}_{k=0}^{\infty}$ converges two-step q-quadratically to $\hat{x}$, i.e.,

$$\|x_{k+1} - \hat{x}\| \leq C\|x_{k-1} - \hat{x}\|^2$$

for some constant $C > 0$.

Proof. Let $\bar{\delta} := 1/(2\|\hat{H}^{-1}\|)$, so that by the Banach lemma (see, e.g., [26, p. 155]) for $B \in \mathcal{L}(W)$,

$$\|B - \hat{H}\| \leq \bar{\delta} \quad \Rightarrow \quad \|B^{-1}\| \leq 2\|\hat{H}^{-1}\|. \tag{3.19}$$

Since (A1), (A2), and (3.18) are assumed, there is a constant $c_0 > 0$ such that

$$\|B_k - \hat{H}\| \leq c_0\|x_k - \hat{x}\|, \quad \text{if } x_k \in D. \tag{3.20}$$

For some scalar $r \in (0,1)$ and $\gamma := 2\max\{\|\hat{H}\|, \|\hat{H}^{-1}\|\}$ define

$$\varepsilon_0 = \min\left\{\frac{\varepsilon_1}{M_\gamma}, \frac{\delta_1}{c_0 M_\gamma}, \frac{\bar{\delta}}{c_0 M_\gamma}\right\}$$

with the corresponding constants M_γ from (3.9) and ε_1, δ_1 from Lemma 3.5. Note that if $B \in \mathcal{L}(W)$ satisfies $\|B - \hat{H}\| \leq \bar{\delta}$, then (3.19) and the definition of γ imply that $B \in \mathcal{B}(\gamma)$.

Assume that $\|x_0 - \hat{x}\| < \varepsilon_0$. We show by induction that for all $k \in \mathbb{N}$

$$B_k \in \mathcal{B}(\gamma), \quad \|B_k - \hat{H}\| < \delta_1, \quad \|e_k\| < \varepsilon_1, \quad \text{and} \quad \|e_{k+1}\| \leq r\|e_{k-1}\|. \tag{3.21}$$

Since $\|x_0 - \hat{x}\| < \varepsilon_0 < \varepsilon_1 < \rho$, it follows from (3.20) that $\|B_0 - \hat{H}\| < c_0\varepsilon_0 < \min\{\delta_1, \bar{\delta}\}$, and (3.9) implies that $\|e_1\| < M_\gamma\varepsilon_0 \leq \varepsilon_1 < \rho$. Again (3.20) shows that

$$\|B_1 - \hat{H}\| < c_0 M_\gamma \varepsilon_0 \leq \min\{\delta_1, \bar{\delta}\},$$

and we can deduce from Lemma 3.5 that $\|e_2\| \leq r\|e_0\| < \varepsilon_0$. Hence, (3.20) implies that $\|B_2 - \hat{H}\| < \min\{\delta_1, \bar{\delta}\}$, and it follows again from Lemma 3.5 that $\|e_3\| \leq r\|e_1\|$.

Now suppose that (3.21) is satisfied for $k = 1, \ldots, m$ and for some $m \geq 2$. Then, $\|e_m\| < \varepsilon_1$ and $\|e_{m+1}\| < \varepsilon_1$, and (3.20) yields that

$$\|B_{m+1} - \hat{H}\| \leq c_0\|e_{m+1}\| < c_0 M_\gamma \varepsilon_0 \leq \min\{\delta_1, \bar{\delta}\}.$$

From Lemma 3.5 one obtains $\|e_{m+2}\| \leq r\|e_m\|$.

From the estimates in (3.17) and (3.20) we can deduce that the convergence is two-step quadratic. □

For the remainder of this section we will study the reduced secant methods defined in Algorithm 2.2. As indicated by Lemma 3.5 it remains to establish a bounded deterioration property for the sequence $\{B_k\}_{k=0}^{\infty}$ to ensure linear convergence. For this purpose we make use of the following inequality which can be found by thoroughly working through the proof of Griewank's [20] Lemma 2.1.

LEMMA 3.7. *Let $B \in \mathcal{L}(W)$, B self-adjoint and positive definite, $\alpha \in [0,1]$, and $v, s \in W$ with $\langle v, s\rangle > 0$ be given. Then*

$$\begin{aligned}\|B + V(B,s,v,\alpha) - I\| &\leq \|B - I\| \\ &+ \left(1 + \left[2 + \alpha\|B\|\left\{4 + \left(\frac{\|v\|^2\|s\|^2}{\langle v,s\rangle^2} - 1\right)^{\frac{1}{2}}\right\}\right]\frac{\|v\|\|s\|}{\langle v,s\rangle}\right)\frac{\|v-s\|}{\|s\|},\end{aligned} \tag{3.22}$$

where V is the update function from (2.1).

Proof. From the inequality

$$\|u\|^2 - 2\langle u, w\rangle \geq -\frac{\langle u, w\rangle^2}{\|u\|^2}$$

for $u \in W, w \in W$ with $\langle u, w\rangle \neq 0$, we deduce that

$$1 - \frac{\|u-w\|^2}{\|w\|^2} = \frac{\|w\|^2 - \|u-w\|^2}{\|w\|^2} = \frac{2\langle u, w\rangle - \|u\|^2}{\|w\|^2} \leq \frac{\langle u, w\rangle^2}{\|u\|^2\|w\|^2}. \tag{3.23}$$

Let $B \in \mathcal{L}(W)$, B self-adjoint and positive definite, $\alpha \in [0,1]$, and $v, s \in W$ with $\langle v, s\rangle > 0$ be given. A straightforward calculation shows that

$$V(B,s,v,\alpha) - V(B,s,s,\alpha) = \frac{v\otimes v}{\langle v,s\rangle} - \frac{s\otimes s}{\|s\|^2} + \alpha\frac{(Bs)\otimes z + z\otimes(Bs)}{\|s\|} + \frac{\alpha}{2}\frac{\langle s, Bs\rangle}{\|s\|^2}\left[z\otimes\left(z-\frac{2s}{\|s\|}\right) + \left(z-\frac{2s}{\|s\|}\right)\otimes z\right],$$

where the vector z is defined by

$$z = \frac{s}{\|s\|} - \frac{\|s\|}{\langle v, s\rangle}v.$$

Using the Cauchy–Schwarz inequality and the identity $\|u\otimes w\| = \|u\|\|w\|$ for $u, w \in W$, one obtains

$$\left\|\frac{v\otimes v}{\langle v,s\rangle} - \frac{s\otimes s}{\|s\|^2}\right\| \leq \left\|\frac{v\otimes v}{\langle v,s\rangle} - \frac{s\otimes v}{\langle v,s\rangle}\right\| + \left\|\frac{s\otimes v}{\langle v,s\rangle} - \frac{s\otimes v}{\|s\|^2}\right\| + \frac{\|s\otimes(v-s)\|}{\|s\|^2}$$
$$\leq \frac{\|v-s\|}{\|s\|}\left(1 + 2\frac{\|v\|\|s\|}{\langle v,s\rangle}\right).$$

Furthermore, one finds by (3.23) and the definition of z that

$$\|z\|^2 = \frac{\|v\|^2\|s\|^2}{\langle v,s\rangle^2} - 1 \leq \frac{\|v\|^2\|s\|^2}{\langle v,s\rangle^2}\frac{\|v-s\|^2}{\|s\|^2}.$$

Combining these identities and estimates, we arrive at

$$\|V(B,s,v,\alpha) - V(B,s,s,\alpha)\|$$
$$\leq \frac{\|v-s\|}{\|s\|}\left(1 + 2\,\frac{\|v\|\|s\|}{\langle v,s\rangle}\right) + \alpha\|B\|\|z\|\left(2 + \left\|z - \frac{2s}{\|s\|}\right\|\right)$$
$$\leq \left(1 + [2 + \alpha\|B\|\,\{4 + \|z\|\}]\,\frac{\|v\|\|s\|}{\langle v,s\rangle}\right)\frac{\|v-s\|}{\|s\|}.$$

Using the definition of the spectral bounds of a self-adjoint operator Griewank [20] shows that

$$\|B + V(B,s,s,\alpha) - I\| \leq \|B - I\|,$$

and the assertion follows immediately from the last two estimates. □

Lemma 3.7 can be applied iteratively to show that the error $\|B_k - \hat{H}\|$ can be maintained sufficiently small, provided that the sequences $\{s_k\}_{k=0}^{\infty}$ and $\{v_k\}_{k=0}^{\infty}$ in the update are defined appropriately. In the next theorem we give a condition for v_k, s_k which ensures a bounded deterioration property for $\{B_k\}_{k=0}^{\infty}$ with respect to the operator norm. If the initial approximations are sufficiently accurate, this property guarantees that B_k is sufficiently close to $\hat{H}$ at each iteration, and two-step q-linear convergence can be achieved.

THEOREM 3.8. *Assume* (A1)–(A3) *and let* $\{x_k\}_{k=0}^{\infty}$ *be generated by Algorithm* 2.2 *with* $B_0 \in \mathcal{L}(W)$ *self-adjoint and positive. Assume further that the sequences* $\{s_k\}_{k=0}^{\infty}$ *and* $\{v_k\}_{k=0}^{\infty}$ *are chosen such that for a constant* $\beta > 0$

$$\langle v_k, s_k \rangle > 0 \quad \text{and} \quad \|v_k - \hat{H}s_k\| \leq \beta\sigma(x_k, x_{k+1})\|s_k\| \tag{3.24}$$

is fulfilled in each iteration with $s_k \neq 0$. *Then for each* $r \in (0,1)$, *there are positive constants* $\varepsilon = \varepsilon(r)$ *and* $\delta = \delta(r)$ *such that if*

$$\|x_0 - \hat{x}\| < \varepsilon \quad \text{and} \quad \|B_0 - \hat{H}\| < \delta,$$

the sequence $\{x_k\}_{k=0}^{\infty}$ *is well-defined and converges to* $\hat{x}$. *Furthermore,*

$$\|x_{k+1} - \hat{x}\| \leq r\|x_{k-1} - \hat{x}\|$$

for each $k \geq 1$, *and* $\{\|B_k\|\}_{k=0}^{\infty}$, $\{\|B_k^{-1}\|\}_{k=0}^{\infty}$ *are bounded.*

Proof. Let $\hat{H}^{1/2}$ and $\hat{H}^{-1/2} = (\hat{H}^{1/2})^{-1}$ denote the square root of $\hat{H}$ and $\hat{H}^{-1}$, respectively. Then $\hat{H}^{1/2}$ and $\hat{H}^{-1/2}$ are self-adjoint and positive definite operators in $\mathcal{L}(W)$. Consider the transformed sequences

$$\bar{s}_k = \hat{H}^{1/2}s_k \quad \text{and} \quad \bar{v}_k = \hat{H}^{-1/2}v_k \quad \text{and} \quad \bar{B}_k = \hat{H}^{-1/2}B_k\hat{H}^{-1/2}.$$

Obviously, $\langle \bar{v}_k, \bar{s}_k \rangle = \langle v_k, s_k \rangle$ for all $k \in \mathbb{N}_0$, and $\bar{B}_{k+1} = \bar{B}_k + V(\bar{B}_k, \bar{s}_k, \bar{v}_k, \alpha_k)$, if the update function V is defined in $(B_k, s_k, v_k, \alpha_k)$.

Let $r \in (0,1)$ be given. Let $c_2 \geq \|\hat{H}\|\|\hat{H}^{-1}\|$, and set

$$\bar{\delta} = 1/(2\|\hat{H}^{-1}\|), \quad \gamma = 2\max\{\|\hat{H}\|, \|\hat{H}^{-1}\|\}.$$

With the corresponding constants M_γ from (3.9), ε_1, δ_1 from Lemma 3.5, and with β from (3.24), define

$$\delta = \min\left\{\frac{\delta_1}{2c_2}, \frac{\bar{\delta}}{2c_2}\right\} \quad \text{and}$$

$$\varepsilon = \min\left\{\frac{\varepsilon_1}{M_\gamma}, \frac{\delta(1-r)}{8M_\gamma\beta(1+\gamma^2)}, \frac{4\bar{\delta}}{3M_\gamma\beta}\right\}.$$

Recall from (3.19) and the definition of γ that each $B \in \mathcal{L}(W)$ with $\|B - \hat{H}\| \leq \bar{\delta}$ belongs to $\mathcal{B}(\gamma)$.

Now suppose that

$$\|x_0 - \hat{x}\| < \varepsilon \quad \text{and} \quad \|B_0 - \hat{H}\| < \delta.$$

We will establish by induction that for all $k \geq 1$ (i) $B_{k-1} \in \mathcal{B}(\gamma), B_k \in \mathcal{B}(\gamma)$, and B_{k-1}, B_k are self-adjoint and positive definite.

$$\|x_{k-1} - \hat{x}\| < \varepsilon_1 \quad \text{and} \quad \|B_k - \hat{H}\| < \delta_1.$$

(ii) $\|\bar{B}_k - I\| \le \|\bar{B}_{k-1} - I\| + 4\beta(1+\gamma^2)\|\hat{H}^{-1}\|\sigma(x_{k-1}, x_k)$.

(iii) It holds that

$$\left.\begin{aligned} \|x_k - \hat{x}\| &\le r^{\frac{k-1}{2}}\|x_1 - \hat{x}\| \\ \sum_{\nu=0}^{k} \sigma(x_\nu, x_{\nu+1}) &\le 2\sigma(x_0, x_1) \sum_{\nu=0}^{\frac{k-1}{2}} r^\nu \end{aligned}\right\} \text{ if } k \text{ is odd.}$$

$$\left.\begin{aligned} \|x_k - \hat{x}\| &\le r^{\frac{k}{2}}\|x_0 - \hat{x}\| \\ \sum_{\nu=0}^{k} \sigma(x_\nu, x_{\nu+1}) &\le 2\sigma(x_0, x_1) \sum_{\nu=0}^{\frac{k-2}{2}} r^\nu + r^{\frac{k}{2}}\sigma(x_0, x_1) \end{aligned}\right\} \text{ if } k \text{ is even.}$$

Then, the assertion of the theorem follows from (i) and Lemma 3.5.

The definition of δ yields that $\|B_0 - \hat{H}\| < \min\{\delta_1, \bar{\delta}\}$ which implies that $B_0 \in \mathcal{B}(\gamma)$. If $s_0 = 0$, then $B_1 = B_0$, and (i)–(iii) are trivially satisfied for $k = 1$. Let $s_0 \neq 0$. Then it follows from (3.24) and the definition of ε that

$$\frac{\|\bar{v}_0 - \bar{s}_0\|}{\|\bar{s}_0\|} \le \beta\|\hat{H}^{-1}\|\sigma(x_0, x_1) < 2/3.$$

This implies by (3.23)

$$\frac{\langle \bar{v}_0, \bar{s}_0\rangle}{\|\bar{v}_0\|\|\bar{s}_0\|} > 2/3 \quad \text{and} \quad \left(\frac{\|\bar{v}_0\|^2\|\bar{s}_0\|^2}{\langle \bar{v}_0, \bar{s}_0\rangle^2} - 1\right)^{\frac{1}{2}} < 1.$$

We use these estimates in Lemma 3.7 and recall the definition of γ and $\bar{\delta}$ to get

$$\begin{aligned} \|B_1 - \hat{H}\| &\le \|\hat{H}\|\|\bar{B}_1 - I\| < \|\hat{H}\|(\|\bar{B}_0 - I\| + 4\beta(1 + 2\|\bar{B}_0\|)\|\hat{H}^{-1}\|\sigma(x_0, x_1)) \\ &\le \|\hat{H}\|(\|\bar{B}_0 - I\| + 4\beta(1+\gamma^2)\|\hat{H}^{-1}\|\sigma(x_0, x_1)) \\ &\le c_2(\|B_0 - \hat{H}\| + 4\beta(1+\gamma^2)\sigma(x_0, x_1)). \end{aligned}$$

In particular, this set of inequalities shows (ii) for $k = 1$. Since

$$\sigma(x_0, x_1) \le M_\gamma\|x_0 - \hat{x}\| < \frac{\delta}{8\beta(1+\gamma^2)},$$

it follows from the same estimates and from the definition of δ that $\|B_1 - \hat{H}\| < \min\{\delta_1, \bar{\delta}\}$. It is obvious that B_1 is maintained self-adjoint and positive definite, since $\langle v_0, s_0\rangle > 0$. This completes the proof of (i) for $k = 1$. Now we can conclude from Lemma 3.5 that $\|e_2\| \le r\|e_0\|$ which yields (iii) for $k = 1$.

If $s_1 \neq 0$, then (3.24) is satisfied and $B_2 = B_1 + V(B_1, s_1, v_1, \alpha_1)$. We apply (ii) and (iii) for $k = 1$ and use the same arguments as before to obtain

$$\begin{aligned} \|B_2 - \hat{H}\| &\le \|\hat{H}\|\|\bar{B}_2 - I\| < \|\hat{H}\|(\|\bar{B}_1 - I\| + 4\beta(1+\gamma^2)\|\hat{H}^{-1}\|\sigma(x_1, x_2)) \\ &\le c_2(\|B_0 - \hat{H}\| + 8\beta(1+\gamma^2)\sigma(x_0, x_1)) < \min\{\delta_1, \bar{\delta}\}. \end{aligned}$$

Since $\|e_1\| \le M_\gamma\|e_0\| < \varepsilon_1$, it follows that (i) and (ii) are true for $k = 2$. Again we deduce from Lemma 3.5 that $\|e_3\| \le r\|e_1\|$. This estimate and (iii) for $k = 1$ show that (iii) also holds for $k = 2$. Clearly, the conditions (i)–(iii) are fulfilled for $k = 2$ in the case $s_1 = 0$.

Now, assuming (i)–(iii) are true for $k = 1, \dots, m$ for some $m \geq 2$, we want to show that they hold for $k = m + 1$.

First note that (iii) implies that $\|e_m\| < \varepsilon_1$. Let $s_m \neq 0$. By (3.24)

$$\frac{\|\bar{v}_m - \bar{s}_m\|}{\|\bar{s}_m\|} \leq \beta \|\hat{H}^{-1}\| \sigma(x_m, x_{m+1}) \leq \beta \|\hat{H}^{-1}\| \sigma(x_0, x_1) < 2/3.$$

Now, exactly the same steps taken for $k = 1$, together with (ii), (iii), $k = 1, \dots, m$, lead to

$$\begin{aligned} \|B_{m+1} - \hat{H}\| &< \|\hat{H}\|(\|\bar{B}_m - I\| + 4\beta(1+\gamma^2)\|\hat{H}^{-1}\|\sigma(x_m, x_{m+1})) \\ &\leq \|\hat{H}\|(\|\bar{B}_0 - I\| + 4\beta(1+\gamma^2)\|\hat{H}^{-1}\| \textstyle\sum_{\nu=0}^{m} \sigma(x_\nu, x_{\nu+1})) \\ &\leq c_2(\|B_0 - \hat{H}\| + 8\beta(1+\gamma^2)(1-r)^{-1}\sigma(x_0, x_1)). \end{aligned}$$

These inequalities imply (ii) for $k = m + 1$. Moreover, since

$$\sigma(x_0, x_1) < M_\gamma \varepsilon \leq \frac{\delta(1-r)}{8\beta(1+\gamma^2)},$$

it follows that $\|B_{m+1} - \hat{H}\| < \min\{\delta_1, \bar{\delta}\}$. Now the arguments used before can be applied without change to ensure that (i) is satisfied for $k = m+1$ and that $\|e_{m+2}\| \leq r\|e_m\|$. This estimate and (iii), $k = 1, \dots, m$, finally yield (iii) for $k = m+1$. Obviously, the assertions (i)–(iii) are satisfied in the case $B_{m+1} = B_m$. □

The convergence result of Theorem 3.8 allows the use of any sequences $\{v_k\}_{k=0}^{\infty}$ and $\{s_k\}_{k=0}^{\infty}$ which satisfy the condition (3.24). In particular, we want to consider a reduced secant method with

$$s_k = w_k = -B_k^{-1} T(x_k)^* \nabla f(x_k), \quad \text{and} \tag{3.25a}$$

$$v_k = T(x_k)^* [\nabla_x L(x_k + T(x_k)w_k, l_k) - \nabla f(x_k)], \tag{3.25b}$$

$$\text{or} \quad v_k = T(x_k + T(x_k)w_k)^* \nabla f(x_k + T(x_k)w_k) - T(x_k)^* \nabla f(x_k). \tag{3.25c}$$

We note that the definition (3.25a), (3.25b), which is the choice proposed and studied by Coleman and Conn [9], can be obtained by applying one quasi-Newton iteration to solve the unconstrained problem $\min L(x_k + T(x_k)w, l_k)$ over $w \in W$, where the objective of this problem is approximated by a quadratic model around $w = 0$.

In the second formula for v_k one only needs the reduced gradient of f, and the calculation l_k of an approximation to the Lagrange multiplier is avoided. This can be advantageous when the separability approach is applied as will be demonstrated for optimal control problems in §4. Both definitions for v_k have the drawback that two gradient evaluations are needed at each iteration. This is in contrast to the formulas studied by Nocedal and Overton [36] which require only one gradient calculation per iteration. However, in their analysis a more complicated updating strategy is used.

It follows from standard arguments that (3.24) can be satisfied if the definition (3.25) is used in step (3) of Algorithm 2.2. Consequently, the corresponding iterates $\{x_k\}_{k=0}^{\infty}$ converge to the solution at least two-step linearly.

COROLLARY 3.9. *Let* (A1)–(A3) *hold. Assume that* $\{x_k\}_{k=0}^{\infty}$ *is generated from Algorithm* 2.2 *with* $B_0 \in \mathcal{L}(W)$ *self-adjoint and positive, and with* s_k, v_k *defined by* (3.25a), (3.25b) *or* (3.25a), (3.25c). *In the former case assume further that* $l_k \in Z^*$ *is chosen so that* (3.18) *is satisfied. If* x_0 *and* B_0 *are sufficiently close to* $\hat{x}$ *and* $\hat{H}$, *respectively, then the sequence* $\{x_k\}_{k=0}^{\infty}$ *converges to* $\hat{x}$ *at least two-step q-linearly.*

Proof. For $x \in D$ and $B \in \mathcal{L}(W)$, B bijective, define

$$w(x,B) = -B^{-1}T(x)^*\nabla f(x).$$

It follows from (A2) that there is a constant $M > 0$ such that

$$\|T(x) - T(y)\| \le M\|x - y\| \quad \text{for all } x, y \in D.$$

We suppose that M is chosen sufficiently large, so that M is a generic Lipschitz constant for the following mappings which, by (A1),(A2), belong to $\mathrm{Lip}(D)$:

$$L_x''(\cdot,\hat{l}), \quad T(\cdot)^*L_x''(\cdot,\hat{l})T(\cdot), \quad \nabla_x L(\cdot,\hat{l}), \quad R(\cdot)h(\cdot).$$

For $x \in D$ and $B \in \mathcal{L}(W)$, B bijective, we have

$$\begin{aligned} \|x + T(x)w(x,B) - \hat{x}\| &\le \|x + \Psi(x,B) - \hat{x}\| + \|R(x)h(x) - R(\hat{x})h(\hat{x})\| \\ &\le \|\Phi(x,B) - \hat{x}\| + M\|x - \hat{x}\|, \quad \text{and} \\ \|T(x)w(x,B)\| &= \|\Phi(x,B) - \hat{x} - x + \hat{x} + R(x)h(x) - R(\hat{x})h(\hat{x})\| \\ &\le \|\Phi(x,B) - \hat{x}\| + (1+M)\|x - \hat{x}\|. \end{aligned} \tag{3.26}$$

Now assume that x_c and $x_+ = x_c + \Psi(x_c, B_c)$ belong to the open ball with radius $\rho/(1+M)$ around $\hat{x}$. For abbreviation we set

$$w_c = w(x_c, B_c) \quad \text{and} \quad \tilde{x} = x_c + T(x_c)w_c.$$

Then, by (3.26), $\|\tilde{x} - \hat{x}\| \le (1+M)\sigma(x_c, x_+) < \rho$, which shows that $T(\tilde{x})^*\nabla f(\tilde{x})$ is well defined. Furthermore, it follows from (3.26) that $\|T(x_c)w_c\| \le (2+M)\sigma(x_c, x_+)$. Using this estimate and the mean value theorem we get

$$\begin{aligned} &\|\nabla_x L(\tilde{x},\hat{l}) - \nabla_x L(x_c,\hat{l}) - L_x''(x_c,\hat{l})T(x_c)w_c\| \\ &\le \int_0^1 (\|L_x''(x_c + tT(x_c)w_c,\hat{l}) - \hat{G}\| + \|L_x''(x_c,\hat{l}) - \hat{G}\|)\, dt\, \|T(x_c)w_c\| \\ &\le M[2\|x_c - \hat{x}\| + \|T(x_c)w_c\|]\|T(x_c)w_c\| \le M(4+M)\sigma(x_c,x_+)\|T(x_c)\|\|w_c\|. \end{aligned}$$

From $\|\tilde{x} - \hat{x}\| \le (1+M)\sigma(x_c, x_+)$ and from the definition of M we obtain

$$\begin{aligned} &\|[T(\tilde{x}) - T(x_c)]^*[\nabla_x L(\tilde{x},\hat{l}) - \nabla_x L(\hat{x},\hat{l})]\| \\ &\le M^2\|T(x_c)w_c\|\|\tilde{x} - \hat{x}\| \le \|T(x_c)\|M^2(1+M)\sigma(x_c,x_+)\|w_c\|. \end{aligned}$$

Combining these estimates we conclude that

$$\begin{aligned} &\|T(\tilde{x})^*\nabla f(\tilde{x}) - T(x_c)^*\nabla f(x_c) - \hat{H}w_c\| \\ &\le \|T(x_c)^*[\nabla_x L(\tilde{x},\hat{l}) - \nabla_x L(x_c,\hat{l}) - L_x''(x_c,\hat{l})T(x_c)w_c]\| \\ &\quad + \|[T(x_c)^*L_x''(x_c,\hat{l})T(x_c) - \hat{H}]w_c\| + \|[T(\tilde{x}) - T(x_c)]^*[\nabla_x L(\tilde{x},\hat{l}) - \nabla_x L(\hat{x},\hat{l})]\| \\ &\le [\|T(x_c)\|^2 M(4+M) + M + \|T(x_c)\|M^2(1+M)]\sigma(x_c,x_+)\|w_c\|. \end{aligned}$$

This shows that there is a constant $\beta > 0$, β only depending on M, such that if $\sigma(x_c, x_+) < \rho/(1+M)$, then

$$\|T(\tilde{x})^*\nabla f(\tilde{x}) - T(x_c)^*\nabla f(x_c) - \hat{H}w_c\| \le \beta\sigma(x_c,x_+)\|w_c\|. \tag{3.27}$$

With m from (A3) define $\bar{m} = m/(2\|T(\hat{x})^{-1}\|^2)$, and set

$$\bar{\rho} = \min\{\rho/(1+M), \bar{m}/\beta\} \quad \text{and} \quad \bar{D} = \{x \in X : \|x - \hat{x}\| < \bar{\rho}\}.$$

Using the Cauchy–Schwarz inequality, (3.27), and (A3), it follows for $x_c, x_+ \in \bar{D}$:

(3.28)
$$\langle T(\tilde{x})^* \nabla f(\tilde{x}) - T(x_c)^* \nabla f(x_c), w_c \rangle \geq \langle w_c, \hat{H} w_c \rangle - \beta\sigma(x_c, x_+)\|w_c\|^2 \geq \bar{m}\|w_c\|^2.$$

With β from (3.27) define the constants ε and δ as in Theorem 3.8. If necessary, restrict ε further so that $\varepsilon \leq \bar{\rho}/M_\gamma$ with M_γ from the proof of Theorem 3.8. Now assume that $\|x_0 - \hat{x}\| < \varepsilon$ and $\|B_0 - \hat{H}\| < \delta$, and let $\{s_k\}_{k=0}^\infty$, $\{v_k\}_{k=0}^\infty$ in Algorithm 2.2 be constructed from (3.25a), (3.25c). Then we can use the induction arguments from the proof of Theorem 3.8 again to show that $x_k \in \bar{D}$ for all $k \in \mathbb{N}_0$, and to deduce from (3.27) and (3.28) the two-step linear convergence of $\{x_k\}_{k=0}^\infty$.

Similar arguments can be used to prove the assertion for the choice (3.25a), (3.25b). □

Now we turn our attention to the superlinear convergence behavior of reduced secant methods. It is well known from unconstrained optimization that in the infinite dimensional case one can expect a quasi-Newton method to be only linearly convergent, unless a certain compactness condition is satisfied; see, e.g., Griewank [20] and Kelley and Sachs [27]. Similarly, under the assumptions of Theorem 3.8, (two-step) superlinear convergence can be guaranteed for Algorithm 2.2, provided that B_0 approximates the true reduced Hessian $\hat{H}$ up to a compact operator.

At this point we must look for conditions which ensure that the consistency property (3.15) is satisfied. Besides the compactness of the initial discrepancy $B_0 - \hat{H}$, the condition (3.24), which was essential in the proof of linear convergence, will play an important role again. For this reason we make a few comments on this condition before proceeding with our convergence results.

More generally, we consider the following condition which contains (3.24) in the case that $\{x_k\}_{k=0}^\infty$ is linearly convergent.

(U) Let $\{s_k\}_{k=0}^\infty$ and $\{v_k\}_{k=0}^\infty$ be sequences in W such that for all $k \in \mathbb{N}_0$

$$\langle v_k, s_k \rangle > 0 \quad \text{and} \quad \|v_k - \hat{D} s_k\| \leq \varepsilon_k \|s_k\|$$

for some self-adjoint and positive definite operator $\hat{D} \in \mathcal{L}(W)$, and for some sequence $\{\varepsilon_k\}_{k=0}^\infty$ with the property $\sum_{k=0}^\infty \varepsilon_k < \infty$.

In Theorem 3.8 we proved linear convergence, making use of a bounded deterioration property that was given in the operator norm. This property was established by combining the estimate (3.22) and the condition (3.24). In finite-dimensional convergence analyses of secant methods a corresponding version of (3.24) is commonly used to establish a certain bounded deterioration property in a weighted Frobenius norm; see Broyden, Dennis, and Moré [3]. This property can be applied to prove both linear *and* superlinear convergence, since the use of the Frobenius norm yields a sharper bound than (3.22). Unfortunately, this technique to prove superlinear convergence cannot be adopted in the infinite-dimensional case, since Frobenius norm estimates are not available. In this event a result by Byrd and Nocedal [6] may be of interest. They show that the finite-dimensional BFGS update possesses a property that is very useful for establishing convergence results without the need of the Frobenius norm.

More specifically, it is proved in [6, Theorem 3.2], that if $\{B_k\}_{k=0}^{\infty}$ is generated by the BFGS formula, i.e., $B_{k+1} = B_k + V(B_k, s_k, v_k, 0)$, where B_0 is symmetric and positive definite, and where $\{v_k\}_{k=0}^{\infty}$ and $\{s_k\}_{k=0}^{\infty}$ satisfy the finite- dimensional analog of property (U), then

$$\lim_{k\to\infty} \frac{\|(B_k - \hat{D})s_k\|}{\|s_k\|} = 0.$$

This statement does not depend on the algorithmic framework of the update, and it is relevant to both constrained and unconstrained optimization. Moreover, the theorem and its proof are independent of any particular bounded deterioration property in the Frobenius norm. This makes the result attractive for the superlinear convergence theory of quasi-Newton methods in infinite-dimensional spaces. Indeed, we extend Theorem 3.2 in [6] to the Hilbert space setting, and, furthermore, we generalize it for the updates of the convex Broyden family. The proof of this infinite-dimensional extension is essentially based on results established by Griewank [20]. In particular, the following general theorem is needed, which is a special case of Theorem 3.2 in [20].

THEOREM 3.10. *Let $\{A_k\}_{k=0}^{\infty}$ be any sequence of self-adjoint and positive operators with $A_k - I \in \mathcal{COM}(W), k \in \mathbb{N}_0$. Assume there exist operators $\{D_k\}_{k=1}^{\infty}$ with the same properties as A_k, such that for some sequence $\{d_k\}_{k=0}^{\infty}$ in $W \setminus \{0\}$, some constant τ, and for all $k \geq 0$ the following conditions are satisfied:*

(i) $D_{k+1}d_k = d_k$,

(ii) $\langle w, D_{k+1}w\rangle \leq \langle w, A_k w\rangle$ *for all* $w \in W$ *with* $\langle w, d_k\rangle = 0$,

(iii) $\|A_{k+1} - D_{k+1}\| \leq \tau\varepsilon_k$

for some sequence $\{\varepsilon_k\}_{k=0}^{\infty}$ with the property $\sum_{k=0}^{\infty}\varepsilon_k < \infty$.

Then

$$\limsup_{k\to\infty} \frac{\langle d_k, A_k d_k\rangle}{\|d_k\|^2} \leq 1.$$

Proof. The assertion would follow immediately from Theorem 3.2 in [20], if the essential norm of A_k were equal one for all $k \in \mathbb{N}_0$. Since the latter condition is not satisfied in general, we consider an extension of spaces and operators which in a similar form can also be found in [20].

Set $H = W \times U$, where U is any real Hilbert space with $\dim U = \infty$. If the norm in H is defined by $\|(w,u)\| = (\|w\|^2 + \|u\|^2)^{\frac{1}{2}}$, then H is a real infinite-dimensional Hilbert space. For $Q \in \mathcal{L}(W)$ let $\tilde{Q}$ be defined as

$$\tilde{Q} : \begin{cases} H & \to & H, \\ (w,u) & \to & (Qw,u). \end{cases}$$

Now, consider the sequences $\{\tilde{A}_k\}_{k=0}^{\infty}$, $\{\tilde{D}_k\}_{k=1}^{\infty}$, and set $\tilde{d}_k = (d_k, 0), k \in \mathbb{N}_0$. Obviously, $\tilde{A}_k, \tilde{D}_k$ are positive, self-adjoint operators in $\mathcal{L}(H)$ with $\tilde{A}_k - I_H \in \mathcal{COM}(H)$ and $\tilde{D}_k - I_H \in \mathcal{COM}(H)$ for all k. Furthermore, since $\dim U = \infty$, it follows directly from the definition of the operator $\tilde{Q}$ that the essential norm $\lambda_*(\tilde{A}_k)$ of $\tilde{A}_k$, defined as

$$\lambda_*(\tilde{A}_k) = \inf\{\|\tilde{A}_k - C\| \,;\, C \in \mathcal{COM}(H)\},$$

satisfies $\lambda_*(\tilde{A}_k) \geq 1$. Consequently, $\lambda_*(\tilde{A}_k) = 1$ for $k \geq 0$, because of the compactness of $\tilde{A}_k - I_H$.

Since $\|\tilde{A} - \tilde{D}\|_{\mathcal{L}(H)} = \|A - D\|_{\mathcal{L}(W)}$ for $A, D \in \mathcal{L}(W)$, it follows from the assumptions (i)–(iii) that these three properties also hold for the extended sequences $\{\tilde{A}_k\}_{k=0}^{\infty}$, $\{\tilde{D}_k\}_{k=1}^{\infty}$, and $\{\tilde{d}_k\}_{k=0}^{\infty}$. Now, we are in the situation to apply Theorem 3.2 in [20], and we obtain that

$$1 \geq \limsup_{k\to\infty} \frac{\langle \tilde{d}_k, \tilde{A}_k \tilde{d}_k \rangle}{\|\tilde{d}_k\|^2} = \limsup_{k\to\infty} \frac{\langle d_k, A_k d_k \rangle}{\|d_k\|^2},$$

which is the assertion of the theorem. □

Remark. Griewank [20] exploits the properties of the spectral values of self-adjoint operators to study the convergence of secant methods when applied to the solution of nonlinear operator equations in Hilbert space. He shows that the asymptotic rate of convergence depends on the essential norm of the discrepancy between the derivative at the solution and its initial approximation. In particular, local superlinear convergence is obtained if this discrepancy is compact. Theorem 3.10 is a version of Theorem 3.2 in [20] for the case that $A_k - I$ is a compact operator. We decided to quote the result in [20] only for this special case, which is sufficient for our purposes, since in this way the need for additional notation and definitions is avoided.

Remark. Griewank [20] shows under certain conditions that the properties (i)–(iii) in Theorem 3.10 are fulfilled for the choice

$$d_k = s_k \quad \text{and} \quad A_k = B_k \quad \text{and} \quad D_{k+1} = B_k + V(B_k, s_k, s_k, \alpha_k),$$

where $\{B_k\}_{k=0}^{\infty}$ is generated by a secant formula from the convex Broyden class with $s_k = -B_k^{-1} g(w_k)$ and $v_k = g(w_k + s_k) - g(w_k)$ given from the quasi-Newton iteration to solve $g(w) = 0$. The same relationship is established for the corresponding sequences of inverse operators. Then Theorem 3.10 is applied to show that the Dennis and Moré characterization of superlinear convergence [11, Theorem 2.2] is satisfied, provided that $B_0 - I$ is compact.

We assume more generally that $\{v_k\}_{k=0}^{\infty}$ and $\{s_k\}_{k=0}^{\infty}$ satisfy the property (U) and use similar arguments as Griewank [20] to prove the following result which is the extension of Theorem 3.2 of Byrd and Nocedal [6] to the infinite-dimensional case.

COROLLARY 3.11. *Let $\{B_k\}_{k=0}^{\infty}$ be generated according to*

$$B_{k+1} = B_k + V(B_k, s_k, v_k, \alpha_k) \quad \textit{with } \alpha_k \in [0, 1],$$

where $B_0 \in \mathcal{L}(W)$ is self-adjoint and positive definite, and where $\{s_k\}_{k=0}^{\infty}$ and $\{v_k\}_{k=0}^{\infty}$ satisfy the condition (U). *If $B_0 - \hat{D}$ is compact, and if the sequences $\{\|B_k\|\}_{k=0}^{\infty}$ and $\{\|B_k^{-1}\|\}_{k=0}^{\infty}$ are bounded, then*

$$\lim_{k\to\infty} \frac{\|(B_k - \hat{D}) s_k\|}{\|s_k\|} = 0,$$

where $\hat{D}$ is the operator given in (U).

Proof. Let $\hat{D}^{1/2}$ and $\hat{D}^{-1/2}$ denote the square root of $\hat{D}$ and $\hat{D}^{-1}$, respectively. For $k \geq 0$ define $\bar{s}_k := \hat{D}^{1/2} s_k$, $\bar{v}_k := \hat{D}^{-1/2} v_k$, and set

$$\bar{B}_k := \hat{D}^{-1/2} B_k \hat{D}^{-1/2} \quad \text{and} \quad D_{k+1} := \bar{B}_k + V(\bar{B}_k, \bar{s}_k, \bar{s}_k, \alpha_k).$$

Since $\langle v_k, s_k\rangle > 0$, the operators $\bar{B}_k$ and D_{k+1} are well defined, self-adjoint, and positive definite for all $k \geq 0$. Let γ be a constant with $\|\bar{B}_k\| \leq \gamma$ and $\|\bar{B}_k^{-1}\| \leq \gamma$ for all $k \in \mathbb{N}_0$.

Since $\varepsilon_k \to 0$, there is $k_0 \in \mathbb{N}_0$ such that $\varepsilon_k \leq 2/(3\|\hat{D}^{-1}\|)$ for all $k \geq k_0$. We show that with $d_k = \bar{s}_{k_0+k}, k \geq 0$, the assumptions of Theorem 3.10 are satisfied for the sequences $\{\bar{B}_k\}_{k=k_0}^{\infty}$, $\{D_{k+1}\}_{k=k_0}^{\infty}$ and $\{\bar{B}_k^{-1}\}_{k=k_0}^{\infty}$, $\{D_{k+1}^{-1}\}_{k=k_0}^{\infty}$, respectively.

Obviously, $D_{k+1}\bar{s}_k = \bar{s}_k$ for all $k \geq 0$, and for $w \in W$ with $\langle w, \bar{s}_k\rangle = 0$ we have

$$\langle w, D_{k+1}w\rangle = \langle w, \bar{B}_k w\rangle - (1-\alpha_k)\frac{\langle w, \bar{B}_k\bar{s}_k\rangle^2}{\langle \bar{s}_k, \bar{B}_k\bar{s}_k\rangle} \leq \langle w, \bar{B}_k w\rangle.$$

It follows from an easy calculation that

$$D_{k+1}^{-1} = \bar{B}_k^{-1} + V(\bar{B}_k^{-1}, \bar{s}_k, \bar{s}_k, \mu_k),$$

$$\text{where} \qquad \mu_k = \frac{1-\alpha_k}{1-\alpha_k(1-\langle \bar{s}_k, \bar{B}_k^{-1}\bar{s}_k\rangle\langle \bar{s}_k, \bar{B}_k\bar{s}_k\rangle/\|\bar{s}_k\|^4)}.$$

Clearly, $D_{k+1}^{-1}\bar{s}_k = \bar{s}_k$, and, since $\mu_k \in [0,1]$, one obtains for $w \perp \bar{s}_k$:

$$\langle w, D_{k+1}^{-1}w\rangle = \langle w, \bar{B}_k^{-1}w\rangle - (1-\mu_k)\frac{\langle w, \bar{B}_k^{-1}\bar{s}_k\rangle^2}{\langle \bar{s}_k, \bar{B}_k^{-1}\bar{s}_k\rangle} \leq \langle w, \bar{B}_k^{-1}w\rangle.$$

Since $\varepsilon_k \leq 2/(3\|\hat{D}^{-1}\|)$ for $k \geq k_0$, we deduce from (U) and (3.23) that for $k \geq k_0$:

$$\frac{\|\bar{v}_k\|\|\bar{s}_k\|}{\langle \bar{v}_k, \bar{s}_k\rangle} < 3/2 \quad \text{and} \quad \left(\frac{\|\bar{v}_k\|^2\|\bar{s}_k\|^2}{\langle \bar{v}_k, \bar{s}_k\rangle^2} - 1\right)^{\frac{1}{2}} < 1.$$

We use the estimates from the proof of Lemma 3.7 to get

$$\|\bar{B}_{k+1} - D_{k+1}\| = \|V(\bar{B}_k, \bar{s}_k, \bar{v}_k, \alpha_k) - V(\bar{B}_k, \bar{s}_k, \bar{s}_k, \alpha_k)\| \leq \tau\varepsilon_k \quad \text{for } k \geq k_0,$$

where $\tau := 4\|\hat{D}^{-1}\|(1+2\gamma)$. Furthermore, it is easily checked that

$$\|D_{k+1}^{-1}\| \leq 1 + 3\|\bar{B}_k^{-1}\| \leq 1 + 3\gamma, \quad k \in \mathbb{N}_0,$$

and it follows for $k \geq k_0$:

$$\|\bar{B}_{k+1}^{-1} - D_{k+1}^{-1}\| = \|\bar{B}_{k+1}^{-1}[\bar{B}_{k+1} - D_{k+1}]D_{k+1}^{-1}\| \leq \gamma(1+3\gamma)\tau\varepsilon_k.$$

Note that $\bar{B}_0 - I \in \mathcal{COM}(W)$ implies that also $\bar{B}_k - I$, $D_{k+1} - I$, and, consequently, $\bar{B}_k^{-1} - I$ and $D_{k+1}^{-1} - I$ are compact operators in W for all $k \geq 0$.

Now we can deduce from Theorem 3.10 that

$$\limsup_{k\to\infty} \frac{\langle \bar{s}_k, \bar{B}_k\bar{s}_k\rangle}{\|\bar{s}_k\|^2} \leq 1 \quad \text{and} \quad \limsup_{k\to\infty} \frac{\langle \bar{s}_k, \bar{B}_k^{-1}\bar{s}_k\rangle}{\|\bar{s}_k\|^2} \leq 1.$$

From the estimate

$$\begin{aligned}\frac{\|(B_k - \hat{D})s_k\|^2}{\|s_k\|^2} &\leq \|\hat{D}\|^2 \frac{\|\bar{B}_k^{1/2}(\bar{B}_k^{1/2} - \bar{B}_k^{-1/2})\bar{s}_k\|^2}{\|\bar{s}_k\|^2} \\ &\leq \gamma\|\hat{D}\|^2\left(\frac{\langle \bar{s}_k, \bar{B}_k\bar{s}_k\rangle}{\|\bar{s}_k\|^2} + \frac{\langle \bar{s}_k, \bar{B}_k^{-1}\bar{s}_k\rangle}{\|\bar{s}_k\|^2} - 2\right)\end{aligned}$$

we finally get the assertion. □

The result of Corollary 3.11 is independent of the algorithmic framework in which the update is used. Together with Theorem 3.8 this indicates that also in infinite-dimensional Hilbert spaces the update property (U) is basic for the linear *and* superlinear convergence behavior of secant methods. We combine Corollary 3.11, Theorem 3.8, and Theorem 3.4 to prove that a reduced secant method is two-step superlinearly convergent, if the sequences $\{s_k\}_{k=0}^{\infty}$ and $\{v_k\}_{k=0}^{\infty}$ are chosen appropriately, and if the initial discrepancy $B_0 - \hat{H}$ is compact.

THEOREM 3.12. *Let* (A1)–(A3) *hold and assume that* $\{x_k\}_{k=0}^{\infty}$ *is generated by Algorithm* 2.2, *where* $B_0 \in \mathcal{L}(W)$ *is self-adjoint and positive. Assume further that the definition* (3.25a) *for* $\{s_k\}_{k=0}^{\infty}$ *is used in the update, and that* $\{v_k\}_{k=0}^{\infty}$ *is chosen such that* (3.24) *is satisfied in each iteration with* $s_k \neq 0$. *Then there exist positive constants* ε *and* δ *such that if* $\|x_0 - \hat{x}\| < \varepsilon$ *and* $\|B_0 - \hat{H}\| < \delta$, *the sequence* $\{x_k\}_{k=0}^{\infty}$ *converges to* $\hat{x}$. *Moreover, if* $B_0 - \hat{H} \in \mathcal{COM}(W)$, *then the convergence is two-step superlinear.*

Proof. Choose $r \in (0,1)$ and define $\varepsilon = \varepsilon(r)$ and $\delta = \delta(r)$ as in the proof of Theorem 3.8. If $\|x_0 - \hat{x}\| < \varepsilon$ and $\|B_0 - \hat{H}\| < \delta$, it follows from Theorem 3.8 that $\{x_k\}_{k=0}^{\infty}$ converges to $\hat{x}$ and, furthermore, the sequence $\{B_k\}_{k=0}^{\infty}$ belongs to $\mathcal{B}(\gamma)$ for some $\gamma > 0$.

If $s_k = 0$ for all but finitely many k, then the superlinear convergence result follows immediately from Theorem 3.4, and the compactness of $B_0 - \hat{H}$ is not needed in this case. Therefore, we can assume without loss of generality that $s_k \neq 0$ for infinitely many k, and we let $\{k_j\}_{j=0}^{\infty}$ be the subsequence of $\{k\}_{k=0}^{\infty}$ with $s_{k_j} \neq 0$ for $j \in \mathbb{N}_0$.

Recall from (3.24) that for all $j \geq 0$

$$\langle v_{k_j}, s_{k_j} \rangle > 0 \quad \text{and} \quad \|v_{k_j} - \hat{H} s_{k_j}\| \leq \beta \sigma(x_{k_j}, x_{k_j+1}) \|s_{k_j}\|.$$

This condition and the linear convergence of $\{x_k\}_{k=0}^{\infty}$ show that (U) is satisfied for the sequences $\{s_{k_j}\}_{j=0}^{\infty}, \{v_{k_j}\}_{j=0}^{\infty}$, and $\varepsilon_j := \beta\sigma(x_{k_j}, x_{k_j+1}), j \in \mathbb{N}_0$. Since the update is skipped if and only if $s_k = 0$, it follows that

$$B_{k_{j+1}} = B_{k_j+1} = B_{k_j} + V(B_{k_j}, s_{k_j}, v_{k_j}, \alpha_{k_j}) \quad \text{for all } j \geq 0.$$

Now assume in addition that $B_0 - \hat{H}$ is compact. Then $B_{k_0} - \hat{H} \in \mathcal{COM}(W)$, and we can apply Corollary 3.11 to the sequence $\{B_{k_j}\}_{j=0}^{\infty}$ to obtain

$$\lim_{j \to \infty} \frac{\|(B_{k_j} - \hat{H}) s_{k_j}\|}{\|s_{k_j}\|} = 0.$$

From $s_k = -B_k^{-1}[T(x_k)^* \nabla f(x_k) - T(\hat{x})^* \nabla f(\hat{x})], k \geq 0$, and from the definition of $\{k_j\}_{j=0}^{\infty}$ we deduce that

$$\lim_{k \to \infty} \frac{\|(B_k - \hat{H}) s_k\|}{\|e_k\|} = 0. \tag{3.29}$$

Now the assertion follows from Theorem 3.4, since the proof of Theorem 3.4 shows that the sufficient condition (3.15) can be replaced by (3.29). □

Remark. We note that the condition (3.24) can be fulfilled if the definition (3.25) is used for $\{s_k\}_{k=0}^{\infty}$ and $\{v_k\}_{k=0}^{\infty}$, provided that x_0 and B_0 are sufficiently close to

$\hat{x}$ and $\hat{H}$, respectively. This was proved in Corollary 3.9 and, as a consequence, we obtained local two-step linear convergence of the sequence $\{x_k\}_{k=0}^{\infty}$. From Theorem 3.12 we can deduce further that if $\{s_k\}_{k=0}^{\infty}$ and $\{v_k\}_{k=0}^{\infty}$ are generated by (3.25), then two-step superlinear convergence can be guaranteed for $\{x_k\}_{k=0}^{\infty}$, provided that $B_0 - \hat{H}$ is compact.

4. Application to optimal control problems. In this section we apply the convergence theory for reduced SQP methods to certain optimal control problems with ordinary differential equations that are formulated in infinite-dimensional Hilbert spaces. For this class of optimization problems we demonstrate how the structure of $\mathcal{N}(h'(x))$ can be utilized conveniently for the definition of a particular right-inverse and a null space parametrization, and we formulate the corresponding reduced secant method. Under standard smoothness assumptions for the control problem the assumption (A2) is easily verified for this choice of R and T. Furthermore, we recall a sufficient condition for (A3) which is known from optimal control theory, and we give a reasonable characterization of the compactness of the initial discrepancy $B_0 - \hat{H}$.

We consider optimal control problems of the following type:

$$\begin{aligned} &\text{minimize} && \int_0^T F(t, y(t), u(t))\, dt \\ &\text{subject to} && \dot{y}(t) = g(y(t), u(t)) \quad \text{a.e. on } [0,T], \\ &&& y(0) = y_0. \end{aligned} \tag{4.1}$$

Here $F : [0,T] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $y \in W_n^{1,2}[0,T]$, and $u \in L_m^2[0,T]$.

Suppose that $\hat{x} = (\hat{y}, \hat{u})$ is a local solution of (4.1), and let $\hat{N}$ be an open ball in $\mathbb{R}^n$ with $\hat{y}(t) \in \hat{N}$ for all $t \in [0,T]$.

We use the notation $L_m^2[0,T]$ for the space of $\mathbb{R}^m$ valued functions on $[0,T]$ having components in $L^2[0,T]$. Similarly, $W_n^{1,2}[0,T]$ is the space of $\mathbb{R}^n$ valued functions having components in $W^{1,2}[0,T]$, where $W^{1,2}[0,T]$ is the space of absolutely continuous functions on $[0,T]$ with derivatives belonging to $L^2[0,T]$. We introduce the following Hilbert spaces

$$\begin{aligned} Y &= (W_n^{1,2}[0,T], \|\cdot\|_Y), & U &= (L_m^2[0,T], \|\cdot\|_U), \\ Z &= (L_n^2[0,T] \times \mathbb{R}^n, \|\cdot\|_Z), & X &= (W_n^{1,2}[0,T] \times L_m^2[0,T], \|\cdot\|_X), \end{aligned}$$

where we define for $v \in W_n^{1,2}[0,T]$, $w \in L_m^2[0,T]$, and $q \in L_n^2[0,T]$ with component functions v_i, w_j, and q_k, respectively, and for $a \in \mathbb{R}^n$:

$$\begin{aligned} \|v\|_Y &= \Big(\textstyle\sum_{i=1}^n (\|v_i\|^2_{L^2[0,T]} + \|\dot{v}_i\|^2_{L^2[0,T]})\Big)^{\frac{1}{2}}, \\ \|w\|_U &= \Big(\textstyle\sum_{j=1}^m \|w_j\|^2_{L^2[0,T]}\Big)^{\frac{1}{2}}, \\ \|(q,a)\|_Z &= \Big(\textstyle\sum_{k=1}^n \|q_k\|^2_{L^2[0,T]} + \|a\|_2^2\Big)^{\frac{1}{2}}, \quad \text{and} \\ \|(v,w)\|_X &= (\|v\|_Y^2 + \|w\|_U^2)^{\frac{1}{2}}. \end{aligned}$$

The problem with the formulation of the optimal control problem in Hilbert spaces is that the objective and the constraint function of (4.1) are generally not Fréchet-differentiable in these spaces. It is only for this reason that we assume throughout the section that F and g in (4.1) have the following form:

$$\begin{aligned} F(t,y,u) &= F_1(y) + F_2(y)^T u + \lambda u^T C(t) u \quad \text{and} \\ g(y,u) &= g_1(y) + g_2(y) u, \end{aligned} \tag{4.2}$$

where $F_1 : \mathbb{R}^n \to \mathbb{R}, F_2 : \mathbb{R}^n \to \mathbb{R}^m, g_1 : \mathbb{R}^n \to \mathbb{R}^n, g_2 : \mathbb{R}^n \to \mathbb{R}^{n\times m}, \lambda \in \mathbb{R}\backslash\{0\}$, and $C \in L^\infty_{m\times m}[0,T]$.

Furthermore, we make the following assumption in the entire section.

(B1) F_1, F_2, g_1, g_2 and their first and second partial derivatives are Lipschitz continuous on $\hat{N}$.

Finally, the following notation will be used: If a mapping $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^t$ is differentiable in $(y,u) \in \mathbb{R}^n \times \mathbb{R}^m$, then we let $G_y(y,u) \in \mathbb{R}^{t\times n}$ and $G_u(y,u) \in \mathbb{R}^{t\times m}$ denote the Jacobians of G in (y,u) with respect to the first and second variable. In particular, note that $g_u(y,u) = g_2(y)$ because of the special structure (4.2) of g. In contrast, if P is a mapping with domain in X, then the partial Fréchet-derivatives of P in $(y,u) \in X$ will be denoted by $P'_y(y,u)$ and $P'_u(y,u)$, respectively. All vectors in $\mathbb{R}^n$ and $\mathbb{R}^m$ will be treated as column vectors, except the gradients of real-valued functions.

Since for $v \in W^{1,2}[0,T]$ the estimate $\|v\|_\infty \le c(T)\|v\|_{w^{1,2}[0,T]}$ is true for a constant $c(T)$, there is an open ball D around $\hat{x}$ such that $(y,u) \in D$ implies that $y(t) \in \hat{N}$ for all $t \in [0,T]$. We set for $x = (y,u) \in D$:

$$f(x) = \int_0^T F(t, y(t), u(t))\, dt \quad \text{and} \quad h(x) = (\dot{y} - g(y,u), y(0) - y_0).$$

Then the optimal control problem (4.1) is equivalent to the abstract optimization problem (1.1). Moreover, it follows from (B1) and from special structure (4.2) of the control problem that the assumption (A1) is satisfied in D.

If one considers the linearized state equation, one gets an idea for a suitable representation of $\mathcal{N}(h'(y,u))$ for $(y,u) \in D$. Obviously, $(\xi,\nu) \in X$ belongs to the null space of $h'(y,u)$ if and only if ξ solves the following linear initial value problem for given ν

$$\dot{\xi}(t) = g_y(y(t),u(t))\xi(t) + g_2(y(t))\nu(t) \quad \text{a.e. on } [0,T], \quad \xi(0) = 0. \tag{4.3}$$

Even in discretized form this does not suggest a parametrization of the null space through an orthonormal basis, obtained from some QR- factorization, which is frequently used in reduced SQP methods for finite-dimensional problems. Instead a very natural approach is to choose the parameter space $W = U$ and to represent all (ξ,ν) which satisfy (4.3) through ν:

$$T(y,u)\nu = (\xi,\nu),$$

where $\xi \in Y$ is the solution of (4.3). Note that this is merely a description of the operator T, and if it were only for the formulation of the reduced SQP iteration, this information about T would really be sufficient. However, in order to apply the convergence results of the previous paragraph we must study the smoothness properties of T when considered as a mapping from D into $\mathcal{L}(U,X)$. This can be done conveniently if T is represented through operators with known differentiability properties. For this purpose note that

$$h'(y,u)(\xi,\nu) = h'_y(y,u)\xi + h'_u(y,u)\nu, \quad (\xi,\nu) \in X,$$

where the partial derivatives are given by

$$h'_y(y,u)\xi = (\dot{\xi} - g_y(y,u)\xi,\ \xi(0)) \quad \text{and} \quad h'_u(y,u)\nu = -(g_2(y)\nu,\ 0).$$

It is well known (see, e.g., [23]) that for each $(q,a) \in L^2_n[0,T] \times \mathbb{R}^n$ the linear initial value problem

$$\dot{\xi}(t) = g_y(y(t),u(t))\xi(t) + q(t) \quad \text{a.e. on } [0,T], \quad \xi(0) = a$$

has a unique solution ξ in $W^{1,2}_n[0,T]$. This implies that the operator $h'_y(y,u)$ has a bounded inverse $h'_y(y,u)^{-1} \in \mathcal{L}(Z,Y)$ for each $(y,u) \in D$, which can be used to define T and R directly as

$$(4.4)\quad T(y,u) = (-h'_y(y,u)^{-1}h'_u(y,u), I_U) \quad \text{and} \quad R(y,u) = (h'_y(y,u)^{-1}, \mathcal{O}_{Z\to U}),$$

where the second component in $R(y,u)$ denotes the null operator in $\mathcal{L}(Z,U)$. It is easily checked that $R(y,u) \in \mathcal{L}(Z,X)$ is a right-inverse of $h'(y,u)$ and that $T(y,u)$ is an injective operator in $\mathcal{L}(U,X)$ satisfying $\mathcal{R}(T(y,u)) = \mathcal{N}(h'(y,u))$ for $(y,u) \in D$.

By (4.4) one can imagine that the smoothness properties of R and T are determined by those of the constraint function h. In particular, the assumption (A2) is an immediate consequence of (A1) as will be demonstrated in the proof of the next result.

LEMMA 4.1. *Assume* (B1) *and let* $T : D \to \mathcal{L}(U,X)$ *and* $R : D \to \mathcal{L}(Z,X)$ *be defined by* (4.4). *Then* T *and* R *are continuously Fréchet-differentiable in* D. *Moreover, there is a ball* $\tilde{D}$ *around* $\hat{x}$, $\tilde{D} \subset D$, *such that* $T' \in \mathrm{Lip}(\tilde{D})$ *and* $R' \in \mathrm{Lip}(\tilde{D})$.

Proof. Let $(y,u) \in D$. To simplify the notation we set

$$A(y,u) := h'_y(y,u)^{-1}.$$

It follows from (B1) that h'_y and h'_u are differentiable in D. Recall that $h''_y(y,u)$ and $h''_u(y,u)$ denote the derivatives of h'_y and h'_u in (y,u) with respect to the pair of variables. Using the familiar differentiation rules we get that A is continuously differentiable in D and

$$A'(y,u)(\xi,\nu) = -A(y,u)h''_y(y,u)(\xi,\nu)A(y,u), \quad (\xi,\nu) \in X.$$

Now one can apply the product rule to show that T and R are continuously differentiable in D, and for $(\xi,\nu) \in X$ we get

$$T'(y,u)(\xi,\nu) = (-A'(y,u)(\xi,\nu)h'_u(y,u) - A(y,u)h''_u(y,u)(\xi,\nu),\ \mathcal{O}_U),$$
$$R'(y,u)(\xi,\nu) = (A'(y,u)(\xi,\nu),\ \mathcal{O}_{Z\to U}).$$

It is sufficient to show that all the parts of R' and T' satisfy a Lipschitz condition at $\hat{x}$. From (B1) it follows immediately that h'_y, h'_u, h''_y, and h''_u belong to $\mathrm{Lip}(D)$. To achieve that also the mapping A is Lipschitz continuous at $\hat{x}$ one possibly must reduce the radius of D. Let $\tilde{M}$ denote the Lipschitz constant of h'_y:

$$\|h'_y(y,u) - h'_y(\hat{y},\hat{u})\|_{\mathcal{L}(Y,Z)} \le \tilde{M}\|(y,u) - (\hat{y},\hat{u})\|_X \quad \text{for } (y,u) \in D.$$

Define $\tilde{c} := \|A(\hat{y},\hat{u})\|^{-1}_{\mathcal{L}(Z,Y)}$, and set

$$\tilde{\rho} = \min\{\rho, \tilde{c}/(2\tilde{M})\} \quad \text{and} \quad \tilde{D} = \{(y,u) \in X : \|(y,u) - (\hat{y},\hat{u})\| < \tilde{\rho}\},$$

where ρ is the radius of D. If $(y,u) \in \tilde{D}$, then we have for $\xi \in Y$:

$$\|h'_y(y,u)\xi\| \ge \tilde{c}\|\xi\| - \tilde{M}\|(y,u) - (\hat{y},\hat{u})\|\|\xi\| \ge (\tilde{c}/2)\|\xi\|,$$

which implies that $\|A(y,u)\| \le 2/\tilde{c}$. This finally shows that

$$\|A(y,u)-A(\hat{y},\hat{u})\| \le \|A(y,u)\|\|A(\hat{y},\hat{u})\|\|h'_y(y,u)-h'_y(\hat{y},\hat{u})\| \le (2\tilde{M}/\tilde{c}^2)\|(y,u)-(\hat{y},\hat{u})\|$$

for all $(y,u) \in \tilde{D}$ and, consequently, $T', R' \in \mathrm{Lip}(\tilde{D})$. □

The last missing ingredient for the calculation of the reduced SQP step is the reduced gradient $T(y,u)^*\nabla f(y,u)$. We give the formula for $T(y_{k+1},u_{k+1})^*\nabla f(y_{k+1},u_{k+1})$, since this element will be needed later for the formulation of the algorithm. Let $\pi_{k+1} \in W_n^{1,2}[0,T]$ denote the solution of the adjoint equation

$$\begin{aligned} -\dot{\pi}(t) &= g_y^T(y_{k+1}(t),u_{k+1}(t))\pi(t) + F_y^T(t,y_{k+1}(t),u_{k+1}(t)) \quad \text{a.e. on } [0,T], \\ \pi(T) &= 0. \end{aligned} \tag{4.5}$$

Furthermore, recall that $T(y_{k+1},u_{k+1})\nu = (\xi,\nu)$ for each $\nu \in U$, where $\xi \in Y$ is the solution of a linear initial value problem depending on ν. The calculation of the reduced gradient is essentially based on these differential equations for ξ and π_{k+1}, together with a partial integration of the identity

$$0 = \xi(T)^T\pi_{k+1}(T) - \xi(0)^T\pi_{k+1}(0).$$

Then, using the integral representation of $\langle \nabla f(y_{k+1},u_{k+1}),(\xi,\nu)\rangle$ and the definition of the adjoint operator one obtains that

$$(T(y_{k+1},u_{k+1})^*\nabla f(y_{k+1},u_{k+1}))(t) = g_2^T(y_{k+1}(t))\pi_{k+1}(t) + F_u^T(t,y_{k+1}(t),u_{k+1}(t)). \tag{4.6}$$

More generally it can be shown (see, e.g., Sachs [38]) that for each $\eta \in X$ which can be represented as

$$\langle \eta,(\xi,\nu)\rangle = \int_0^T \eta_1(t)^T\xi(t)\,dt + \int_0^T \eta_2(t)^T\nu(t)\,dt \quad \text{for all } (\xi,\nu) \in X$$

with $\eta_1 \in L_n^2[0,T]$ and $\eta_2 \in L_m^2[0,T]$, the element $T(y,u)^*\eta$ is given by

$$(T(y,u)^*\eta)(t) = g_2^T(y(t))\pi(t) + \eta_2(t) \tag{4.7}$$

for $(y,u) \in D$. Here $\pi \in W_n^{1,2}[0,T]$ is the solution of

$$-\dot{\pi}(t) = g_y^T(y(t),u(t))\pi(t) + \eta_1(t) \quad \text{a.e. on } [0,T], \quad \pi(T) = 0.$$

This formula will be needed later for the calculation of the reduced Hessian $\hat{H}$.

We are now in the position to formulate a reduced SQP method for the solution of the optimal control problem (4.1), (4.2). To determine the new state y_{k+1} in a convenient way it is helpful to consider the restoration step $R(y_k,u_k)h(y_k,u_k)$, although this element is not needed explicitly for the iteration. If R is chosen according to (4.4), then

$$R(y_k,u_k)h(y_k,u_k) = (y_k - \zeta_k, 0),$$

where $\zeta_k \in Y$ solves

$$\begin{aligned} \dot{\zeta}(t) &= g_y(y_k(t),u_k(t))\zeta(t) + g(y_k(t),u_k(t)) \\ &\quad -g_y(y_k(t),u_k(t))y_k(t) \quad \text{a.e. on } [0,T], \\ \zeta(0) &= y_0. \end{aligned} \tag{4.8}$$

With the solution s_k of the equation

$$B_k s = -T(y_k, u_k)^* \nabla f(y_k, u_k)$$

the new iterate in the reduced SQP method is given by

$$(y_{k+1}, u_{k+1}) = (y_k, u_k) + T(y_k, u_k)s_k - R(y_k, u_k)h(y_k, u_k) = (\bar{y}_k + \zeta_k, u_k + s_k),$$

where $\bar{y}_k$ solves (4.3) with control input $\nu = s_k$ (and with $(y, u) = (y_k, u_k)$), and where ζ_k is the solution of (4.8). A combination of (4.3) and (4.8) shows that y_{k+1} then solves the following linear initial value problem

$$\begin{aligned} \dot{y}(t) &= g_y(y_k(t), u_k(t))y(t) + g(y_k(t), u_{k+1}(t)) \\ &\quad -g_y(y_k(t), u_k(t))y_k(t) \quad \text{a.e. on } [0, T], \\ y(0) &= y_0. \end{aligned} \tag{4.9}$$

We want to focus on a reduced secant iteration, where the sequences for the update are generated according to (3.25a), (3.25c). Again we note that this is only one possible choice for $\{s_k\}_{k=0}^{\infty}$ and $\{v_k\}_{k=0}^{\infty}$. In particular, similar techniques as for the calculation of the reduced gradient can be applied to calculate v_k from (3.25b).

The computation of the correction to B_k requires an additional effort, since the reduced gradient must be evaluated at an intermediate step. The element

$$(y_k, u_k) + T(y_k, u_k)s_k = (y_k + \bar{y}_k, u_{k+1})$$

can be obtained by computing $\bar{y}_k$ as the solution of

$$\dot{\bar{y}}(t) = g_y(y_k(t), u_k(t))\bar{y}(t) + g_2(y_k(t))s_k(t) \quad \text{a.e. on } [0, T], \quad \bar{y}(0) = 0. \tag{4.10}$$

Then we form $y_{k+} = y_k + \bar{y}_k$ to determine the solution π_{k+} of

$$\begin{aligned} -\dot{\pi}(t) &= g_y^T(y_{k+}(t), u_{k+1}(t))\pi(t) + F_y^T(t, y_{k+}(t), u_{k+1}(t)) \quad \text{a.e. on } [0, T], \\ \pi(T) &= 0. \end{aligned} \tag{4.11}$$

Finally, we compute the reduced gradient in (y_{k+}, u_{k+1}) which is needed for the update

$$(T(y_{k+}, u_{k+1})^* \nabla f(y_{k+}, u_{k+1}))(t) = g_2^T(y_{k+}(t))\pi_{k+}(t) + F_u^T(t, y_{k+}(t), u_{k+1}(t)). \tag{4.12}$$

In detail, the reduced secant method given in Algorithm 2.2 requires the following steps when applied to the optimal control problem (4.1), (4.2).

ALGORITHM 4.1 (Reduced secant method for OCP)

Given control u_k, state y_k, gradient $T(y_k, u_k)^* \nabla f(y_k, u_k)$, and $B_k \in \mathcal{L}(U), B_k$ bijective.

(1) Solve $B_k s_k = -T(y_k, u_k)^* \nabla f(y_k, u_k)$ for $s_k \in U$.

(2) Set $u_{k+1} = u_k + s_k$.

(3) Compute the solution y_{k+1} of the linear initial value problem (4.9).

(4) Determine the solution π_{k+1} of the linear final value problem (4.5).

(5) Compute the gradient $T(y_{k+1}, u_{k+1})^* \nabla f(y_{k+1}, u_{k+1})$ from (4.6).

(6) Compute the solution $\bar{y}_k$ of the linear initial value problem (4.10).

(7) Set $y_{k+} = y_k + \bar{y}_k$.

(8) Determine the solution π_{k+} of the linear final value problem (4.11).

(9) Compute $T(y_{k+}, u_{k+1})^* \nabla f(y_{k+}, u_{k+1})$ from (4.12).

(10) Set $v_k = T(y_{k+}, u_{k+1})^* \nabla f(y_{k+}, u_{k+1}) - T(y_k, u_k)^* \nabla f(y_k, u_k)$ and choose $\alpha_k \in [0, 1]$.

(11) Set $B_{k+1} = B_k + V(B_k, s_k, v_k, \alpha_k)$, if it is defined.
Else set $B_{k+1} = B_k$.

Remark. If (B1) holds and if $(y_k, u_k), (y_{k+}, u_{k+1})$, and (y_{k+1}, u_{k+1}) belong to D, then all initial and final value problems that appear in Algorithm 4.1 are uniquely solvable in $W_n^{1,2}[0, T]$ (see, e.g., [23]), and the iteration is well defined.

Remark. The solution for (4.9) and (4.10) involves the same linear differential equation with different inhomogeneous terms. The adjoint equations in (4.5) and (4.11) are not the same because the coefficients in the differential equations differ. Note that y_{k+1} can also be determined from $y_{k+1} = \bar{y}_k + \zeta_k$, after ζ_k and $\bar{y}_k$ have been computed from (4.8) and (4.10). Another alternative is to compute in a first step the solutions y_{k+1} and ζ_k of (4.9) and (4.8), and then to form $\bar{y}_k = y_{k+1} - \zeta_k$. However, the effort from the solution of the initial value problems is the same in all cases.

Remark. Each iteration in Algorithm 4.1 requires the solution of four systems of linear differential equations. We could avoid the solution of two of these systems for example by using (y_{k+1}, u_{k+1}) instead of (y_{k+}, u_{k+1}) in the definition of v_k. However, it is not clear how superlinear convergence can be retained in this case.

Remark. In Algorithm 4.1 we only have to store and update an operator $B_k \in \mathcal{L}(U)$ in contrast to full SQP methods, where the approximating operators are defined on the entire space $X = Y \times U$. In an implementation of the algorithm the reduction to the discretized control space results in a significant decrease in storage. This effect of reduction can be observed even better for boundary control problems governed by parabolic differential equations. The application of a reduced secant method to these problems leads to an iteration scheme similar to Algorithm 4.1. For an implementation and a numerical investigation of reduced secant algorithms for parabolic boundary control problems the interested reader is referred to Kupfer [31] and Kupfer and Sachs [33].

Next we turn our attention to the second order sufficiency condition (A3). We define the Hamiltonian function

$$H(t, \pi, y, u) = \pi^T g(y, u) + F(t, y, u),$$

and we let $\hat{\pi} \in W_n^{1,2}[0, T]$ denote the solution of the adjoint equation

$$-\dot{\pi}(t) = g_y^T(\hat{y}(t), \hat{u}(t))\pi(t) + F_y^T(t, \hat{y}(t), \hat{u}(t)) \quad \text{a.e. on } [0, T] \quad \text{with } \pi(T) = 0.$$

For simplicity, arguments of functions involving $\hat{y}(t), \hat{u}(t), \hat{\pi}(t)$ will be abbreviated by $[t]$; for instance $H_{yy}[t] = H_{yy}(t, \hat{\pi}(t), \hat{y}(t), \hat{u}(t))$.

It is known that under certain assumptions the strengthened Legendre condition and the global existence of a solution of the matrix Riccati equation are sufficient for (A3); see, e.g., Maurer [35]. For our Hilbert space formulation we state these sufficient conditions in the following version which, in the more natural Banach space setting

for optimal control problems, has been formulated by Kelley, Sachs, and Watson [28], for example.

(B2) The matrix function H_{uu} is positive definite,

$$\nu^T H_{uu}[t]\nu > 0 \text{ for all } \nu \in \mathbb{R}^m \setminus \{0\} \text{ and for a.a. } t \in [0,T],$$

and

$$H_{uu}[t]^{-1} \text{ is essentially bounded on } [0,T].$$

Furthermore let a solution $Q \in W^{1,2}_{n\times n}[0,T], Q$ symmetric, to the following final value problem for a matrix Riccati equation exist:

$$\dot{Q}(t) = U_{12}^T[t]Q(t) + Q(t)U_{12}[t] + Q(t)U_{11}[t]Q(t) - U_{22}[t] \quad \text{a.e. on } [0,T],$$

$$Q(T) = 0$$

where

$$\begin{aligned} U_{11}[t] &= g_2[t]H_{uu}[t]^{-1}g_2^T[t], \\ U_{12}[t] &= -g_y[t] + g_2[t]H_{uu}[t]^{-1}H_{uy}[t], \\ U_{22}[t] &= H_{yy}[t] - H_{yu}[t]H_{uu}[t]^{-1}H_{uy}[t]. \end{aligned}$$

Assuming (B1) and (B2) and following the arguments in [35] and [28] one can prove that the reduced Hessian $\hat{H}$ is positive definite which is equivalent to (A3).

We have seen in Theorem 3.12 that a compactness condition for the reduced Hessian of the Lagrangian is important for superlinear convergence. The following result corresponds to Theorem 3.1 by Kelley and Sachs [27] and yields a reasonable characterization of the compactness of $B_0 - \hat{H}$.

LEMMA 4.2. *Let* (B1) *hold. Then the operator* $E : U \to U$ *defined by*

$$(E\nu)(t) = (\hat{H}\nu)(t) - H_{uu}[t]\nu(t)$$

is compact.

Proof. Using known techniques (see, e.g., Sachs [38]) one obtains the following representation of the second derivative of the Lagrangian:

$$\langle L_x''(\hat{y},\hat{u},\hat{l})(\xi,\nu),(\bar{\xi},\bar{\nu})\rangle = \int_0^T (\bar{\xi}(t)^T, \bar{\nu}(t)^T)\begin{pmatrix} H_{yy}[t] & H_{yu}[t] \\ H_{uy}[t] & H_{uu}[t] \end{pmatrix}\begin{pmatrix} \xi(t) \\ \nu(t) \end{pmatrix} dt$$

for $(\xi,\nu),(\bar{\xi},\bar{\nu}) \in X$. From the definition of the operator T we get for $(\bar{\xi},\bar{\nu}) \in X$, $\nu \in U$,

$$\begin{aligned} \langle L_x''(\hat{y},\hat{u},\hat{l})T(\hat{y},\hat{u})\nu,(\bar{\xi},\bar{\nu})\rangle = &\int_0^T [H_{yy}[t](S_1\nu)(t) + H_{yu}[t]\nu(t)]^T\bar{\xi}(t)\,dt \\ &+ \int_0^T [H_{uy}[t](S_1\nu)(t) + H_{uu}[t]\nu(t)]^T\bar{\nu}(t)\,dt, \end{aligned}$$

and it follows from (4.7) that

$$(\hat{H}\nu)(t) = H_{uy}[t](S_1\nu)(t) + g_2^T[t](S_2\nu)(t) + H_{uu}[t]\nu(t).$$

Here $S_1\nu = \xi_1$ denotes the solution of

$$\dot{\xi}(t) = g_y[t]\xi(t) + g_2[t]\nu(t) \quad \text{a.e. on } [0,T], \quad \xi(0) = 0$$

and $S_2\nu = \xi_2$ solves

$$-\dot{\xi}(t) = g_y^T[t]\xi(t) + H_{yy}[t](S_1\nu)(t) + H_{yu}[t]\nu(t) \quad \text{a.e. on } [0,T], \quad \xi(T) = 0.$$

Note that $H_{uy}(\cdot, \hat{\pi}, \hat{y}, \hat{u})$ and $g_2^T(\hat{y})$ belong to $\mathcal{L}(L_n^2[0,T], U)$ and that $S_1, S_2 \in \mathcal{L}(U,Y)$. Then the assertion follows from the fact that the embedding $Y \longrightarrow (L_n^2[0,T], \|\cdot\|_{L_n^2[0,T]})$ is compact. □

Lemma 4.2 shows that the only noncompact part in $\hat{H}$ is H_{uu}, which must be known exactly in order to ensure superlinear convergence. It follows from the structure of the control problem (4.1), (4.2) that $H_{uu}[t] = \lambda(C(t) + C(t)^T)$ so that it is practicable to require $B_0 - H_{uu}$ to be compact. More generally, all control problems with F_{uu} known and $g_{uu} = 0$, both at the solution $\hat{u}$, are examples where this demand is reasonable.

We finish this paragraph with a convergence theorem that is an immediate consequence of Theorem 3.12 and the results of this section.

THEOREM 4.3. *Assume that* (B1) *and* (B2) *hold, and let the sequence* $\{(y_k, u_k)\}_{k=0}^{\infty}$ *be generated by Algorithm* 4.1 *with* $B_0 \in \mathcal{L}(U)$ *self-adjoint and positive. Then there exist positive constants* ε *and* δ *such that if* $\|(y_0, u_0) - (\hat{y}, \hat{u})\| < \varepsilon$ *and* $\|B_0 - \hat{H}\| < \delta$, *then* $\{(y_k, u_k)\}_{k=0}^{\infty}$ *is well defined and converges to* $(\hat{y}, \hat{u})$ *at least two-step q-linearly. Moreover, if* $B_0 - H_{uu}(\cdot, \hat{\pi}, \hat{y}, \hat{u})$ *is compact, then the convergence is two-step superlinear:*

$$\lim_{k\to\infty} \frac{\|(y_{k+1}, u_{k+1}) - (\hat{y}, \hat{u})\|}{\|(y_{k-1}, u_{k-1}) - (\hat{y}, \hat{u})\|} = 0.$$

In this section we considered optimal control problems with ordinary state equations, and we were led by the objective to verify the assumptions that are needed to ensure superlinear convergence. We made use of the structure of the control problem to develop a suitable reduced SQP method. Moreover, the iteration established in Algorithm 4.1 can be regarded as the general pattern of a reduced SQP method in the framework of separable variables. Consequently, similar iteration schemes can be derived for other infinite-dimensional problems where a separation of variables occurs naturally; as for example, in parameter identification problems (see Kunisch and Sachs [30]) and in more general optimal control problems. In our opinion the effect of reduction makes this method competitive, and it can be applied to problems where a full SQP method is not usable because of the order of the matrices that must be updated. Also the superlinear convergence of reduced secant algorithms is quite promising as could be observed numerically in several infinite-dimensional applications [30, 31, 33].

Acknowledgment. I would like to thank Ekkehard Sachs for various helpful discussions and suggestions and for his friendly and constant advice.

REFERENCES

[1] W. ALT, *The Lagrange-Newton method for infinite- dimensional optimization problems*, Numer. Funct. Anal. Optim., 11 (1990), pp. 201–224.

[2] P. T. BOGGS, J. W. TOLLE, AND P. WANG, *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.

[3] C. G. BROYDEN, J. E. DENNIS, AND J. J. MORÉ, *On the local and superlinear convergence of quasi-Newton methods*, J. Inst. Math. Appl., 12 (1973), pp. 223–245.
[4] R. H. BYRD, *An example of irregular convergence in some constrained optimization methods that use the projected Hessian*, Math. Programming, 32 (1985), pp. 232–237.
[5] ———, *On the convergence of constrained optimization methods with accurate Hessian information on a subspace*, SIAM J. Numer. Anal., 27 (1990), pp. 141–153.
[6] R. H. BYRD AND J. NOCEDAL, *A tool for the analysis of quasi- Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.
[7] ———, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1991), pp. 285–323.
[8] R. H. BYRD AND R. B. SCHNABEL, *Continuity of the null space basis and constrained optimization*, Math. Programming, 35 (1986), pp. 32–41.
[9] T. F. COLEMAN AND A. R. CONN, *On the local convergence of a quasi–Newton method for the nonlinear programming problem*, SIAM J. Numer. Anal., 21 (1984), pp. 755–769.
[10] T. F. COLEMAN AND D. C. SORENSEN, *A note on the computation of an orthonormal basis for the null space of a matrix*, Math. Programming, 29 (1984), pp. 234–242.
[11] J. E. DENNIS AND J. J. MORÉ, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.
[12] R. FONTECILLA, *Local convergence of secant methods for nonlinear constrained optimization*, SIAM J. Numer. Anal., 25 (1988), pp. 692–712.
[13] D. GABAY, *Minimizing a differentiable function over a differential manifold*, J. Optim. Theory Appl., 37 (1982), pp. 177–219.
[14] ———, *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Math. Programming Study, 16 (1982), pp. 18–44.
[15] J. C. GILBERT, *Sur quelques problèmes d'identification et d'optimisation rencontrés en physique des plasmas*, Ph.D. thesis, Université Paris VI, 1986.
[16] ———, *Une méthode de quasi-Newton réduite en optimisation sous contraintes avec priorité à la restauration*, in Proc. 7th Internat. Conf. Analysis and Optimization of Systems, A. Bensoussan and J. L. Lions, eds., Springer-Verlag, Heidelberg, 1986, pp. 40–53.
[17] ———, *On the local and global convergence of a reduced quasi-Newton method*, Optimization, 20 (1989), pp. 421–450.
[18] P. E. GILL, W. MURRAY, M. A. SAUNDERS, G. W. STEWART, AND M. H. WRIGHT, *Properties of a representation of a basis for the null space*, Math. Programming, 33 (1985), pp. 172–186.
[19] J. GOODMAN, *Newton's method for constrained optimization*, Math. Programming, 33 (1985), pp. 162–171.
[20] A. GRIEWANK, *Rates of convergence for secant methods on nonlinear problems in Hilbert space*, in Numerical Analysis, Proc. Guanajuato, Mexico, 1984, J. P. Hennart, ed., Springer-Verlag, Heidelberg, 1986, pp. 138–157.
[21] C. W. GROETSCH, *Elements of Applicable Functional Analysis*, Marcel Dekker, Inc., New York, 1980.
[22] S.-P. HAN, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11 (1976), pp. 263–282.
[23] H. HERMES AND J. P. LASALLE, *Functional Analysis and Time Optimal Control*, Academic Press, New York, 1969.
[24] W. HOYER, *Variants of the reduced Newton method for nonlinear equality constrained optimization problems*, Optimization, 17 (1986), pp. 757–774.
[25] D. M. HWANG AND C. T. KELLEY, *Sequential quadratic programming for parameter identification problems*, in 5th Symp. Control of Distributed Parameter Systems, A. E. Jai and M. Amouroux, eds., 1989, pp. 105–109.
[26] L. V. KANTOROVICH AND G. P. AKILOV, *Functional Analysis*, 2nd ed., Pergamon Press, New York, 1982.
[27] C. T. KELLEY AND E. W. SACHS, *Quasi-Newton methods and unconstrained optimal control problems*, SIAM J. Control Optim., 25 (1987), pp. 1503–1516.
[28] C. T. KELLEY, E. W. SACHS, AND B. WATSON, *Pointwise quasi-Newton method for unconstrained optimal control problems*, II, J. Optim. Theory Appl., 71 (1991), pp. 535–547.
[29] C. T. KELLEY AND S. J. WRIGHT, *Sequential quadratic programming for certain parameter identification problems*, Math. Programming, 51 (1991), pp. 281–305.
[30] K. KUNISCH AND E. W. SACHS, *Reduced* SQP *methods for parameter identification problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1793–1820.
[31] F.-S. KUPFER, *Reduced Successive Quadratic Programming in Hilbert Space with Applications to Optimal Control*, Ph.D. Thesis, Universität Trier, 1992.
[32] F.-S. KUPFER AND E. W. SACHS, *A prospective look at* SQP *methods for semilinear parabolic*

control problems, in Optimal Control of Partial Differential Equations, Irsee 1990, K.-H. Hoffmann and W. Krabs, eds., Vol. 149, Springer Lect. Notes in Control and Information Sciences, 1991, pp. 143–157.

[33] ———, *Numerical solution of a nonlinear parabolic control problem by a reduced* SQP *method*, Comput. Optim. Appl., 1 (1992), pp. 113–135.

[34] K. C. P. MACHIELSEN, *Numerical Solution of Optimal Control Problems with State Constraints by Sequential Quadratic Programming in Function Space*, CWI Tract 53, Amsterdam, 1988.

[35] H. MAURER, *First and second order sufficient optimality conditions in mathematical programming and optimal control*, Math. Programming Study, 14 (1981), pp. 163–177.

[36] J. NOCEDAL AND M. L. OVERTON, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[37] M. J. D. POWELL, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1978, pp. 27–63.

[38] E. W. SACHS, *Successive quadratic programming in function space*, Tech. report, Universität Trier, Fachbereich IV–Mathematik, 1991.

[39] J. STOER AND R. A. TAPIA, *The local convergence of sequential quadratic programming methods for solving nonlinear programs*, Tech. Report 87-4, Department of Mathematical Sciences, Rice University, Houston, TX, 1987.

[40] R. A. TAPIA, *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, J. Optim. Theory Appl., 22 (1977), pp. 135–194.

[41] ———, *Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods and a new implementation*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1978, pp. 125–164.

[42] ———, *On secant updates for use in general constrained optimization*, Math. Comp., 51 (1988), pp. 181–202.

[43] Y. YUAN, *An only 2-step q-superlinear convergence example for some algorithms that use reduced Hessian approximations*, Math. Programming, 32 (1985), pp. 224–231.

[44] J. ZHANG AND D. ZHU, *A trust region typed dogleg method for nonlinear optimization*, Optimization, 21 (1990), pp. 543–557.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 164–177, February 1996
© 1996 Society for Industrial and Applied Mathematics
009

CONVEX ANALYSIS ON THE HERMITIAN MATRICES*

A. S. LEWIS†

Abstract. There is growing interest in optimization problems with real symmetric matrices as variables. Generally the matrix functions involved are *spectral*: they depend only on the eigenvalues of the matrix. It is known that convex spectral functions can be characterized exactly as symmetric convex functions of the eigenvalues. A new approach to this characterization is given, via a simple Fenchel conjugacy formula. We then apply this formula to derive expressions for subdifferentials, and to study duality relationships for convex optimization problems with positive semidefinite matrices as variables. Analogous results hold for Hermitian matrices.

Key words. convexity, matrix function, Schur convexity, Fenchel duality, subdifferential, unitarily invariant, spectral function, semidefinite programming, quasi-Newton update

AMS subject classifications. 15A45, 49N15, 90C25, 65K10

1. Introduction.

A matrix norm on the $n \times n$ complex matrices is called *unitarily invariant* if it satisfies $\|XV\| = \|X\| = \|VX\|$ for all unitary V. A well-known result of von Neumann [30] states that if f is a symmetric gauge function on $\mathbb{R}^n$ then f induces a unitarily invariant norm, namely, $\|X\|_f = f(\sigma_1(X), \ldots, \sigma_n(X))$, where $\sigma_1(X) \leq \cdots \leq \sigma_n(X)$ are the singular values of X. Conversely, every unitarily invariant norm can be written in this form. A good exposition may be found in [14]. More generally, a matrix norm is called *weakly unitarily invariant* if it satisfies $\|V^*XV\| = \|X\|$ for all unitary V.

The $n \times n$ complex Hermitian matrices may be regarded as a *real* inner product space $\mathcal{H}$, with inner product $\langle X, Y \rangle$ defined as traceXY. Let us now ask a similar question about general convex functions on $\mathcal{H}$: What can be said about unitarily invariant convex functions $F : \mathcal{H} \to (-\infty, +\infty]$, where now by *unitarily invariant* we mean $F(V^*XV) = F(X)$ whenever V lies in $\mathcal{U}$, the $n \times n$ unitary matrices? Such functions clearly depend only on the eigenvalues of X: they are sometimes called *spectral functions* (see [10]).

Observe first that if we write diag(λ) (given λ in $\mathbb{R}^n$) for the diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$, and define a function $f : \mathbb{R}^n \to (-\infty, +\infty]$ by $f(\lambda) = F(\text{diag}(\lambda))$, then clearly f is convex and *symmetric*: $f(\lambda) = f(P\lambda)$ for all P in $\mathcal{P}$, the $n \times n$ permutation matrices. In fact the converse is also true: if $f : \mathbb{R}^n \to (-\infty, +\infty]$ is a symmetric convex function then it induces a unitarily invariant, convex matrix function $f_{\mathcal{H}} : \mathcal{H} \to (-\infty, +\infty]$, defined by

$$f_{\mathcal{H}}(X) = f(\lambda(X)), \tag{1.1}$$

where $\lambda(X) = (\lambda_1(X), \ldots, \lambda_n(X))^T$ is the vector of eigenvalues of X in nondecreasing order. This result was first proved in [4], for everywhere finite f: the proof extends immediately to allow f to take the value $+\infty$.

As an example (see §4), if we take

$$f(\lambda) = \begin{cases} -\sum_{i=1}^n \log \lambda_i, & \text{if } \lambda > 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

* Received by the editors December 21, 1993; accepted for publication (in revised form) July 6, 1994. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

† Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (aslewis@orion.uwaterloo.ca).

then we obtain the well-known convex matrix function

$$f_{\mathcal{H}}(X) = \begin{cases} -\log\det X, & \text{if } X \text{ positive definite,} \\ +\infty, & \text{otherwise.} \end{cases}$$

The approach to the basic result in [4] is direct, using a technique also appearing in [18] (strict convexity is not discussed). An independent approach appears in [10], revealing the role of "Schur convexity." Let us define a convex cone

$$K = \{x \in \mathbb{R}^n \mid x_1 \geq x_2 \geq \cdots \geq x_n\},$$

which has dual cone $K^+ = \{y \mid y^T x \geq 0 \ \forall x \in K\}$ given by

$$K^+ = \left\{ y \in \mathbb{R}^n \mid \sum_1^j y_i \geq 0 \ (j = 1, 2, \ldots, n-1), \ \sum_1^n y_i = 0 \right\}.$$

We say that a function $f : K \to (-\infty, +\infty]$ is *Schur convex* if it is K^+-isotone: in other words, $f(x) \leq f(z)$ whenever $z - x \in K^+$. Implicit in the argument in [10] (which focuses on differentiable functions f) is the fact that the spectral function $f_{\mathcal{H}}$ is convex exactly when f restricted to K is convex and Schur convex.

In fact it is not difficult to see that convex, Schur convex functions are precisely restrictions to K of symmetric convex functions. Thus $f_{\mathcal{H}}$ is convex whenever f is convex and symmetric. By contrast with our approach, lower semicontinuity of f is not required in either [4] or [10]. On the other hand, these approaches give no insight into conjugacy or subdifferentials, which of course are of fundamental interest in an optimization context.

We will follow a new approach to the basic result via Fenchel conjugation: this approach is close in spirit to von Neumann's technique in the norm case (see [14]). For a function $f : \mathbb{R}^n \to (-\infty, +\infty]$, the *Fenchel conjugate* $f^* : \mathbb{R}^n \to [-\infty, +\infty]$ is the lower semicontinuous, convex function

$$f^*(y) = \sup\{x^T y - f(x) \mid x \in \mathbb{R}^n\}.$$

(We will make frequent use of ideas and notation from [26].) By analogy, for a matrix function $F : \mathcal{H} \to (-\infty, +\infty]$ we can define a *conjugate* matrix function $F^* : \mathcal{H} \to [-\infty, +\infty]$ (cf. [8]) by

$$F^*(Y) = \sup\{\mathrm{tr} XY - F(X) \mid X \in \mathcal{H}\}. \tag{1.2}$$

Exactly as in $\mathbb{R}^n$, because F^* is expressed as a supremum of (continuous) linear functions of Y, it must be convex and lower semicontinuous.

The idea of our key result is then rather simple. We will prove (Theorem 2.3) that if the function f is symmetric on $\mathbb{R}^n$ then the conjugate of the induced matrix function $f_{\mathcal{H}}$ defined in (1.1) is given by

$$(f_{\mathcal{H}})^* = (f^*)_{\mathcal{H}}. \tag{1.3}$$

Since every lower semicontinuous, convex function g (excepting $g \equiv +\infty$) can be written as a conjugate, $g = f^*$, it follows from this formula that the matrix function it induces, $g_{\mathcal{H}}$, is a conjugate function, and hence it is lower semicontinuous and convex: in fact, to be specific, $g_{\mathcal{H}} = ((g^*)_{\mathcal{H}})^*$. An analogous argument proves the corresponding result for real-orthogonally invariant, convex functions on the $n \times n$ real

symmetric matrices. Using the conjugacy formula (1.3) it becomes straightforward to link strict convexity and differentiability properties of the underlying function f with those of the induced matrix function $f_{\mathcal{H}}$. Furthermore, (1.3) results in a simple expression for the subdifferential of $f_{\mathcal{H}}$ in terms of the subdifferential of f.

It is possible to follow an analogous route to the study of the real vector space of $m \times n$ complex matrices (with inner product $\langle X, Y \rangle = \mathrm{Re}(\mathrm{tr} XY^*)$) and (strongly) unitarily invariant functions F on this space (meaning $F(X) = F(UXV)$ for any unitary U and V). By analogy with (1.1), such functions have the form $F(X) = f(\sigma(X))$, where $\sigma_i(X)$ is the ith singular value of X (for $i = 1, 2, \ldots, q = \min\{m, n\}$), arranged in nondecreasing order, and f is symmetric and *absolute* ($f(\sigma) = f(|\sigma_1|, |\sigma_2|, \ldots, |\sigma_q|)$ for any σ in $\mathbb{R}^q$). In a fashion very similar to our present development we arrive at the analogue of formula (1.3) and hence expressions for subdifferentials. Details are deferred to a forthcoming note: we simply observe that such expressions have been the topic of a number of recent papers in the special case where f is a symmetric gauge function, and hence F is a unitarily invariant norm (see [36, 31, 32, 37, 5]). This approach will also yield characterizations of strict convexity and smoothness in this setting analogous to those in our present development: such results for unitarily invariant norms have appeared in [3, 36].

Studying convex matrix functions via their Fenchel conjugates is not a new idea. It is implicit for example in some of the techniques in [7], and was used explicitly in [8] to study the sum of the largest k eigenvalues of a real symmetric matrix, an approach also followed in [12] (see also [13]). The primary aim of these latter papers is to study sensitivity results via the subdifferential set. Various representations of this set were investigated in [22]–[24].

We present a number of well-known convex matrix functions, showing how their (strict) convexity follows easily. To conclude, we use the conjugacy formula to study duality relationships for various convex optimization problems posed over the cone of positive semidefinite, real symmetric matrices. Interest in matrix optimization problems (and duality in particular) has been growing in recent years (for instance [27, 23, 1, 33, 35, 28]). The examples we choose are of recent interest in applications of interior point methods (see for example [1, 15, 21, 2]), as well as for variational characterizations of certain quasi-Newton updates (see for example [9, 34]).

2. Conjugates of induced matrix functions. We begin with a technical lemma (cf. [11, Theorem 368]).

LEMMA 2.1. *Suppose that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$ and $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$ are real numbers and that P is an $n \times n$ permutation matrix. Then $\alpha^T P \beta \leq \alpha^T \beta$, with equality if and only if there exists an $n \times n$ permutation matrix Q with $Q\alpha = \alpha$ and $QP\beta = \beta$.*

Proof. Consider permuting the components of $P\beta$ in the following fashion.

Phase 1. Whenever we find indices i and j with $\alpha_i < \alpha_j$ and $(P\beta)_i > (P\beta)_j$, swap $(P\beta)_i$ and $(P\beta)_j$, giving a new sum $\alpha^T P' \beta > \alpha^T P \beta$ (because $(\alpha_i - \alpha_j)((P\beta)_i - (P\beta)_j) < 0$). We repeat this procedure until it terminates, say with the sum $\alpha^T P'' \beta$. Notice that the sum $\alpha^T P' \beta$ increases strictly at each step, and can take only finitely many values.

Phase 2. Now partition $\{1, 2, \ldots, n\}$ into sets $I_1 \leq I_2 \leq \cdots \leq I_k$ so that $\alpha_i = \gamma_r$ for all i in I_r, where γ_r increases strictly with r. Finally choose a permutation with matrix Q, fixing each index set I_r, and permuting the components $\{(P''\beta)_i \mid i \in I_r\}$ into nondecreasing order for each r.

Now note that $Q\alpha = \alpha$, while since $(P''\beta)_i \leq (P''\beta)_j$ whenever $\alpha_i < \alpha_j$ we deduce that $QP''\beta = \beta$. Notice that $\alpha^T P''\beta = (Q\alpha)^T(QP''\beta) = \alpha^T\beta$. Hence we see that $\alpha^T P\beta \leq \alpha^T\beta$. If equality holds then Phase 1 must be vacuous, and hence $P = P''$. The converse is immediate. □

The basis of the following key result is fairly standard and due to von Neumann [30] (see, for example, [19, p. 248] and the discussion in [5]). The full result (including conditions for attainment) may be found in [29] via an algebraic approach. In keeping with the variational spirit of this paper, and for completeness, we present here an optimization-based proof, following ideas from [25]. The underlying variational problem originated once again with von Neumann (see [20]).

THEOREM 2.2. *For Hermitian matrices X and Y,*

$$\mathrm{tr}XY \leq \lambda(X)^T\lambda(Y), \tag{2.1}$$

with equality if and only if there exists a unitary matrix V with $V^*XV = \mathrm{diag}\lambda(X)$ and $V^*YV = \mathrm{diag}\lambda(Y)$.

Proof. Consider the optimization problem

$$\left\{\begin{array}{ll} \text{maximize} & \mathrm{tr}Z^*XZY \\ \text{subject to} & Z^*Z = I, \\ & Z \in \mathbb{C}^{n\times n}. \end{array}\right. \tag{2.2}$$

This problem is solvable by compactness. We can regard the constraint as a linear map between two real vector spaces, $\Phi : \mathbb{C}^{n\times n} \to \mathcal{H}$, with a nonsingular derivative at any feasible point. Thus corresponding to any optimal solution Z_0 there will exist a Lagrange multiplier Λ in $\mathcal{H}$, so that

$$\nabla_Z(\mathrm{tr}Z^*XZY - \mathrm{tr}Z^*Z\Lambda)|_{Z_0} = 0.$$

Thus for all W in $\mathbb{C}^{n\times n}$,

$$\begin{aligned} 0 &= \lim_{t\to 0} t^{-1}(\mathrm{tr}(Z_0+tW)^*(X(Z_0+tW)Y - (Z_0+tW)\Lambda) \\ &\quad - Z_0^*(XZ_0Y - Z_0\Lambda)) \\ &= \mathrm{tr}Z_0^*XWY + \mathrm{tr}W^*XZ_0Y - \mathrm{tr}Z_0^*W\Lambda - \mathrm{tr}W^*Z_0\Lambda \\ &= \mathrm{tr}(YZ_0^*X - \Lambda Z_0^*)W + \mathrm{tr}W^*(XZ_0Y - Z_0\Lambda). \end{aligned}$$

Choosing $W = XZ_0Y - Z_0\Lambda$ shows that $XZ_0Y = Z_0\Lambda$, and hence

$$Z_0^*XZ_0Y = \Lambda = \Lambda^* = YZ_0^*XZ_0.$$

Thus Y commutes with $Z_0^*XZ_0$, so there is a unitary matrix U diagonalizing Y and $Z_0^*XZ_0$. In other words

$$\begin{aligned} U^*YU &= \mathrm{diag}(P_1\lambda(Y)) \quad \text{and} \\ U^*Z_0^*XZ_0U &= \mathrm{diag}(P_2\lambda(X)) \end{aligned} \tag{2.3}$$

for some permutation matrices P_1 and P_2. Now we have

$$\begin{aligned} \mathrm{tr}XY &\leq \mathrm{tr}Z_0^*XZ_0Y = \mathrm{tr}(U^*Z_0^*XZ_0U)(U^*YU) \\ &= (P_1\lambda(Y))^T(P_2\lambda(X)) = \lambda(Y)^T(P_1^*P_2)\lambda(X) \\ &\leq \lambda(Y)^T\lambda(X) \end{aligned}$$

by Lemma 2.1.

If equality holds in (2.1) then $Z_0 = I$ is optimal for (2.2), and equality holds above. Again by Lemma 2.1 there is a permutation matrix Q with $Q\lambda(Y) = \lambda(Y)$ and $QP_1^*P_2\lambda(X) = \lambda(X)$. From (2.3) we know that $P_1^*U^*YUP_1 = \mathrm{diag}\lambda(Y)$, so

$$QP_1^*U^*YUP_1Q^* = \mathrm{diag}(Q\lambda(Y)) = \mathrm{diag}\lambda(Y).$$

Also from (2.3) we have

$$QP_1^*U^*XUP_1Q^* = \mathrm{diag}(QP_1^*P_2\lambda(X)) = \mathrm{diag}\lambda(X),$$

and the result follows if we choose $V = UP_1Q^*$. □

We can now prove the main result.

THEOREM 2.3. *Suppose that the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is symmetric. Then $(f_{\mathcal{H}})^* = (f^*)_{\mathcal{H}}$.*

Proof. For a Hermitian matrix Y we have

$$\begin{aligned}(f_{\mathcal{H}})^*(Y) &= \sup\{\mathrm{tr}XY - f(\lambda(X)) \mid X \in \mathcal{H}\} \\ &= \sup\{\mathrm{tr}XY - f(P\lambda(X)) \mid X \in \mathcal{H},\ P \in \mathcal{P}\} \\ &= \sup\{\mathrm{tr}XY - f(\lambda) \mid X \in \mathcal{H},\ P \in \mathcal{P},\ \lambda \in \mathbb{R}^n,\ P\lambda(X) = \lambda\}.\end{aligned}$$

Now considering the supremum over λ first, we can rewrite this as

$$\begin{aligned}(f_{\mathcal{H}})^*(Y) &= \sup_{\lambda \in \mathbb{R}^n} \{\sup\{\mathrm{tr}XY \mid X \in \mathcal{H},\ P \in \mathcal{P},\ P\lambda(X) = \lambda\} - f(\lambda)\} \\ &= \sup_{\lambda \in \mathbb{R}^n} \{\sup\{\lambda(Y)^T(Q\lambda) \mid Q \in \mathcal{P}\} - f(\lambda)\} \\ &= \sup\{\lambda(Y)^T(Q\lambda) - f(Q\lambda) \mid \lambda \in \mathbb{R}^n,\ Q \in \mathcal{P}\} \\ &= f^*(\lambda(Y)) = (f^*)_{\mathcal{H}}(Y),\end{aligned}$$

where we used Lemma 2.1 and Theorem 2.2 in the first step to see that the inner suprema are equal. □

COROLLARY 2.4. *If the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is symmetric, convex, and lower semicontinuous then the matrix function $f_{\mathcal{H}} : \mathcal{H} \to (-\infty, +\infty]$ is convex and lower semicontinuous.*

Proof. We can assume that f is somewhere finite. Then since f^* is nowhere $-\infty$, with $f^{**} = f$ [26, Theorem 12.2], and since f^* is symmetric [26, Corollary 12.3.1], we have $f_{\mathcal{H}} = ((f^*)^*)_{\mathcal{H}} = ((f^*)_{\mathcal{H}})^*$. Thus $f_{\mathcal{H}}$ is a conjugate function, so is convex and lower semicontinuous. □

Exactly analogous results for functions on the real symmetric matrices may be derived by replacing unitary by real orthogonal matrices throughout.

3. Subgradients, differentiability, and strict convexity. Suppose that $\mathcal{X}$ is a finite-dimensional, real inner product space. The *conjugate* of a function $F : \mathcal{X} \to [-\infty, +\infty]$ is the function $F^* : \mathcal{X} \to [-\infty, +\infty]$ defined by

$$F^*(Y) = \sup_{X \in \mathcal{X}} \{\langle X, Y\rangle - F(X)\}.$$

Since $\mathcal{X}$ is isomorphic to $\mathbb{R}^n$ with its usual inner product, convex-analytic results on $\mathbb{R}^n$ can be translated directly.

The *domain* of F is the set $\mathrm{dom}F = \{X \in \mathcal{X} \mid F(X) < +\infty\}$. If this set is nonempty and F never takes the value $-\infty$, then we say that F is *proper*. By [26,

Theorem 12.2], if F is proper and convex, then F^* is proper, convex, and lower semicontinuous. For proper F with X in $\mathrm{dom}F$, we can define the *(convex) subdifferential* of F at X as the convex set

$$\partial F(X) = \{Y \in \mathcal{X} \mid F(X) + F^*(Y) = \langle X, Y \rangle\}, \tag{3.1}$$

and when F is also convex this set is a singleton $\{Y\}$ exactly when F is differentiable at X, with gradient $\nabla F(X) = Y$ [26, Theorem 25.1].

We say that the proper, convex function F is *essentially smooth* if it is differentiable on the interior of $\mathrm{dom}F$ (assumed nonempty), with $\|\nabla F(X^r)\| \to +\infty$ whenever X^r approaches a boundary point of $\mathrm{dom}F$. We say that F is *essentially strictly convex* if F is strictly convex on any convex subset of $\{X \in \mathrm{dom}F \mid \partial F(X) \neq \emptyset\}$ (and hence in particular on the interior of $\mathrm{dom}F$) [26, Chapter 26]. A lower semicontinuous, proper, convex function F satisfies $F = F^{**}$ [26, Theorem 12.2], and F^* is essentially strictly convex if and only if F is essentially smooth [26, Theorem 26.3]: this is the case exactly when $\partial F(X)$ is single-valued when nonempty [26, Theorem 26.1].

THEOREM 3.1. *Suppose that the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is symmetric. Then $Y \in \partial f_{\mathcal{H}}(X)$ if and only if $\lambda(Y) \in \partial f(\lambda(X))$ and there exists a unitary matrix V with $V^*XV = \mathrm{diag}\lambda(X)$ and $V^*YV = \mathrm{diag}\lambda(Y)$.*

Proof. For Hermitian matrices X and Y, Y lies in $\partial f_{\mathcal{H}}(X)$ exactly when

$$\mathrm{tr}XY = f_{\mathcal{H}}(X) + (f_{\mathcal{H}})^*(Y) = f(\lambda(X)) + f^*(\lambda(Y)) \geq \lambda(X)^T\lambda(Y) \geq \mathrm{tr}XY,$$

by Theorem 2.3, and the result follows by Theorem 2.2. □

COROLLARY 3.2. *Suppose that the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is symmetric, convex, and lower semicontinuous. Then the function $f_{\mathcal{H}} : \mathcal{H} \to (-\infty, +\infty]$ is essentially smooth if and only if f is essentially smooth. In this case, for any Hermitian X in $\mathrm{int}(\mathrm{dom}f_{\mathcal{H}})$ we have that*

$$\nabla f_{\mathcal{H}}(X) = V\mathrm{diag}(\nabla f(\lambda(X)))V^* \tag{3.2}$$

for any unitary V satisfying $V^*XV = \mathrm{diag}\lambda(X)$.

Proof. Suppose that f is essentially smooth (the converse is straightforward by restricting to diagonal matrices). If $\partial f_{\mathcal{H}}(X)$ is nonempty then by Theorem 3.1 it is exactly

$$\{V\mathrm{diag}(\nabla f(\lambda(X)))V^* \mid V^*XV = \mathrm{diag}\lambda(X),\ V \in \mathcal{U}\}.$$

Therefore every element of the convex set $\partial f_{\mathcal{H}}(X)$ has the same Frobenius norm, $\|\nabla f(\lambda(X))\|_2$, and hence this set is a singleton (because the Frobenius norm is strict). Thus $f_{\mathcal{H}}$ is essentially smooth. □

Some additional comments are warranted in regard to this corollary. Notice that the proof above actually shows that if the function f is symmetric, lower semicontinuous, and convex then the function $f_{\mathcal{H}}$ is differentiable at X whenever f is differentiable at $\lambda(X)$, with gradient given by (3.2). Furthermore, using Davis's result [4] in place of Corollary 2.4 allows us to dispense with the assumption of lower semicontinuity in this observation. In fact a completely different approach [17] shows that convexity is not needed for the gradient formula (3.2): this paper also derives a result analogous to Theorem 3.1 for the Clarke generalized derivative.

Taking conjugates gives the following result.

COROLLARY 3.3. *Suppose that the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is symmetric, convex and lower semicontinuous. Then the function $f_{\mathcal{H}} : \mathcal{H} \to (-\infty, +\infty]$ is essentially strictly convex if and only if f is essentially strictly convex.*

Again, exactly parallel arguments show the corresponding results for real symmetric matrices.

4. Examples. In this section we will see that many of the classically known convex functions on the Hermitian matrices can be derived from our main result. Any symmetric convex function is Schur-convex [19, Proposition 3.C.2], and not surprisingly, many of the standard Schur-convex functions are symmetric, convex, and lower semicontinuous. We will simply illustrate a variety of examples.

The simplest class of examples are functions of $\lambda \in \mathbb{R}^n$ of the form

$$\sum_{i=1}^{n} g(\lambda_i) \text{ for } g : \mathbb{R} \to (-\infty, +\infty] \text{ convex, lower semicontinuous.} \tag{4.1}$$

In particular, in (4.1) we could take

$$g(\mu) = \begin{cases} 0, & \text{if } \mu \geq 0, \\ +\infty, & \text{if } \mu < 0, \end{cases} \tag{4.2}$$

$$g(\mu) = \begin{cases} 1/\mu, & \text{if } \mu > 0, \\ +\infty, & \text{if } \mu \leq 0, \end{cases} \tag{4.3}$$

$$g(\mu) = \begin{cases} -\log \mu, & \text{if } \mu > 0, \\ +\infty, & \text{if } \mu \leq 0. \end{cases} \tag{4.4}$$

More generally, we could consider

$$\sum_{P \in \mathcal{P}} h(P\lambda) \text{ for } h : \mathbb{R}^n \to (-\infty, +\infty] \text{ convex, lower semicontinuous.} \tag{4.5}$$

This will encompass such functions as $\sum_i |\lambda_i - \bar{\lambda}|$ (where $\bar{\lambda} = \sum_i \lambda_i / n$) and $\sum_{i,j} |\lambda_i - \lambda_j|$ for example.

For any symmetric set $C \subset \mathbb{R}^n$ (in other words, closed under coordinate permutations) the support function $\sup\{\phi^T \lambda \mid \phi \in C\}$ will be convex, lower semicontinuous and symmetric. In this way we obtain the examples (for $k = 1, 2, \ldots, n$)

$$\text{sum of the } k \text{ largest elements of } \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \tag{4.6}$$

(by taking $C = \{\phi \mid 0 \leq \phi_i \leq 1 \ (i = 1, 2, \ldots, n),\ \sum \phi_i = k\}$), and similarly

$$-\text{ sum of the } k \text{ smallest elements of } \{\lambda_1, \lambda_2, \ldots, \lambda_n\}. \tag{4.7}$$

In particular, any symmetric gauge function will be symmetric, convex, and continuous (see [14, p. 438]). Examples are $\|\lambda\|_p$ for $1 \leq p \leq +\infty$, and

$$\text{sum of the } k \text{ largest elements of } \{|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|\}. \tag{4.8}$$

For $k = 1, 2, \ldots, n$, the elementary symmetric function $S_k(\lambda)$ and the complete symmetric function $C_k(\lambda)$ have the property that $-(S_k(\lambda))^{1/k}$ and $(C_k(\lambda))^{1/k}$ are both symmetric, convex, and continuous on the nonnegative orthant $\mathbb{R}^n_+$ [19, 3.F.2 and 3.F.5]. A particular example is

$$\begin{cases} -(\lambda_1 \lambda_2 \ldots \lambda_n)^{1/n}, & \text{if } \lambda \geq 0, \\ +\infty, & \text{otherwise} \end{cases} \tag{4.9}$$

Furthermore, for any strictly positive real a, the function

$$\begin{cases} S_k(\lambda_1^{-a}, \lambda_2^{-a}, \ldots, \lambda_n^{-a}), & \text{if } \lambda > 0, \\ +\infty, & \text{otherwise} \end{cases} \tag{4.10}$$

is symmetric, convex, and lower semicontinuous [19, 3.G.1.m]. Somewhat analogously to (4.5), we could consider

$$\max_{P \in \mathcal{P}} h(P\lambda) \text{ for } h : \mathbb{R}^n \to (-\infty, +\infty] \text{ convex, lower semicontinuous.}$$

Examples are (4.6),(4.7) and (on the domain $\mathbb{R}^n_+$)

$$-(\text{product of the } k \text{ largest elements of } \{\lambda_1, \lambda_2, \ldots, \lambda_n\})^{1/k}. \tag{4.11}$$

For $n \times n$ Hermitian matrices X and Y we write $X \preceq Y$ if $Y - X$ is positive semidefinite, and $X \prec Y$ if $Y - X$ is positive definite. We denote the identity matrix by I. Now by Corollary 2.4, each of the examples above induces a lower semicontinuous convex function on the Hermitian matrices, and Theorem 2.3 gives a formula for the conjugate. Thus example (4.2) induces the indicator function of the cone of positive semidefinite matrices $\{X \succeq 0\}$, which is thus a closed, convex cone, and computing the conjugate shows that this cone is self-dual (Fejer's Theorem; see [14]):

$$\operatorname{tr} XY \geq 0 \text{ for all } X \succeq 0 \quad \Leftrightarrow \quad Y \succeq 0.$$

The functions (4.3), (4.4), (4.6), and (4.7) (whose conjugates may be computed directly) induce, respectively, the lower semicontinuous, convex, matrix functions

$$\begin{cases} \operatorname{tr} X^{-1}, & \text{if } X \succ 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

$$\begin{cases} -\log\det X, & \text{if } X \succ 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

$$\sum_{i=n-k+1}^{n} \lambda_i(X) \text{ and} \tag{4.12}$$

$$-\sum_{i=1}^{k} \lambda_i(X),$$

and by applying Theorem 2.3 we see that the corresponding conjugate functions are

$$\begin{cases} -2\operatorname{tr}(-Y)^{1/2}, & \text{if } Y \prec 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

$$\begin{cases} -n - \log\det(-Y), & \text{if } Y \prec 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

$$\begin{cases} 0, & \text{if } 0 \preceq Y \preceq I, \text{ with } \operatorname{tr} Y = k, \\ +\infty, & \text{otherwise, and} \end{cases} \tag{4.13}$$

$$\begin{cases} 0, & \text{if } -I \preceq Y \preceq 0, \text{ with } \mathrm{tr}Y = n-k, \\ +\infty, & \text{otherwise.} \end{cases}$$

The function $\|\lambda\|_p$ induces the Schatten p-norm, special cases being the trace norm ($p = 1$), the Frobenius norm ($p = 2$), and the spectral norm ($p = \infty$). The function (4.8) induces the Ky Fan k-norm $\sum_{i=n-k+1}^{n} |\lambda_i(X)|$. The functions (4.9), (4.10), and (4.11) induce the matrix functions

$$\begin{cases} -(\det X)^{1/n}, & \text{if } X \succeq 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

$$\begin{cases} S_k(\lambda_1(X)^{-a}, \lambda_2(X)^{-a}, \ldots, \lambda_n(X)^{-a}), & \text{if } X \succ 0, \\ +\infty, & \text{otherwise, and} \end{cases}$$

$$\begin{cases} -(\prod_{i=n-k+1}^{n} \lambda_i(X))^{1/k}, & \text{if } X \succeq 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

All of these examples may be found in [19, 16.F] for example. Many are easily seen to be strictly convex with the help of Corollary 3.3.

As a final example, suppose that the set $C \subset \mathbb{R}^n$ is closed, convex, and symmetric. By applying Corollary 2.4 with f the indicator function of C we see immediately that the set of Hermitian matrices X with $\lambda(X) \in C$ is a closed, convex set (cf. [16]).

Theorem 3.1 can be used to calculate subdifferentials. Consider for example the sum of the k largest eigenvalues of X (example (4.12)). The problem of deriving expressions for the subdifferential of this function is considered via the computation of the conjugate function (4.13) in [8, 12, 24]. If the function $f(\lambda)$ is given by (4.6) then at any point λ in $\mathbb{R}^n$ satisfing $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ it is a straightforward calculation to check that $\mu \in \partial f(\lambda)$ if and only if

$$\mu_i \begin{cases} = & 0 & \text{if } \lambda_i < \lambda_{n-k+1}, \\ \in & [0,1] & \text{if } \lambda_i = \lambda_{n-k+1}, \\ = & 1 & \text{if } \lambda_i > \lambda_{n-k+1}, \end{cases}$$

with $\sum \mu_i = k$. Using Theorem 3.1 we see that the subdifferential of X is exactly the set of matrices $V\mathrm{diag}(\mu)V^*$ with unitary V satisfying $V^*XV = \mathrm{diag}\lambda(X)$ and μ in $\mathbb{R}^n$ satisfying

$$\mu_i \begin{cases} = & 0 & \text{if } \lambda_i(X) < \lambda_{n-k+1}(X), \\ \in & [0,1] & \text{if } \lambda_i(X) = \lambda_{n-k+1}(X), \\ = & 1 & \text{if } \lambda_i(X) > \lambda_{n-k+1}(X), \end{cases}$$

and $\sum \mu_i = k$. In particular, for example, for the maximum eigenvalue function $\lambda_n(X)$ (which is the case $k = 1$) we obtain the well-known result

$$\partial\lambda_n(X) = \mathrm{conv}\{vv^* \mid \|v\| = 1,\ Xv = \lambda_n(X)v\}.$$

A similar expression can be obtained for the subdifferential of the Ky Fan k-norm.

5. Fenchel duality, semidefinite programming, and quasi-Newton updates. In this section we illustrate how the conjugacy formula derived in §2 can be used to study duality properties of optimization problems involving real symmetric matrices. In particular we can study analogues of linear programming over the cone of positive semidefinite matrices (see [27, 23, 1, 33, 35, 2]), penalized versions of such problems (see, for example, [1, 15, 21, 2]), and convex optimization problems leading to well-known quasi-Newton formulae for minimization algorithms [9, 34].

Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are finite-dimensional inner-product spaces. For functions $F : \mathcal{X} \to (-\infty, +\infty]$ and $G : \mathcal{Y} \to (-\infty, +\infty]$, and a linear map $A : \mathcal{X} \to \mathcal{Y}$, consider the optimization problem

$$\alpha = \inf_{X \in \mathcal{X}} \{F(X) + G(AX)\}. \tag{5.1}$$

If we define the *adjoint map* $A^T : \mathcal{Y} \to \mathcal{X}$ by

$$\langle AX, Y \rangle = \langle X, A^T Y \rangle \quad \text{for all } X \in \mathcal{X}, Y \in \mathcal{Y}, \tag{5.2}$$

then we can associate with the *primal* problem (5.1) a *dual* problem

$$\beta = \sup_{Y \in \mathcal{Y}} \{-F^*(A^T Y) - G^*(-Y)\}. \tag{5.3}$$

The *weak duality inequality* $\alpha \geq \beta$ is trivial to check. Fenchel duality results give conditions ensuring that $\alpha = \beta$. We consider one such particular result. We say that the function G is *polyhedral* if its *epigraph* $\operatorname{epi} G = \{(Y, r) \in \mathcal{Y} \times \mathbb{R} \mid r \geq G(Y)\}$ is a polyhedron. We denote the interior of a convex set $C \subset \mathcal{X}$ with respect to its affine span by $\operatorname{ri} C$. The various parts of the following result (stated for $\mathcal{X} = \mathbb{R}^n$) may be found in [26].

THEOREM 5.1. *Suppose in problem* (5.1) *that the functions F and G are convex, with G polyhedral. Then providing that there exists an X in* $\operatorname{ri}(\operatorname{dom} F)$ *with AX in* $\operatorname{dom} G$, *the primal and dual values* (5.1) *and* (5.3) *are equal, and the dual value is attained when finite. In this case, X_0 and Y_0 are optimal for the primal and dual problems respectively if and only if $-Y_0 \in \partial G(AX_0)$ and $A^T Y_0 \in \partial F(X_0)$. In particular, if F is lower semicontinuous and F^* is differentiable at $A^T Y_0$ then the unique primal optimal solution is $X_0 = \nabla F^*(A^T Y_0)$.*

As an example, consider the semidefinite programming problem (cf. [21]):

$$\begin{cases} \inf & \operatorname{tr} EX \\ \text{subject to} & X \in B + \mathcal{L}, \\ & 0 \preceq X \in \mathcal{S}, \end{cases} \tag{5.4}$$

where $\mathcal{S}$ denotes the $n \times n$ real symmetric matrices, E and B are given symmetric matrices, and $\mathcal{L}$ is a given subspace of $\mathcal{S}$. Observe that for any function $H : \mathcal{X} \to (-\infty, +\infty]$,

$$(H + \langle E, \cdot \rangle)^*(Y) = H^*(Y - E). \tag{5.5}$$

If we choose spaces $\mathcal{X} = \mathcal{Y} = \mathcal{S}$, the map A to be the identity,

$$F(X) = \begin{cases} \operatorname{tr} EX, & \text{if } X \succeq 0, \\ +\infty, & \text{otherwise, and} \end{cases}$$

$$G(X) = \begin{cases} 0, & \text{if } X \in B + \mathcal{L}, \\ +\infty, & \text{otherwise,} \end{cases}$$

then it is easy to calculate directly (or using (4.2) with (5.5)) that

$$F^*(Y) = \begin{cases} 0, & \text{if } Y \preceq E, \\ +\infty, & \text{otherwise, and} \end{cases}$$

$$G^*(Y) = \begin{cases} \mathrm{tr}BY, & \text{if } Y \in \mathcal{L}^\perp, \\ +\infty, & \text{otherwise,} \end{cases}$$

where the orthogonal complement $\mathcal{L}^\perp$ is the subspace of symmetric matrices Y satisfying $\mathrm{tr}XY = 0$ whenever $X \in \mathcal{L}$. Hence the dual problem is (cf. [27])

$$\begin{cases} \sup & \mathrm{tr}BY \\ \text{subject to} & Y \in \mathcal{L}^\perp, \\ & E \succeq Y \in \mathcal{S}. \end{cases} \tag{5.6}$$

We can emphasize the symmetry with the primal problem (5.4) by setting $Z = E - Y$, if so desired. Now Theorem 5.1 shows that providing there exists a positive definite X in $B + \mathcal{L}$, the primal and dual values are equal, with attainment in the dual if it is feasible. In this case complementary slackness results are also straightforward to derive: feasible X_0 and Y_0 are respectively primal and dual optimal if and only if $\mathrm{tr}X_0(E - Y_0) = 0$.

A related problem, arising for example when we replace the constraint $X \succeq 0$ in problem (5.4) by adding a penalty function to the objective function, is

$$\begin{cases} \inf & \mathrm{tr}EX + \delta f(\lambda(X)) \\ \text{subject to} & X \in B + \mathcal{L}, \end{cases} \tag{5.7}$$

where $\delta > 0$ is a small penalty parameter and the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is lower semicontinuous and convex with $\mathrm{cl}(\mathrm{dom} f) = \mathbb{R}^n_+$. An example is (4.4), giving the "logarithmic barrier" penalized problem

$$\begin{cases} \inf & \mathrm{tr}EX - \delta \log\det X \\ \text{subject to} & X \in B + \mathcal{L}, \\ & 0 \prec X \in \mathcal{S}. \end{cases} \tag{5.8}$$

The dual problem for (5.7), using the real symmetric version of Theorem 2.3, is

$$\begin{cases} \inf & \mathrm{tr}BY - \delta f^*(\delta^{-1}\lambda(Y - E)) \\ \text{subject to} & Y \in \mathcal{L}^\perp. \end{cases} \tag{5.9}$$

Again, providing there is a positive definite matrix X in $B + \mathcal{L}$, Theorem 5.1 shows that the primal and dual values are equal, with attainment in (5.9) when it is feasible. For the primal problem (5.8) we obtain the dual problem

$$\begin{cases} \inf & \mathrm{tr}BY - \delta \log\det(E - Y) + \delta n(\log\delta - 1) \\ \text{subject to} & Y \in \mathcal{L}^\perp, \\ & E \succ Y, \end{cases}$$

which is just the logarithmic barrier penalized version of the original dual problem (5.6). Semidefinite programming problems involving other objective functions can also be studied using these techniques. The maximum eigenvalue is an example [22].

To conclude, we consider problems of the form

$$\left\{\begin{array}{ll} \inf & \mathrm{tr}EX + f(\lambda(X)) \\ \text{subject to} & Xs = y, \\ & X \in \mathcal{S}, \end{array}\right. \tag{5.10}$$

where again the function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is lower semicontinuous and convex with $\mathrm{cl}(\mathrm{dom} f) = \mathbb{R}^n_+$, and the vectors s and y in $\mathbb{R}^n$ are given. Such problems arise in the context of characterizing quasi-Newton Hessian updates satisfying the "secant equation" $Xs = y$ (see, for example, [9]). The given real symmetric matrix E is derived from the old Hessian approximation. Once again a good example comes from (4.4), which gives the problem

$$\left\{\begin{array}{ll} \inf & \mathrm{tr}EX - \log\det X \\ \text{subject to} & Xs = y, \\ & 0 \prec X \in \mathcal{S}. \end{array}\right. \tag{5.11}$$

The adjoint of the linear map $A : \mathcal{S} \to \mathbb{R}^n$ defined by $AX = Xs$ is easily computed to be given by $A^T z = (zs^T + sz^T)/2$ for z in $\mathbb{R}^n$. If we choose $F : \mathcal{S} \to (-\infty, +\infty]$ to be given by $F(X) = \mathrm{tr}EX + f(\lambda(X))$ and $G : \mathbb{R}^n \to (-\infty, +\infty]$ defined by $G(w) = 0$ if $w = y$ and $+\infty$ otherwise, then applying Theorem 2.3 gives the dual problem

$$\sup_{z \in \mathbb{R}^n} \{y^T z - f^*(\lambda(-E + (zs^T + sz^T)/2))\}. \tag{5.12}$$

If $s^T y > 0$ then it is well known that there exists a positive definite matrix X satisfying the secant equation $Xs = y$ (for this and other standard theory of quasi-Newton updates, see [6]). Hence Theorem 5.1 applies to show that the primal and dual values are equal, and that (5.12) is attained when finite. Furthermore, if z_0 solves (5.12) and we denote the matrix $-E + (z_0 s^T + s z_0^T)/2$ by E_0, and if f^* is differentiable at $\lambda(E_0)$, then by the comment after Corollary 3.2 the unique optimal solution of (5.10) is $X_0 = V \nabla f^*(\lambda(E_0)) V^*$ for any unitary matrix V satisfying $V^* E_0 V = \mathrm{diag}\lambda(E_0)$.

In particular, the dual problem for (5.11) becomes

$$\sup_{z \in \mathbb{R}^n} \{y^T z + \log\det(\lambda(E - (zs^T + sz^T)/2))\} + n. \tag{5.13}$$

This is straightforward to solve explicitly, using the fact that $\nabla \log\det X = X^{-1}$, and assuming that E is positive definite (which ensures that (5.13) has the feasible solution $z = 0$). The resulting optimal solution X_0 of (5.11) is the "BFGS update" of E^{-1} (see [6, p. 205] and [9]).

Acknowledgments. The author wishes to thank O. Güler, M. Teboulle, and H. Wolkowicz for drawing his attention to [4, 18, 10], and an anonymous referee for many helpful suggestions, including a number of references to literature on matrix norms.

REFERENCES

[1] F. Alizadeh, *Optimization over the positive definite cone: interior point methods and combinatorial applications*, in Advances in Optimization and Parallel Computing, P. Pardolos, ed., North-Holland, Amsterdam, 1992, pp. 1–25.

[2] ———, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[3] J. ARAZY, *On the geometry of the unit ball of unitary matrix spaces*, Integral Equations Operator Theory, 4 (1981), pp. 151–171.
[4] C. DAVIS, *All convex invariant functions of hermitian matrices*, Archiv der Mathematik, 8 (1957), pp. 276–278.
[5] E.M. DE SÁ, *Exposed faces and duality for symmetric and unitarily invariant norms*, Linear Algebra Appl., 197,198 (1994), pp. 429–450.
[6] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[7] P. A. FILLMORE AND J. P. WILLIAMS, *Some convexity theorems for matrices*, Glasgow Math. J., 12 (1971), pp. 110–117.
[8] R. FLETCHER, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.
[9] ———, *A new variational result for quasi-Newton formulae*, SIAM J. Optim., 1 (1991), pp. 18–21.
[10] S. FRIEDLAND, *Convex spectral functions*, Linear Multilinear Algebra, 9 (1981), pp. 299–316.
[11] G. H. HARDY, J. E. LITTLEWOOD, AND G. PÓLYA, *Inequalities*, Cambridge University Press, Cambridge, UK, 1952.
[12] J.-B. HIRIART-URRUTY, A. SEEGER, AND D. YE, *Sensitivity analysis for a class of convex functions defined over a space of symmetric matrices*, Vol. 382, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York, 1992, pp. 133–154.
[13] J.-B. HIRIART-URRUTY AND D. YE, *Sensitivity Analysis of all Eigenvalues of a Symmetric Matrix*, Tech. report, Laboratoire d'analyse numerique, Université Paul Sabatier, Toulouse, France, 1992.
[14] R. A. HORN AND C. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[15] F. JARRE, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.
[16] F. JOHN, *On symmetric matrices whose eigenvalues satisfy linear inequalities*, Proc. American Mathematical Society, 17 (1966), pp. 1140–1146.
[17] A. S. LEWIS, *Derivatives of Spectral Functions*, Tech. Report CORR 94-04, University of Waterloo, 1994; Math. Oper. Res., to appear.
[18] M. MARCUS, *Convex functions of quadratic forms*, Duke Math. J., 24 (1957), pp. 321–326.
[19] A. W. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, 1979.
[20] L. MIRSKY, *On the trace of matrix products*, Math. Nach., 20 (1959), pp. 171–174.
[21] Y. E. NESTEROV AND A. S. NEMIROVSKY, *Interior Point Polynomial Methods in Convex Programming*, Society for Industrial and Applied Mathematics, Philadelphia, 1994.
[22] M.L. OVERTON, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.
[23] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.
[24] M.L. OVERTON AND R.S. WOMERSLEY, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Math. Programming, Series B, 62 (1993), pp. 321–357.
[25] F. RENDL AND H. WOLKOWICZ, *Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem*, Math. Programming, 53 (1992), pp. 63–78.
[26] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[27] A. SHAPIRO, *Extremal problems on the set of nonnegative definite matrices*, Linear Algebra Appl., 67 (1985), pp. 7–18.
[28] A. SHAPIRO AND M.K.H. FAN, *On eigenvalue optimization*, SIAM J. Optim., 5 (1995), pp. 552–569.
[29] C.M. THEOBALD, *An inequality for the trace of the product of two symmetric matrices*, Math. Proc. Cambridge Philosophical Soc., 77 (1975), p. 265.
[30] J. VON NEUMANN, *Some matrix inequalities and metrization of matric-space*, Tomsk University Rev., 1 (1937), pp. 286–300. In Collected Works, Vol. IV, Pergamon Press, Oxford, 1962, pp. 205–218.
[31] G. A. WATSON, *Characterization of the subdifferential of some matrix norms*, Linear Algebra Appl., 170 (1992), pp. 33–45.
[32] ———, *On matrix approximation problems with Ky Fan k norms*, Numer. Algorithms, 5 (1993), pp. 263–272.
[33] H. WOLKOWICZ, *Explicit Solutions for Interval Semidefinite Linear Programs*, Tech. Report CORR 93-29, Department of Combinatorics and Optimization, University of Waterloo, 1993.

[34] H. WOLKOWICZ AND Q. ZHAO, *An Efficient Region of Optimal Updates for Least Change Secant Methods*, Tech. Report CORR 92-27, Department of Combinatorics and Optimization, University of Waterloo, 1992.

[35] B. YANG AND R.J. VANDERBEI, *The Simplest Semidefinite Programs are Trivial*, Tech. report, Program in Statistics and Operations Research, Princeton University, Princeton, NJ, 1993.

[36] K. ZIĘTAK, *On the characterization of the extremal points of the unit sphere of matrices*, Linear Algebra Appl., 106 (1988), pp. 57–75.

[37] ———, *Subdifferentials, faces and dual matrices*, Linear Algebra Appl., 185 (1993), pp. 125–141.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 178–197, February 1996
© 1996 Society for Industrial and Applied Mathematics
010

CLASSICAL OPTIMALITY CONDITIONS UNDER WEAKER ASSUMPTIONS*

SIMON DI†

Dedicated to Nanjing University.

Abstract. In this article, an optimization problem that incorporates equality constraints, inequality constraints, and an abstract constraint is considered. We prove the classical optimality conditions (first-order and second-order, necessary, and sufficient) under weaker assumptions: functions involved are continuous around and differentiable at the optimal point instead of continuously differentiable or strictly differentiable and/or twice differentiable. The proof is based on the exact expressions for the contingent cone and the second-order contingent hull to the feasible set established in this article. It is interesting to note that our second-order Lagrange multiplier rule for minimization problems incorporating a more generalized abstract constraint has an improved appearance. Direct applications to "max-type" minimization problems and Pareto optimality are mentioned.

Key words. optimality conditions, constraint qualification, the contingent cone, the second-order contingent hull, differentiably regular set

AMS subject classifications. 49J52, 49K10, 90C30, 90C25

1. Introduction.

One of the most fundamental problems in optimization theory is to find *optimality conditions* for various kinds of minimization problems, e.g., to find necessary conditions and sufficiency conditions for a local optimal solution to the following problems: $\mathcal{P}$,

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } h_i(x) = 0 \quad i = 1, 2, \dots, m, \\ &\qquad h_j(x) \geq 0 \quad j = m+1, \dots, s, \\ &\qquad x \in D, \end{aligned} \tag{1.1}$$

where f, h_i $(i = 1, \dots, s)$ are the objective function and the constraint functions from a Hilbert space $\mathbf{X}$ to $\mathbb{R}$, D is the abstract constraint set in $\mathbf{X}$, which is closed, and m and s are given integers such that $m \leq s$. In smooth optimization, all functions are assumed to be continuously differentiable, and the abstract constraint set is assumed to be a polyhedron. The optimality conditions usually include first-order and second-order necessary conditions and first-order and second-order sufficiency conditions. In nonsmooth optimization, all functions could be Lipschitzian, convex, or lower semicontinuous functions; see Clarke [4], Rockafellar [18], and Ward and Borwein [21]. The optimality conditions for a nonsmooth situation are weaker in the sense that they appear in the form of set inclusions involving sets of various kinds of generalized differentials. An interesting question can be asked: Are the same classical optimality conditions (in the form of equations instead of set inclusions) true for nonsmooth functions?

* Received by the editors July 28, 1993; accepted for publication (in revised form) August 19, 1994.

† Department of Mathematical Sciences, Lakehead University, 955 Oliver Road, Thunder Bay, Ontario P7B 5E1, Canada (simon.di@lakeheadu.ca). This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

In this article, we prove the classical optimality conditions for (1.1) under weaker assumptions on the functions involved. Instead of continuous differentiable for first-order classical optimality conditions, we assume that the objective function and the constraint functions f, h_i $(i = 1, \ldots, s)$ are continuous around and differentiable at some local optimum point x_0. For second-order necessary and sufficient conditions, the functions are assumed to be continuous around x_0 and admit first-order Fréchet derivative at x_0 and second-order Fréchet derivative at x_0 (without assuming any kind of differentiability around x_0). Along with the weaker conditions on the functions, the abstract constraint set D is generalized: D can be a closed convex set or any set determined by continuous constraint functions which are also differentiable at the local optimum. Closed convex sets, polyhedra, and sets determined by smooth constraint functions are examples of the abstract constraint sets considered in this article.

In §2, we review and specify some notations and definitions, including the definition of the contingent cone $K_B(x_0)$, the second-order contingent hull $K_B^2(x_0, v)$, and the strict normal cone $\widehat{N}_B(x_0)$. It is important to review the definitions of first-order and second-order Fréchet derivatives at a point, because they are the basic conditions assumed on functions in our problems. Also in §2, we prove an alternative theorem, which is different from the traditional one. A generalized intermediate value theorem and a fundamental theorem on a system of linear equations are reviewed in the section for the purpose of later use.

In §3, we provide explicit expressions for the contingent cone and the second-order contingent hull to the set C determined by equality and inequality constraints in (1.1), which are the main approaches to prove our optimality conditions. The expressions are not new, but the conditions on functions h_i are weaker. For the expression of the contingent cone (Theorem 3.3), we assume that h_i are continuous around x_0 and Fréchet differentiable at x_0, instead of strictly differentiable at x_0 or continuously differentiable around x_0. For the expression of the second-order contingent hull (Theorem 3.6), we assume that h_i are continuous around x_0, Fréchet differentiable at x_0, and second-order Fréchet differentiable at x_0, which does not mean strictly differentiable at x_0 or continuously differentiable around x_0.

In §4, the same kind of results in §3 are expanded for the feasible set of the problem (1.1) assuming D is closed and convex or differentiably regular; i.e., the exact expressions are established for the contingent cone and the second-order contingent hull to the set determined by equality constraints, inequality constraint, and an abstract constraint with the abstract set being closed and convex or differentiably regular. Differential regularity is a new notion used to describe a set property. The expansions are significant in considering convex nonsmooth problems and accommodating more general types of practical problems.

In §5, optimality conditions are established, and the optimality conditions are expressed in the form of Lagrange multiplier rule. The results are proved based on the expressions of the contingent cone and the second-order contingent hull to the feasible set established in §§3 and 4. Our optimality conditions are significant in the following meanings: (i) they assume weaker conditions on all functions; (ii) they accommodate a more general type of minimization problem; (iii) they are in terms of the contingent cone and the second-order contingent hull, which are more conclusive about the necessity and sufficiency of the local optimum; (iv) the results have many practical applications. We expect that we can explain some numerical algorithms under the new assumptions. For an example, according to Theorems 5.1 and 5.2, we do not need differentiability anywhere other than the optimal point in optimality condi-

tions; therefore we do not have to say (and it is awkward to do so at nondifferentiable points x_i) that the finite difference $\big(f(x_i + t_i v) - f(x_i)\big)/t_i$ is an approximation to the gradient of f at x_i during iterations. It is also expected to develop new numerical algorithms to solve nonsmooth optimization problems more efficiently based on the results developed in this section. According to our results or our analytic methods, the optimality conditions for many other nonsmooth minimization problems such as the "max-type" problems and the discrete minimax problems can be updated.

2. Notations and tools. With respect to problem $\mathcal{P}$ defined by (1.1), we denote by $I_1 = \{1, 2, \ldots, m\}$, $I_2 = \{m+1, m+2, \ldots, s\}$, and

$$
\begin{aligned}
C_i &:= \{ x \in \mathbf{X} \mid h_i(x) = 0 \} \quad \text{for} \quad i \in I_1, \\
C_i &:= \{ x \in \mathbf{X} \mid h_i(x) \geq 0 \} \quad \text{for} \quad i \in I_2, \\
C &:= \bigcap_{i=1}^{s} C_i = \left\{ x \in \mathbf{X} \mid \begin{array}{ll} h_i(x) = 0 & \text{for} \quad i \in I_1 \\ h_i(x) \geq 0 & \text{for} \quad i \in I_2 \end{array} \right\}.
\end{aligned} \tag{2.1}
$$

Let $A = C \cap D$, then A is the feasible set of $\mathcal{P}$. Given any closed convex cone S contained in a Hilbert space $\mathbf{X}$, we denote by $int\, S$ the interior set of S and $S^* = \{ b \in \mathbf{X} \mid \langle b, s \rangle \geq 0 \text{ for all } s \in S \}$ the polar cone of S. Given a set B, and a point $x_0 \in B$, one way to define the *contingent cone* (or *Bouligand tangent cone*) is

$$
K_B(x_0) := \left\{ \lim_{i \to \infty} \frac{x_i - x_0}{t_i} \mid \{x_i\} \subset B,\ x_i \to x_0, t_i \downarrow 0 \right\}. \tag{2.2}
$$

For any $v \in K_B(x_0)$, we define the *second-order contingent hull* to B at x_0 for v to be

$$
K_B^2(x_0, v) := \left\{ \lim_{i \to \infty} \frac{x_i - x_0 - t_i v}{t_i^2/2} \mid \{x_i\} \subset B, x_i \to x_0, t_i \downarrow 0 \right\}. \tag{2.3}
$$

Another equivalent way to define the contingent cone and the second-order contingent hull is

$$
K_B(x_0) := \limsup_{t \downarrow 0} \frac{B - x_0}{t}, \qquad K_B^2(x_0, v) := \limsup_{t \downarrow 0} \frac{B - x_0 - tv}{t^2/2}.
$$

We should mention at this point that there is another commonly used notion of "tangency" called *Ursescu* tangency. The *Ursescu tangent cone* and the *second-order Ursescu tangent set* can be defined in the following way (see Cominetti [5]):

$$
\overline{K}_B(x_0) := \liminf_{t \downarrow 0} \frac{B - x_0}{t}, \qquad \overline{K}_B^2(x_0, v) := \liminf_{t \downarrow 0} \frac{B - x_0 - tv}{t^2/2}.
$$

The concept of the *second-order contingent hull* is very useful in dealing with second-order optimality conditions that we develop later on. It is easy to prove that $K_B^2(x_0, v)$ is closed, and furthermore, we have the following results.

LEMMA 2.1. *Given a closed convex set B, assume $x_0 \in B$ and $v \in K_B(x_0)$, then*

(i) $K_B^2(x_0, v) + cl\ cone(B - x_0) \subset K_B^2(x_0, v)$, *where cl cone$(B - x_0)$ denotes the smallest closed cone containing $B - x_0$;*

(ii) $K_B^2(x_0, v) - \alpha v \subset K_B^2(x_0, v)$ *for any* $\alpha \in \mathbb{R}$;

(iii) $\alpha K_B^2(x_0, v) + \beta K_B^2(x_0, v) \subset K_B^2(x_0, v)$ *for any* $\alpha \geq 1$ *and* $\beta \geq 0$.

Proof. (i) Given $y \in B$, $u \in K_B^2(x_0, v)$, and $\alpha \geq 0$, there exist $\{x_i\} \subset B$ and $t_i \downarrow 0$ such that $(x_i - x_0 - t_i v)/\frac{1}{2} t_i^2 \to u$. Denote by $y_i := x_i + \frac{1}{2} t_i^2 \alpha (y - x_i)$, then by the

convexity of B, we have $y_i \in B$ for i large enough, and

$$\frac{y_i - x_0 - t_i v}{t_i^2/2} = \frac{x_i - x_0 - t_i v + t_i^2 \alpha(y - x_i)/2}{t_i^2/2}$$
$$= \frac{x_i - x_0 - t_i v}{t_i^2/2} + \alpha(y - x_i) \longrightarrow u + \alpha(y - x_0),$$

i.e., $u + \alpha(y - x_0) \in K_B^2(x_0, v)$, which finishes the proof by the closedness of $K_B^2(x_0, v)$.

(ii) Given $u \in K_B^2(x_0, v)$ and $\alpha \in \mathbb{R}$, there exist $\{x_i\} \subset B$ and $t_i \downarrow 0$ such that $(x_i - x_0 - t_i v)/\frac{1}{2}t_i^2 \to u$. Therefore,

$$\frac{x_i - x_0 - (t_i + \alpha t_i^2/2)v}{(t_i + \alpha t_i^2/2)^2/2} = \frac{x_i - x_0 - t_i v}{(t_i + \alpha t_i^2/2)^2/2} - \frac{(\alpha t_i^2/2)v}{(t_i + \alpha t_i^2/2)^2/2} \longrightarrow u - \alpha v,$$

i.e., $u - \alpha v \in K_B^2(x_0, v)$.

(iii) Let $u \in K_B^2(x_0, v)$, then there exist $\{x_i\} \subset B$ and $t_i \downarrow 0$ such that $(x_i - x_0 - t_i v)/\frac{1}{2}t_i^2 \to u$. Given any number $\alpha \geq 1$, denote $t = 1/\alpha$, $y_i := x_0 + t(x_i - x_0)$, then by the convexity of B, $y_i \in B$, and

$$\frac{y_i - x_0 - t t_i v}{(t t_i)^2/2} = \frac{t(x_i - x_0 - t_i v)}{(t t_i)^2/2} \to \frac{1}{t} u = \alpha u,$$

which proves that $\alpha u \in K_B^2(x_0, v)$. By (i) and (ii), $\alpha u + \beta(z_i - x_0 - s_i v)/(s_i^2/2) \in K_B^2(x_0, v)$ for $\beta \geq 0$, $z_i \in B$, and $s_i > 0$, which proves (iii). □

For a closed convex set B, the second-order Ursescu tangent cone $\overline{K}_B^2(x_0, v)$ is convex, but we do not know if the second-order contingent hull $K_B^2(x_0, v)$ is convex or not. It is also desirable to find a simple expression for $K_B^2(x_0, v)$ like the one for the contingent cone $K_B(x_0) = cl\ cone(B - x_0)$. However, We will see later in §3 that $K_B^2(x_0, v)$ is convex and has a nice expression if B is determined by a number of equality and/or inequality constraints.

A vector $v \in \mathbf{X}$ is a *strict normal* vector, as it is often called, to a closed set B at $x_0 \in B$, written as $v \in \widehat{N}_B(x_0)$ if $\langle v, y - x_0 \rangle \leq 0$ for all $y \to x_0$ and $y \in B$, i.e.,

$$\limsup_{\substack{y \to x_0 \\ y \neq x_0, y \in B}} \langle v, y - x_0 \rangle \leq 0. \tag{2.4}$$

The strict normal cone is often used in discussing the properties of a constraint set, and it is easy to see that the strict normal cone is the negative polar to the contingent cone, i.e., $\widehat{N}_B(x_0) = -K_B^*(x_0)$.

Given g, a function from a Hilbert space $\mathbf{X}$ to $\mathbb{R}$, we say that g admits *(first-order) Fréchet derivative* at x_0 if there exists an element $\nabla g(x_0) \in \mathbf{X}$, such that

$$\lim_{t \downarrow 0} \frac{g(x_0 + tv) - g(x_0)}{t} = \langle \nabla g(x_0), v \rangle, \tag{2.5}$$

and the limit is uniform with respect to v in bounded sets. First-order Fréchet derivative $\nabla g(x_0)$ can be considered by Riesz's theorem as a linear operator from $\mathbf{X}$ to $\mathbb{R}$. We say that g admits *second-order Fréchet derivative* at x_0 if there exists an element $\nabla^2 g(x_0)$ in $\mathcal{L}(\mathbf{X}, \mathbf{X})$, the space of all continuous linear transformations from $\mathbf{X}$ to $\mathbf{X}$, such that

$$\lim_{t \downarrow 0} \frac{g(x_0 + tv) - g(x_0) - t\langle \nabla g(x_0), v \rangle}{t^2/2} = \langle v, \nabla^2 g(x_0) v \rangle, \tag{2.6}$$

and the limit is uniform with respect to v in bounded sets. Second-order Fréchet differentiability at a point is not a local property, and it does not imply that the function is strictly differentiable at the point, or continuously differentiable around the point, or Lipschitz around the point. It is important to clarify this, otherwise, one might mix up this kind of second-order derivative with twice differentiability at x_0 in Cominetti [5], where first-order differentiability is assumed around x_0. For the rest of this paper, differentiability, if not specified, means Fréchet differentiability.

The alternative theorem has been used to prove the optimality conditions for various kinds of optimization problems (see Craven [7]). The following alternative theorem, which is used in the proof of our main results, is different from the one in Craven [7].

LEMMA 2.2 (alternative theorem). *Let* $\mathbf{X}$ *be a Hilbert space, and* $\mathcal{A} \in \mathcal{L}(\mathbb{R}^m, \mathbf{X})$. *Suppose* Γ *and* S *are any two closed convex cones in* $\mathbb{R}^m$ *such that* $-S^* \subset \Gamma$ *and* $\operatorname{int} S \neq \emptyset$. *Then exactly one of the following systems has a solution:*

(i). $-\mathcal{A}^T\mathcal{A}x \in \operatorname{int} S$ *for* $x \in \Gamma$ *such that* $|x| \leq 1$;

(ii). $\mathcal{A}p = 0 \in \mathbf{X}$ *for* $0 \neq p \in S^*$.

Proof. First, (i) and (ii) cannot have solutions simultaneously. Otherwise, there exist $x \in \Gamma$ such that $p' := -\mathcal{A}^T\mathcal{A}x \in \operatorname{int} S$ and nonzero $p \in S^*$ such that $\mathcal{A}p = 0$. Because $p \in S^*$ and $p' \in \operatorname{int} S$, then $pp' \neq 0$, i.e. $\langle -p, \mathcal{A}^T\mathcal{A}x\rangle \neq 0$, which contradicts $\mathcal{A}p = 0$.

Second, if (i) admits no solutions, then

$$E := \{\, \mathcal{A}^T\mathcal{A}x + s \mid x \in \Gamma,\ |x| \leq 1,\ s \in \operatorname{int} S\}$$

is a convex, open subset of $\mathbb{R}^m$ not containing 0. By the separation theorem, there exists $p \in \mathbb{R}^m$ such that $p(E) \subset \mathbb{R}^+ := \{\, \alpha \in \mathbb{R} \mid \alpha \geq 0\,\}$. Because $\operatorname{int} S \subset E$, $p \in S^*$. Now we conclude that $\mathcal{A}p = 0$. Otherwise, fix any $s \in \operatorname{int} S$, there exists $\alpha > 0$ such that $-\alpha p \in \Gamma$ and $|-\alpha p| \leq 1$ (because $-S^* \subset \Gamma$), and for any positive integer n, $a_n := -\mathcal{A}^T\mathcal{A}\alpha p + \frac{1}{n}s \in E$, but $\langle p, a_n\rangle = -\alpha\langle p, \mathcal{A}^T\mathcal{A}p\rangle + \frac{1}{n}\langle p, s\rangle < 0$ when n is large enough, which contradicts the choice of p. □

The following generalized intermediate value theorem plays an essential role in the proof of the results in the next section. The theorem uses the notion of m-dimensional regular box. An m-dimensional regular box in a finite dimensional space $\mathbb{R}^m$ is a nonempty closed polyhedron with 2^m vertices and $2m$ faces such that it is linearly homomorphic to the standard box $[0,1]^m$. Like a line or a square in three-dimensional space, we can similarly define m-dimensional regular boxes in any finite-dimensional space $\mathbb{R}^n$ with $n \geq m$.

LEMMA 2.3 (generalized intermediate value theorem). *Let* $g = (g_1, \ldots, g_m)$ *be functions defined on an open set containing an* m*-dimensional regular box* $\mathcal{B}$ *in* $\mathbb{R}^n$. *Denote its* i*th pair of opposite sides by* Π_i' *and* Π_i''. *If for each* $i = 1, 2, \ldots, m$, $g_i(x) > a_i$ *for* $x \in \Pi_i'$ *and* $g_i(x) < a_i$ *for* $x \in \Pi_i''$, *then there exists* $\xi \in \operatorname{int} \mathcal{B}$ *such that* $g(\xi) = a := (a_1, a_2, \ldots, a_m)$.

Proof. The result follows from the antipodal theorem of Borsuk and Krasnoselski; see Corollary 16.7 in [22]. □

The *fundamental theorem on systems of linear equations* can be found in subjects like functional analysis and optimization theory, and therefore we state it without providing a proof.

LEMMA 2.4. *Let* x, a_i $(i = 0, 1, \ldots, s) \in \mathbf{X}$, *be a Hilbert space, and* b_i $(i = 1, \ldots, s) \in \mathbb{R}$. *Then for an inequality* $\langle a_0, x\rangle \leq b_0$ *to be a consequence of a consistent*

system

$$\langle a_i, x\rangle = b_i \quad \text{for} \quad i \in I_1, \qquad \langle a_i, x\rangle \le b_i \quad \text{for} \quad i \in I_2, \tag{2.7}$$

it is necessary and sufficient to have the following condition: There exists $\lambda = (\lambda_1, \dots, \lambda_s)$ *with* $\lambda_i \ge 0$ *for* $i \in I_2$ *such that*

$$a_0 = \sum_{i=1}^{s} \lambda_i a_i \quad \text{and} \quad \sum_{i=1}^{s} \lambda_i b_i \le b_0.$$

If not specified, functions $h_i : \mathbf{X} \to \mathbb{R}$ are assumed to be continuous around x_0 and admit first-order Fréchet derivative at x_0 where x_0 is a given feasible point throughout the rest of the article.

3. Expressions for contingent cone (hull). In this section, we establish the exact expressions of the contingent cone and the second-order contingent hull to C, defined by (2.1), at some point x_0 of C. All inequality constraints can be divided into two types: active and inactive. Let us denote by I_2' the index set (subset of I_2) of active inequality constraints at x_0, i.e., those $i \in I_2$ such that $h_i(x_0) = 0$, and by I_2'' the index set (subset of I_2) of inactive inequality constraints at x_0, i.e., those $i \in I_2$ such that $h_i(x_0) > 0$. We define

$$\Lambda := \{\, \lambda \in \mathbb{R}^s \mid \lambda_i \ge 0 \quad \text{for } i \in I_2' \;\; \lambda_i = 0 \quad \text{for } i \in I_2'' \,\}; \tag{3.1}$$

then

$$\Lambda^* = \{\, \lambda \in \mathbb{R}^s \mid \lambda_i = 0 \quad \text{for } i \in I_1 \quad \text{and } \lambda_i \ge 0 \quad \text{for } i \in I_2' \,\}. \tag{3.2}$$

Before proving our main results, we need the following lemma.

LEMMA 3.1. *Let* $v_1, v_2, \dots, v_s$ *be elements in a Hilbert space* $\mathbf{X}$. *If there is no nonzero* $\lambda \in \Lambda$ *such that*

$$\sum_{i=1}^{s} \lambda_i v_i = 0, \tag{3.3}$$

then there exists an m*-dimensional regular box* $\mathcal{B}$ *in* $\mathbb{R}^s$, *a linearly bijective image of* $[0,1]^m$, *with the* j*th pair of opposite sides denoted by* Π_j^1, *and* Π_j^2, *such that for* $j \in I_1$, $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle > 0$ *for* $\alpha \in \Pi_j^1$ *and* $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle < 0$ *for* $\alpha \in \Pi_j^2$, *and for* $j \in I'_2$, $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle > 0$ *for* α *in all sides of* $\mathcal{B}$, *where* $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_s) \in \mathbb{R}^s$.

Proof. There are 2^m cones like

$$\{\, \lambda \in \mathbb{R}^s \mid (\lambda_i \ge 0 \quad \text{or} \quad \lambda_i \le 0 \quad \text{for} \quad i \in I_1) \quad \text{and} \quad (\lambda_i \ge 0 \quad \text{for} \quad i \in I'_2) \,\},$$

denoted by $S_1, S_2, \dots, S_{2^m}$, respectively. Then S_i^* is a subset of Λ. Define $\Gamma_i = -S_i^*$, and let $\mathcal{A}$ be a linear operator from $\mathbb{R}^s$ to $\mathbf{X}$ defined by $\mathcal{A}\alpha = \sum_{i=1}^s \alpha_i v_i$; then the linear operator $\mathcal{A}$ is continuous (bounded). Now (3.3) guarantees that the conditions of Lemma 2.2 are satisfied and there is no solution to (ii) for each $i = 1, 2, \dots, 2^m$. Therefore there exists a solution to (i) in Γ_i , say x_i, such that $|x_i| \le 1$ and $-\mathcal{A}^T\mathcal{A}x_i \in \operatorname{int} S_i$ for $i = 1, \dots, 2^m$. These 2^m $-x_i$ are in different "quadrants" of an m-dimensional subspace of $\mathbb{R}^s$, and their convex combination is an m-dimensional regular box $\mathcal{B}$ containing the origin. If we denote the set of these 2^m $-x_i$ by V, then V can be divided into two subsets V_i^1 and V_i^2 with equal number (2^{m-1}) of points in each subset, say $V_i^1 = \{\, -x_{i_k} \mid k = 1, 2, \dots, 2^{m-1} \,\}$ and $V_i^2 = \{\, -x_{i_k} \mid k = 2^{m-1} + 1, 2^{m-1} + 2, \dots, 2^m \,\}$ such that the convex combinations of the points in V_i^1 and the convex combinations of the points in V_i^2 formulate a pair of opposite sides (say Π_i^1 and Π_i^2) of the regular

box, then there are m different ways to make such a division (corresponding to m pairs of opposite sides of the box). Because $\Gamma_i = -S_i^*$ and $-\mathcal{A}^T\mathcal{A}x_i \in \operatorname{int} S_i$, for each such a pair of opposite sides Π_j^1 and Π_j^2, there exists one and only one $j \in I_1$ such that the jth components of $\operatorname{int} S_k$ are all positive for $k = 1, 2, \ldots, 2^{m-1}$ and the jth components of $\operatorname{int} S_k$ are all negative for $k = 2^{m-1}+1, 2^{m-1}+2, \ldots, 2^m$, that is $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle > 0$ for $\alpha \in V_j^1$, $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle < 0$ for $\alpha \in V_j^2$, and simultaneously, for all $i \in I_2'$, $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle > 0$ for $\alpha \in V$, because $-\mathcal{A}^T\mathcal{A}x_i \in \operatorname{int} S_i$ and the ith components of the elements in $\operatorname{int} S$ are all positive. Then the result follows from the linear property of the function $\langle v_j, \sum_{i=1}^s \alpha_i v_i\rangle$ (with respect to variable α) and the fact that a side (say Π_j^1) is the convex linear combination of all its vertices (points in V_j^1). □

Before proving the exact expression of the contingent cone to a set determined by a number of equalities and/or inequalities, we state a special case (see Flett [10]) for a set determined by a single equality constraint or a single inequality constraint.

PROPOSITION 3.2 (case of one constraint). *Let $\mathbf{X}$ be a Hilbert space. Suppose $h : \mathbf{X} \to \mathbb{R}$ is continuous around x_0 and differentiable at x_0. Assume $h(x_0) = 0$ and $\nabla h(x_0) \neq 0$. Then for $C = \{ y : h(y) = 0 \}$, we have*

$$K_C(x_0) = \{ v \in \mathbf{X} : \langle \nabla h(x_0), v\rangle = 0 \}. \tag{3.4}$$

For $C = \{ y : h(y) \geq 0 \}$, we have

$$K_C(x_0) = \{ v \in \mathbf{X} : \langle \nabla h(x_0), v\rangle \geq 0 \}. \tag{3.5}$$

The condition $\nabla h(x_0) \neq 0$ for the above result is essential. For the set C defined by (2.1), i.e., the situation that C is determined by a number of equality and/or inequality constraints, the corresponding condition is equivalent to a constraint qualification in discussing optimality conditions. The following is the *Mangasarian–Fromovitz constraint qualification.*

B_1. BASIC CONSTRAINT QUALIFICATION. *Let $h = (h_1, h_2, \ldots, h_s)$ be continuous around x_0 and Fréchet differentiable at x_0. The only $\lambda \in \Lambda$, defined by* (3.1), *such that $\sum_{i=1}^s \lambda_i \nabla h_i(x_0) = 0$ is $\lambda = 0$.*

THEOREM 3.3 (expression of the contingent cone). *Let $x_0 \in C$, defined by* (2.1). *Under the basic constraint qualification, we have*

$$\begin{aligned} K_C(x_0) &= \bigcap_{i=1}^s K_{C_i}(x_0) = \{ v \in \mathbf{X} \mid \nabla h^T(x_0)v \in \Lambda^* \} = \nabla h(x_0)^{-1}\Lambda^* \\ &= \{ v \in \mathbf{X} \mid \langle \nabla h_i(x_0), v\rangle = 0 \text{ for } i \in I_1;\ \langle \nabla h_i(x_0), v\rangle \geq 0 \text{ for } i \in I_2' \}, \end{aligned} \tag{3.6}$$

where Λ^* is given by (3.2) *and $\nabla h(x_0)^{-1}\Lambda^*$ is an inverse image of the linear operator $\nabla h(x_0) := (\nabla h_1(x_0), \nabla h_2(x_0), \ldots, \nabla h_s(x_0))$.*

Proof. By the definition of the contingent cone, we only have to prove $K_C(x_0) \supset \bigcap_{i=1}^s K_{C_i}(x_0)$ because the other inclusion is trivially verified and the other equalities are direct consequences of Proposition 3.2. Let $v \in \bigcap_{i=1}^s K_{C_i}(x_0)$, from Proposition 3.2, $\langle \nabla h_i(x_0), v\rangle = 0$ for $i \in I_1$ and $\langle \nabla h_i(x_0), v\rangle \geq 0$ for $i \in I_2'$. By the *basic constraint qualification* and Lemma 3.1, there exists an m-dimensional regular box $\mathcal{B}$ in $\mathbb{R}^s$ with m pairs of opposite sides $\{ (\Pi_j^1, \Pi_j^2) : j = 1, 2, \ldots, m \}$, such that for

$j \in I_1$,

$$\text{(3.7)} \qquad \begin{aligned} &\left\langle \nabla h_j(x_0), \sum_{i=1}^{s} \alpha_i \nabla h_i(x_0) \right\rangle > 0 \quad \text{for} \quad \alpha \in \Pi_j^1; \\ &\left\langle \nabla h_j(x_0), \sum_{i=1}^{s} \alpha_i \nabla h_i(x_0) \right\rangle < 0 \quad \text{for} \quad \alpha \in \Pi_j^2, \end{aligned}$$

and for $j \in I_2'$,

$$\text{(3.8)} \qquad \left\langle \nabla h_j(x_0), \sum_{i=1}^{s} \alpha_i \nabla h_i(x_0) \right\rangle > 0 \quad \text{for } \alpha \text{ in all sides of the box } \mathcal{B}.$$

Given an integer k, define

$$\text{(3.9)} \qquad x_{k,t}(\alpha) = x_0 + t\left(v + \frac{1}{k}\sum_{i=1}^{s} \alpha_i \nabla h_i(x_0)\right),$$

then, by Fréchet differentiability of the functions at x_0, for any $j \in I_1$ or I'_2, we have

$$\begin{aligned} h_j(x_{k,t}(\alpha)) &= h_j(x_0) + t\left\langle \nabla h_j(x_0), v + \frac{1}{k}\sum_{i=1}^{s} \alpha_i \nabla h_i(x_0) \right\rangle \\ &\quad + o\left(t\left(v + \frac{1}{k}\sum_{i=1}^{s} \alpha_i \nabla h_i(x_0)\right)\right) \\ &= t\left\langle \nabla h_j(x_0), v + \frac{1}{k}\sum_{i=1}^{s} \alpha_i \nabla h_i(x_0) \right\rangle + o(t). \end{aligned}$$

For an inactive constraint, i.e., for $j \in I_2''$ ($h_j(x_0) > 0$), because h_j is continuous around x_0, $h_j(x_{k,t}(\alpha)) > 0$ for all $\alpha \in \mathcal{B}$, when t is small enough. Also from (3.7) and (3.8), we can choose t small enough, denoted by t_k, such that for all $j \in I_1$, $h_j(x_{k,t_k}(\alpha)) > 0$ for $\alpha \in \Pi_j^1$ and $h_j(x_{k,t_k}(\alpha)) < 0$ for $\alpha \in \Pi_j^2$, and for all $j \in I_2'$, $h_j(x_{k,t_k}(\alpha)) \geq 0$ for $\alpha \in \mathcal{B}$. By Lemma 2.3, the generalized intermediate value theorem, there exists $\alpha'_k \in \mathcal{B}$ such that

$$\text{(3.10)} \qquad h_j(x_{k,t_k}(\alpha'_k)) = 0 \quad \text{for} \quad j \in I_1 \quad \text{and} \quad h_j(x_{k,t_k}(\alpha'_k)) \geq 0 \quad \text{for} \quad j \in I_2.$$

Therefore $x_k := x_{k,t_k}(\alpha_k') \in C$ and $(x_k - x_0)/t_k \to v$, i.e., $v \in K_C(x_0)$, which proves the result. □

It is necessary to mention that Ledzewicz-Kowalewska and Walczak [16] proved the same result for $h : \mathbf{X} \to \mathbf{Y}$ in Banach spaces $\mathbf{X}$ and $\mathbf{Y}$ under one more condition that h is Lipschitzian around x_0. Because the strict normal cone is the polar of the contingent cone, we have an expression for the strict normal cone to the same constraint set.

COROLLARY 3.4. *Under the assumptions of Theorem* 3.3, *we have*

$$\widehat{N}_C(x_0) = \left\{ -\sum_{i=1}^{s} \lambda_i \nabla h_i(x_0) : \lambda \in \Lambda \right\} = -\nabla h(x_0)\Lambda.$$

The same conclusions as Proposition 3.2, Theorem 3.3, and Corollary 3.4 for smooth constraint functions have appeared in the literature many times, see Borwein [3], Rockafellar [18], and Aubin and Frankowska [1]. Proposition 3.2 and Theorem 3.3

in finite dimensional space have appeared in Di [8] and Di and Poliquin [9] under a stricter constraint qualification. For the second-order contingent hull, we have the following exact expression.

PROPOSITION 3.5. (case of one constraint for the second-order contingent hull) *Assume the conditions of Proposition* 3.2. *If, in addition, h has second-order Fréchet derivative at x_0 and $v \in K_C(x_0)$. Then for $C = \{y : h(y) = 0\}$, we have*

$$K_C^2(x_0, v) = \{\, u \in \mathbf{X} \mid \langle v, \nabla^2 h(x_0)v\rangle + \langle \nabla h(x_0), u\rangle = 0 \,\}. \tag{3.11}$$

For $C = \{y : h(y) \geq 0\}$, we have

$$K_C^2(x_0, v) = \{\, u \in \mathbf{X} \mid \langle v, \nabla^2 h(x_0)v\rangle + \langle \nabla h(x_0), u\rangle \geq 0 \,\}. \tag{3.12}$$

Proof. Similar to Proposition 3.3, we prove only the case of equality constraint. Let $u \in K_C^2(x_0, v)$. By the definition of the second-order contingent hull, there exists $\{x_i\} \subset C$ and $t_i \downarrow 0$ such that $(x_i - x_0 - t_i v)/(\frac{1}{2}t_i^2) \to u$, i.e., $x_i = x_0 + t_i v + \frac{1}{2}t_i^2 u + o(t_i^2)$. By the differentiability of h at x_0,

$$\begin{aligned} 0 = h(x_i) - h(x_0) &= \left\langle \nabla h(x_0), t_i v + \frac{1}{2}t_i^2 u + o(t_i^2) \right\rangle \\ &\quad + \frac{1}{2}\left\langle t_i v + \frac{1}{2}t_i^2 u + o(t_i^2), \nabla^2 h(x_0)\big(t_i v + \frac{1}{2}t_i^2 u + o(t_i^2)\big) \right\rangle \\ &\quad + o\left(t_i v + \frac{1}{2}t_i^2 u + o(t_i^2)\right)^2 \\ &= \frac{1}{2}t_i^2\big(\langle \nabla h(x_0), u\rangle + \langle v, \nabla^2 h(x_0)v\rangle\big) + o(t_i^2), \end{aligned}$$

which means $\langle \nabla h(x_0), u\rangle + \langle v, \nabla^2 h(x_0)v\rangle = 0$, i.e., u belongs to the right-hand side of (3.11).

On the other hand, let $\langle \nabla h(x_0), u\rangle + \langle v, \nabla^2 h(x_0)v\rangle = 0$ for some u. For any positive integer i, we consider $y_i'(t) = x_0 + t\big(v + \frac{t}{2}u + \frac{t}{i}\nabla h(x_0)\big)$ and $y_i''(t) = x_0 + t\big(v + \frac{t}{2}u - \frac{t}{i}\nabla h(x_0)\big)$. Because h has second-order Fréchet derivative at x_0, we have

$$\begin{aligned} h(y_i'(t)) - h(x_0) &= t\left\langle \nabla h(x_0), v + \frac{t}{2}u + \frac{t}{i}\nabla h(x_0) \right\rangle \\ &\quad + \frac{t^2}{2}\left\langle v + \frac{t}{2}u + \frac{t}{i}\nabla h(x_0), \nabla^2 h(x_0)\left(v + \frac{t}{2}u + \frac{t}{i}\nabla h(x_0)\right)\right\rangle \\ &\quad + o\left(tv + \frac{t^2}{2}u + \frac{t^2}{i}\nabla h(x_0)\right)^2 \\ &= + \frac{t^2}{i}\langle \nabla h(x_0), \nabla h(x_0)\rangle + \frac{t^2}{2}\big(\langle \nabla h(x_0), u\rangle + \langle v, \nabla^2 h(x_0)v\rangle\big) + o(t^2), \\ &= + \frac{t^2}{i}\langle \nabla h(x_0), \nabla h(x_0)\rangle + o(t^2), \end{aligned} \tag{3.13}$$

and similarly,

$$h(y_i''(t)) - h(x_0) = -\frac{t^2}{i}\langle \nabla h(x_0), \nabla h(x_0)\rangle + o(t^2). \tag{3.14}$$

By uniform convergence in Fréchet differentiability, there exists a t, denoted by t_i with $t_i \to 0$, such that (3.13) is positive and (3.14) is negative. Thus, by the intermediate value theorem (one-dimensional situation), along the line segment between $y_i'(t_i)$ and

$y_i''(t_i)$, there exists y_i such that $h(y_i) - h(x_0) = 0$, i.e., $y_i \in C$. Obviously, we have $(y_i - x_0 - t_i v)/\frac{1}{2}t_i^2 \to u$, which means $u \in K_C^2(x_0, v)$, and the result is true. □

For the set C determined by a number of equality and/or inequality constraints, we have the following result about the second-order contingent hull.

THEOREM 3.6 (expression for the second-order contingent hull). *Assume the conditions of Theorem* 3.3. *If, in addition,* $h_i : \mathbf{X} \to \mathbb{R}$ *admit second-order Fréchet derivatives at* x_0 *for* $i = 1, 2, \ldots, s$, *and* $v \in K_C(x_0)$, *then*

$$
\begin{aligned}
K_C^2(x_0, v) &= \cap K_{C_i}^2(x_0, v) = \{\, u \in \mathbf{X} \mid \langle v, \nabla^2 h(x_0)v\rangle + \langle u, \nabla h(x_0)\rangle \in \Lambda^* \,\} \\
&= \left\{ u \in \mathbf{X} \;\middle|\; \begin{array}{lll} \langle v, \nabla^2 h_i(x_0)v\rangle + \langle u, \nabla h_i(x_0)\rangle = 0 & \text{for} & i \in I_1 \\ \langle v, \nabla^2 h_i(x_0)v\rangle + \langle u, \nabla h_i(x_0)\rangle \geq 0 & \text{for} & i \in I_2' \end{array} \right\}.
\end{aligned} \tag{3.15}
$$

Proof. We only have to prove $K_C^2(x_0, v) \supset \bigcap_{i=1}^s K_{C_i}^2(x_0, v)$. Let $u \in \bigcap_{i=1}^s K_{C_i}^2(x_0, v)$. Given a positive integer k, define

$$
x_{k,t}(\alpha) = x_0 + t\left(v + \frac{t}{2}u + \frac{t}{k}\sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right), \tag{3.16}
$$

then, by Fréchet differentiability of h_j at x_0, we have

$$
\begin{aligned}
h_j(x_{k,t}(\alpha)) &= h_j(x_0) + t\left\langle \nabla h_j(x_0), v + \frac{t}{2}u + \frac{t}{k}\sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right\rangle \\
&\quad + \frac{t^2}{2}\left\langle v + \frac{t}{2}u + \frac{t}{k}\sum_{i=1}^{s}\alpha_i \nabla h_i(x_0), \nabla^2 h_j(x_0)\left(v + \frac{t}{2}u + \frac{t}{k}\sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right)\right\rangle \\
&\quad + o\left(\left\| t\left(v + \frac{t}{2}u + \frac{t}{k}\sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right)\right\|^2\right) \\
&= h_j(x_0) + t\langle \nabla h_j(x_0), v\rangle + \frac{t^2}{k}\left\langle \nabla h_j(x_0), \sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right\rangle \\
&\quad + \frac{1}{2}t^2\big(\langle \nabla h_j(x_0), u\rangle + \langle v, \nabla^2 h_j(x_0)v\rangle\big) + o(t^2),
\end{aligned}
$$

therefore

$$
h_j(x_{k,t}(\alpha)) = \frac{t^2}{k}\left\langle \nabla h_j(x_0), \sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right\rangle + o(t^2),
$$

for $j \in I_1$, because $h_j(x_0) = 0$, $\langle \nabla h_j(x_0), v\rangle = 0$, and $\langle \nabla h_j(x_0), u\rangle + \langle v, \nabla^2 h_j(x_0)v\rangle = 0$ for such a j; and

$$
h_j(x_{k,t}(\alpha)) \geq \frac{t^2}{k}\left\langle \nabla h_j(x_0), \sum_{i=1}^{s}\alpha_i \nabla h_i(x_0)\right\rangle + o(t^2),
$$

for $j \in I_2'$, because $h_j(x_0) \geq 0$, $\langle \nabla h_j(x_0), v\rangle \geq 0$ and $\langle \nabla h_j(x_0), u\rangle + \langle v, \nabla^2 h_j(x_0)v\rangle \geq 0$ for such a j; and $h_j(x_{k,t}(\alpha)) > 0$ for $j \in I_2''$ if t is small enough. By the same reasoning as in the proof of Theorem 3.3, there exist t_k and α_k such that $t_k \downarrow 0$, $x_{k,t_k}(\alpha_k)$ satisfy (3.10), Therefore $x_k := x_{k,t_k}(\alpha_k) \in C$. It is easy to check $(x_k - x_0 - t_k v)/(\frac{1}{2}t_k^2) \to u$, i.e. $u \in K_C^2(x_0, v)$, and the result is proved. □

From (3.15), we know that $K_C^2(x_0, v)$ is convex for the set C determined by a number of equality and/or inequality constraints. For a closed convex set C, Lemma 2.1

describes some properties of $K_C^2(x_0, v)$, but its convexity is unknown.

As we have seen, the basic constraint qualification B_1 ensures that the contingent cone and the second-order contingent hull have the exact expressions in Theorems 3.3 and 3.6. The generalized intermediate value theorem, the alternative theorem, and Lemma 3.1 are major tools to prove these results.

4. Incorporating an abstract constraint set. In this section, we generalize Theorems 3.3 and 3.6 by incorporating an additional abstract constraint set. Correspondingly, the basic constraint qualification must be modified to accommodate the abstract constraint set.

B_2. BASIC CONSTRAINT QUALIFICATION. *Suppose that D in* (1.1) *is closed and $h = (h_1, h_2, \dots, h_s)$ are continuous around x_0 and Fréchet differentiable at x_0 for $i = 1, 2, \dots, s$. Assume that there is no nonzero $\lambda \in \Lambda$, defined by* (3.1), *such that* $\sum_{i=1}^s \lambda_i \nabla h_i(x_0) \in \widehat{N}_D(x_0)$.

THEOREM 4.1 (expression of the contingent cone). *Suppose that $\mathbf{X}$ is a finite dimensional space, D is a closed convex set, and $x_0 \in C \cap D$. Under the basic constraint qualification B_2, we have*

$$K_{C\cap D}(x_0) = \bigcap_{i=1}^s K_{C_i}(x_0) \bigcap K_D(x_0). \tag{4.1}$$

Proof. We only have to prove "$\supset$" in (4.1). Let v belong to the right-hand side of (4.1). Then there exists $\{x_k\} \subset D$ and $t_k \downarrow 0$ such that $v_k := (x_k - x_0)/t_k \to v$, $\langle \nabla h_i(x_0), v\rangle = 0$ for $i \in I_1$ and $\langle \nabla h_i(x_0), v\rangle \geq 0$ for $i \in I_2'$. Define $\beta_k := |(x_k - x_0)/t_i - v|$, then $\beta_k \to 0$ and $x_k = x_0 + t_k v + t_k \beta_k v_k'$ for some v_k' with $|v_k'| \leq 1$. Similar to the proof of Theorem 3.3, there exists an m-dimensional regular box $\mathcal{B}$ in $\mathbb{R}^s$ with m pairs of opposite sides $\{(\Pi_j^1, \Pi_j^2) : j = 1, 2, \dots, m\}$ and with 2^m vertices $\alpha'_1, \alpha'_2, \dots, \alpha'_{2^m}$, such that for $j \in I_1$,

$$\begin{aligned} &\left\langle \nabla h_j(x_0), \sum_{i=1}^s \alpha_i \nabla h_i(x_0) \right\rangle > 0 \quad \text{for} \quad \alpha \in \Pi_j^1; \\ &\left\langle \nabla h_j(x_0), \sum_{i=1}^s \alpha_i \nabla h_i(x_0) \right\rangle < 0 \quad \text{for} \quad \alpha \in \Pi_j^2, \end{aligned} \tag{4.2}$$

and for $j \in I_2'$,

$$\left\langle \nabla h_j(x_0), \sum_{i=1}^s \alpha_i \nabla h_i(x_0) \right\rangle > 0 \quad \text{for } \alpha \text{ in all sides of the box } \mathcal{B}. \tag{4.3}$$

Define

$$W := \left\{ w \in \mathbf{X} \;\middle|\; |w| \leq 1, \begin{array}{lll} \langle w, \nabla h_j(x_0)\rangle = 0 & \text{for} & j \in I_1 \\ \langle w, \nabla h_j(x_0)\rangle \geq 0 & \text{for} & j \in I_2' \end{array} \right\}. \tag{4.4}$$

Given an integer k, $\alpha \in \mathcal{B}$, $w \in W$, and $t > 0$, define

$$x_{k,t}(\alpha, w) = x_0 + t \cdot t_k \left(v + \beta_k^{\frac{1}{2}} \sum_{i=1}^s \alpha_i \nabla h_i(x_0) + \beta_k^{\frac{1}{3}} w \right), \tag{4.5}$$

then, by Fréchet differentiability of the functions at x_0, for any $j \in I_1$ or I_2, we have

$$h_j(x_{k,t}(\alpha,w)) = h_j(x_0) + t\cdot t_k\langle \nabla h_j(x_0), v\rangle + t\cdot t_k\beta_k^{\frac{1}{2}}\left\langle \nabla h_j(x_0), \sum_{i=1}^{s}\alpha_i\nabla h_i(x_0)\right\rangle$$
$$+ t\cdot t_k\beta_k^{\frac{1}{3}}\langle \nabla h_j(x_0).w\rangle + o\left(t\cdot t_k\left|v+\beta_k^{\frac{1}{2}}\sum_{i=1}^{s}\alpha_i\nabla h_i(x_0)+\beta_k^{\frac{1}{3}}w\right|\right),$$

which is positive for $j \in I_2''$, $\alpha \in \mathcal{B}$ and $w \in W$ when t is small enough (say for $t \le s_k$), because $h_j(x_0) > 0$; and is positive for $j \in I_2'$, $\alpha \in \mathcal{B}$ and $w \in W$ when t is small enough (say for $t \le s_k$), because $h_j(x_0) = 0$ and $\langle \nabla h_j(x_0), w\rangle \ge 0$; and equals

$$t\cdot t_k\beta_k^{\frac{1}{2}}\left\langle \nabla h_j(x_0), \sum_{i=1}^{s}\alpha_i\nabla h_i(x_0)\right\rangle + o\left(t\cdot t_k\left|v+\beta_k^{\frac{1}{2}}\sum_{i=1}^{s}\alpha_i\nabla h_i(x_0)+\beta_k^{\frac{1}{3}}w\right|\right),$$

for $j \in I_1$, $\alpha \in \mathcal{B}$, $w \in W$ and for all t, because $h_j(x_0) = 0$, $\langle \nabla h_j(x_0), v\rangle = 0$ and $\langle \nabla h_j(x_0), w\rangle = 0$. Therefore, for $j \in I_1$, we have

$$\begin{aligned} &h_j(x_{k,s_k}(\alpha,w)) > 0 \quad \text{for} \quad \alpha \in \Pi_j^1 \quad \text{and} \quad w \in W,\\ &h_j(x_{k,s_k}(\alpha,w)) < 0 \quad \text{for} \quad \alpha \in \Pi_j^2 \quad \text{and} \quad w \in W; \end{aligned} \tag{4.6}$$

and for $j \in I_2$, we have

$$h_j(x_{k,s_k}(\alpha,w)) \ge 0 \quad \text{for} \quad \alpha \in \mathcal{B} \quad \text{and} \quad w \in W. \tag{4.7}$$

Define $\mathcal{B}_k := \{\, x_{k,s_k}(\alpha,w) \mid \alpha \in \mathcal{B},\ w \in W \,\}$.

Assuming that there exists an $\alpha^0 \in \mathcal{B}$ such that the compact convex set

$$S_{k,s_k}(\alpha^0) := \{\, x_{k,s_k}(\alpha^0,w) \mid w \in W \,\}$$

is disjoint with D, then by the Separation Theorem, we know there exists a hyperplane strongly separating $S_{k,s_k}(\alpha^0)$ from D, i.e.,

$$\langle u_k, y\rangle \le \langle u_k, x_{k,s_k}(\alpha^0,w)\rangle, \tag{4.8}$$

for all $w \in W$ and $y \in D$, where $u_k \ne 0$. Without any restriction of generality, we assume $|u_k| = 1$ and $u_k \to v_0$. Hence in the limit, $\langle v_0, y - x_0\rangle \le 0$ for all $y \in D$, which means $v_0 \in \widehat{N}_D(x_0)$. On the other hand, given $w \in W$ such that $|w| = 1$, denote by $z_k(s_k) := s_k\cdot x_k + (1-s_k)\cdot x_0$ and $v_k(w) := x_{k,s_k}(\alpha^0,w) - z_k(s_k)$, then we have

$$z_k(s_k) = x_0 + s_k(x_k - x_0) = x_0 + s_k t_k v_k,$$

$$v_k(w) = s_k t_k\left(v - v_k + \beta_k^{\frac{1}{2}}\sum_{i=1}^{s}\alpha_i^0\nabla h_i(x_0) + \beta_k^{\frac{1}{3}}w\right),$$

and by (4.8),

$$\langle u_k, v_k(w)/|v_k(w)|\rangle \ge 0. \tag{4.9}$$

Noting $|v - v_k| = \beta_k$, we know $v_k(w)/|v_k(w)| \to w$. Thus, by taking limit in (4.9) as $k \to \infty$, we have $\langle v_0, w\rangle \ge 0$, which means $v_0 \in W^*$. By Lemma 2.4, there exists a nonzero $\lambda \in \Lambda$, defined by (3.1), such that $\sum_{i=1}^{s}\lambda_i\nabla h_i(x_0) = v_0 \in \widehat{N}_D(x_0)$. This contradicts the constraint qualification B_2, which means, for any given $\alpha \in \mathcal{B}$, $S_{k,s_k}(\alpha) \cap D \ne \emptyset$. Therefore, for any vertex α_i of $\mathcal{B}$, there exists $w_i \in W$ such that $z_{k,i} := x_{k,s_k}(\alpha_i, w_i) \in D$. Now $z_{k,1},\dots,z_{k,2^m}$ form a new m-dimensional regular box

$\mathcal{B}'$ (a subset of D, because D is convex). From (4.6) and (4.7), we can apply the generalized intermediate value theorem to $\mathcal{B}'$. Thus, there exists $x'_k \in \mathcal{B}' \in D \cap \mathcal{B}_k$ such that

$$h_j(x'_k) = 0 \quad \text{for} \quad j \in I_1 \quad \text{and} \quad h_j(x'_k) \geq 0 \quad \text{for} \quad j \in I_2,$$

i.e., $x'_k \in C \cap D$. It is easy to verify that $(x'_k - x_0)/(t_k s_k) \to v$, i.e., $v \in K_{C\cap D}(x_0)$, because $x'_k \in \mathcal{B}_k$, which proves the result. □

The Clarke normal cone and the strict normal cone to a convex set D are the same. The following is a generalization to Corollary 3.4.

COROLLARY 4.2. *Under the assumptions of Theorem* 4.1, *we have*

$$\widehat{N}_{C\cap D}(x_0) = -\nabla h(x_0)\Lambda + \widehat{N}_D(x_0). \tag{4.10}$$

Proof. We must prove that the right side is closed. Assume $z_n := x_n + y_n$ with $x_n \in -\nabla h(x_0)\Lambda$, $y_n \in \widehat{N}_D(x_0)$, and $z_n \to z_0$. Because $-\nabla h(x_0)\Lambda$ is a closed set in a finite dimensional subspace, if $\{x_n\}$ is bounded, then we can assume that $x_n \to \bar{x}_0 \in -\nabla h(x_0)\Lambda$, and hence $y_n = z_n - x_n \to z_0 - \bar{x}_0 \in \widehat{N}_D(x_0)$, i.e., z_0 belong to the right side of the equality. If $\{x_n\}$ is not bounded, then we can assume that $|x_n| \to \infty$ and $x_n/|x_n| \to \bar{x}_0 \neq 0$, thus, the limit of $z_n/|x_n| = x_n/|x_n| + y_n/|x_n|$ contradicts the basic constraint qualification, which finishes the proof. □

In the same way, we can prove the following generalization of Theorem 3.6.

THEOREM 4.3 (expression for the second-order contingent hull). *Assume the conditions of Theorem* 4.1. *If, in addition,* $h_i : \mathbf{X} \to \mathbb{R}$ *admit second-order Fréchet derivatives at* x_0 *for* $i = 1, 2, \ldots, s$, *and* $v \in K_{C\cap D}(x_0)$, *then*

$$\begin{aligned} K^2_{C\cap D}(x_0, v) &= \cap\, K^2_{C_i}(x_0, v) \cap K^2_D(x_0, v) \\ &= \{\, u \in K^2_D(x_0, v) \mid \langle v, \nabla^2 h(x_0)v\rangle + \langle u, \nabla h(x_0)\rangle \in \Lambda^* \,\}. \end{aligned} \tag{4.11}$$

Incorporating a closed convex set D in (4.1)–(4.3) is significant in dealing with optimization problems involving a convex set or convex functions. Using the results of Theorems 3.3 and 3.6, we can incorporate another kind of abstract constraint set, called a differentiably regular set, into the feasible set.

DEFINITION 4.4. *A set* C *is called differentiably regular at* x_0 *if* C *is determined by some functions satisfying the basic constraint qualification* B_1. *A set* C *is called strong differentiably regular at* x_0 *if* C *is differentiably regular and the functions determining* C *admit second-order Fréchet derivatives at* x_0.

Therefore a differentiably regular set is a set satisfying the condition of Theorem 3.3. It is trivial that all polyhedra are these kinds of sets, and sets determined by smooth functions are these kinds of sets subject to the basic constraint qualification B_1.

Corresponding to the results of (4.1)–(4.3) for closed convex sets, we have the similar results for differentiably regular sets.

THEOREM 4.5 (expression of the contingent cone). *Suppose that* $\mathbf{X}$ *is a Hilbert space,* D *is differentiably regular at* $x_0 \in C \cap D$. *Under the basic constraint qualification* B_2, *we have* (4.1).

Proof. Because D is differentiably regular at x_0, we can assume that D is determined by some constraint functions $h_{s+1}, \ldots, h_q$. Then $A = C \cap D$ is determined by $h_1, \ldots, h_m, \ldots, h_s, \ldots, h_q$. By Corollary 3.4, Lemma 2.4, and B_2, we know that A satisfies the basic constraint qualification B_1, therefore the proof is finished by applying Theorem 3.3 to D and A. □

COROLLARY 4.6. *Under the assumptions of Theorem* 4.5, *we have* (4.10).

We proved Theorem 4.5 by assimilating D into C. In the same way, we can prove the following result about the second-order contingent hull.

THEOREM 4.7 (expression for the second-order contingent hull). *Assume the conditions of Theorem* 4.5. *If, in addition,* $h_i : \mathbf{X} \to \mathbb{R}$ *admit second-order Fréchet derivatives at* x_0 *for* $i = 1, 2, \ldots, s$, *and* D *is strong differentiably regular, then* (4.11) *holds.*

A differentiably regular set is a set determined by some nonsmooth constraint functions differentiable at x_0. It is closed but not necessarily convex, and it can have very "uneven" boundary. Incorporating this kind of abstract constraint set is significant to deal with more general types of practical problems.

5. Optimality conditions. The classical optimality conditions of constrained optimization for smooth functions are the fundamental results in optimization theory, mathematical programming, and control theory. The optimality conditions are also the basic support to the practical applications in the area of numerical computing, operations research, transportation management, actuarial science, and so on. In this section, we prove the same classical optimality conditions under weaker assumptions on all functions involved. Instead of continuous differentiability or strict differentiability, we assume that all functions, including the objective function and the constraint functions, are continuous around x_0 and differentiable at x_0, a potential local optimum. Our results include first-order and second-order necessary conditions and first-order and second-order sufficiency conditions.

In §4, we considered two types of abstract constraint sets. Closed convex sets, polyhedra, and sets determined by smooth functions are specific examples of these two types of sets. To develop optimality conditions incorporating both kinds of abstract constraints, we organize the constraint qualifications in the following way.

H_1. FIRST-ORDER BASIC CONDITION. *Let* D *be a closed set in* $\mathbf{X}$, *a Hilbert space,* $h = (h_1, h_2, \ldots, h_s)$ *be functions on* $\mathbf{X}$. *The constraint of* $\mathcal{P}$ *is said to satisfy the first-order basic condition at* x_0 *if* h *satisfies the basic constraint qualification* B_1, *and* (4.1) *holds.*

H_2. SECOND-ORDER BASIC CONDITION. *Under the first-order basic conditions, we say that the constraint of* $\mathcal{P}$ *satisfies the second-order basic condition at* x_0, *if* h *admits second-order derivatives at* x_0, *and for any* $v \in K_{C\cap D}(x_0)$, (4.11) *holds.*

It is interesting to note that H_1 and H_2 are the mixture of the basic constraint qualification and conditions on D. Obviously, by the theorems in §§3 and 4, H_1 and H_2 are satisfied if D is a closed convex set, and H_1 (H_2) is satisfied if D is (strong) differentiably regular at x_0 subject to the basic constraint qualification B_1.

THEOREM 5.1 (optimality conditions for nonsmooth functions). *Let* x_0 *be a feasible solution to* $\mathcal{P}$ *defined by* (1.1). *Suppose that* $f : \mathbf{X} \to \mathbb{R}$ *is continuous around* x_0 *and differentiable at* x_0 .

I. *The necessary conditions for* x_0 *to be a local minimum are the following.*

(i) (*first-order necessary condition*). *Under the first-order basic condition* H_1, $\langle \nabla f(x_0), v\rangle \geq 0$ *for any* $v \in K_D(x_0)$ *such that*

$$\langle v, \nabla h(x_0)\rangle \in \Lambda^*. \tag{5.1}$$

(ii) (*second-order necessary condition*). *In the case of* $\langle \nabla f(x_0), v\rangle = 0$ *for such a* v *in* I(i), *under the second-order basic condition* H_2,

$$\langle v, \nabla^2 f(x_0)v\rangle + \langle u, \nabla f(x_0)\rangle \geq 0, \tag{5.2}$$

for any $u \in K_D^2(x_0, v)$ *such that*

$$\langle v, \nabla^2 h(x_0)v\rangle + \langle u, \nabla h(x_0)\rangle \in \Lambda^*. \tag{5.3}$$

II. *Let* $\mathbf{X}$ *be a finite dimensional space. Then* x_0 *is a strong local minimum (see the proof for a definition) if the following is true.*

(i) (*first-order sufficiency condition*). $\langle \nabla f(x_0), v\rangle > 0$ *for any nonzero* $v \in K_D(x_0)$ *satisfying* (5.1).

(ii) (*second-order sufficiency condition*). *Assume that both* D *and* $K_D^2(x_0, v)$ *are convex with* $0 \in K_D^2(x_0, v)$ *or* D *is strong differentiably regular. In the case of* $\langle \nabla f(x_0), v\rangle = 0$ *for such a nonzero* $v \in K_D(x_0)$ *satisfying* (5.1), *for any* $u \in K_D^2(x, v)$ *satisfying* (5.3), *the strict inequality holds in* (5.2).

Proof. I(i). See Guignard [11].

I(ii). Assume v and u satisfy the conditions in I(ii). By (5.3), Theorem 3.6, and H_2, we have $u \in K_A^2(x_0, v)$, which means there exists $\{x_i\} \subset A$ and $t_i \downarrow 0$ such that and $(x_i - x_0 - t_i v)/(\frac{1}{2}t_i^2) \to u$, i.e., $x_i = x_0 + t_i v + \frac{1}{2}t_i^2 u + o(t_i^2)$. Because f is differentiable at x_0, and x_0 is a local minimum, by Taylor expansion, we have

$$0 \le f(x_i) - f(x_0) = \frac{t_i^2}{2}\left(\langle \nabla f(x_0), u\rangle + \langle v, \nabla^2 f(x_0)v\rangle\right) + o(t_i^2), \tag{5.4}$$

which means $\langle \nabla f(x_0), u\rangle + \langle v, \nabla^2 f(x_0)v\rangle \ge 0$.

II(i) is implied by Corollary 2.1 in Studniarski [20]. This proves that if x_0 is not a strong local minimum, then there exists $\delta_k \downarrow 0$, $\epsilon_k \downarrow 0$ and $\{x_k\} \subset A$ such that we have $|x_k - x_0| \le \epsilon_k$, $f(x_k) < f(x_0) + \delta_k |x_k - x_0|^2$, $v \in K_A(x_0) \subset K_C(x_0) \cap K_D(x_0)$, and the following contradiction by denoting $t_k := |x_k - x_0|$ and $v_k := (x_k - x_0)/t_k$ and assuming $v_k \to v$:

$$\begin{aligned} &\langle v, \nabla f(x_0)\rangle \le 0, \\ &\langle v, \nabla h_i(x_0)\rangle = 0 \quad \text{for} \quad i \in I_1, \\ &\langle v, \nabla h_i(x_0)\rangle \ge 0 \quad \text{for} \quad i \in I_2'. \end{aligned} \tag{5.5}$$

For II(ii), either D and $K_D^2(x_0, v)$ are closed and convex or D is strong differentiably regular. If D is strong differentiably regular, like the proof of Theorem 4.5, we can assimilate D into a part of C, which means that we can assume, without loss of generality, $D = \mathbf{X}$, $K_D(x_0) = \mathbf{X}$, and $K_D^2(x_0, v) = \mathbf{X}$. Therefore, we can just assume that D and $K_D^2(x_0, v)$ are closed and convex.

Denote C' the set of u such that

$$\begin{aligned} &\langle \nabla h_i(x_0), u\rangle = -\langle v, \nabla^2 h_i(x_0)v\rangle \quad \text{for} \quad i \in I_1, \\ &\langle \nabla h_i(x_0), u\rangle \ge -\langle v, \nabla^2 h_i(x_0)v\rangle \quad \text{for} \quad i \in I_2', \end{aligned} \tag{5.6}$$

and define $A' := C' \cap K_D^2(x_0, v)$, then for any $u \in A'$, we have $\langle \nabla f(x_0), u\rangle > -\langle v, \nabla^2 f(x_0)v\rangle$. By Lemma 2.1, $\alpha K_D^2(x_0, v) \in K_D^2(x_0, v)$ for any $\alpha \ge 1$. So the minimization problem $\min_{u \in A'} \langle \nabla f(x_0), u\rangle$ has a solution $\alpha_0 > -\langle v, \nabla^2 f(x_0)v\rangle$ reached at $u' \in A'$, which means $\langle \nabla f(x_0), u - u'\rangle \ge 0$ for any $u \in A'$, i.e., $-\nabla f(x_0) \in \widehat{N}_{A'}(u')$. Define $E' := u' + cl\ \mathrm{cone}(D - x_0)$, then by Lemma 2.1, $E' \subset K_D^2(x_0, v)$, which means

$$\widehat{N}_D(x_0) = \widehat{N}_{D - x_0}(0) \supset \widehat{N}_{cl\ \mathrm{cone}(D - x_0)}(0) = \widehat{N}_{E'}(u') \supset \widehat{N}_{K_D^2(x_0, v)}(u'). \tag{5.7}$$

Now we can apply Corollary 4.2 to A' (because H_1 and (5.7) ensures the condition of Corollary 4.2 for A'), i.e., there exist $\lambda_1, \dots, \lambda_s$ with $\lambda_i \geq 0$ if i is in I_2' and i is also an active index for u' satisfying (5.6), and $\lambda_i = 0$ for other $i \in I_2'$ or $i \in I_2''$, and $u^* \in \widehat{N}_{K_D^2(x_0,v)}(u')$ such that

$$\nabla f(x_0) - \sum_{i=1}^{s} \lambda_i \nabla h_i(x_0) + u^* = 0.$$

Correspondingly (from (5.6) and the properties of λ_i), we have

$$\langle v, \nabla^2 f(x_0)v\rangle - \sum_{i=1}^{s} \lambda_i \langle v, \nabla^2 h_i(x_0)v\rangle - \langle u^*, u'\rangle > 0.$$

Because $u^* \in \widehat{N}_{K_D^2(x_0,v)}(u')$, therefore $\langle u^*, u - u'\rangle \leq 0$ for all $u \in K_D^2(x_0, v)$. Now $0 \in K_D^2(x_0, v)$, and by Lemma 2.1, $2u' \in K_D^2(x_0, v)$, thus $\langle u^*, u'\rangle = \langle u^*, 2u' - u'\rangle \leq 0$, and $\langle u^*, -u'\rangle = \langle u^*, 0 - u'\rangle \leq 0$, which means

$$\langle v, \nabla^2 f(x_0)v\rangle - \sum_{i=1}^{s} \lambda_i \langle v, \nabla^2 h_i(x_0)v\rangle > 0.$$

Writing $g(x) := f(x) - \sum_{i=1}^{s} \lambda_i h_i(x) + \langle u^*, x - x_0\rangle$, we obtain $\nabla g(x_0) = 0$ and $\langle v, \nabla^2 g(x_0)v\rangle > 0$.

If $\langle v, \nabla f(x_0)\rangle = 0$, then the second-order condition comes into force. By second-order Fréchet differentiability and the fact $\nabla g(x_0) = 0$,

$$g(x_0 + t_k w) - g(x_0) = \frac{t_k^2}{2}\langle w, \nabla^2 g(x_0)w\rangle + o(t_k^2),$$

and o is uniform with respect to w in any bounded set, say $\{\, w \mid \langle w, w\rangle \leq 1 \,\}$. Therefore

$$g(x_0 + t_k v_k) - g(x_0) = \frac{t_k^2}{2}\langle v_k, \nabla^2 g(x_0)v_k\rangle + o(t_k^2), \tag{5.8}$$

where o is uniform with respect to all k. Furthermore, we have

$$g(x_0 + t_k v_k) \leq g(x_0) + \langle u^*, x_k - x_0\rangle + \delta_k t_k^2.$$

By (5.7), $u^* \in \widehat{N}_D(x_0)$, i.e., $\langle u^*, x_k - x_0\rangle \leq 0$, thus, $g(x_0 + t_k v_k) \leq g(x_0) + \delta_k t_k^2$. Therefore,

$$\delta_k t_k^2 \geq \frac{t_k^2}{2}\langle v_k, \nabla^2 g(x_0)v_k\rangle + o(t_k^2),$$

which means $0 \geq \langle v, \nabla^2 g(x_0)v\rangle > 0$. This contradiction finishes the proof. □

It is necessary to mention that the proof of II is motivated by Rockafellar's lecture notes [19] for smooth functions and a polyhedral set D. In the above proof, care is taken to handle the abstract constraint set D. The condition of Fréchet differentiability is necessary, because we need uniform convergence with respect to a bounded set of v_k in (5.8) as $t_k \downarrow 0$ in the proof.

The Lagrange function for $\mathcal{P}$ is $L(x, \lambda) = f(x) - \sum_{i=1}^{s} \lambda_i h_i(x)$ which is defined on $D \times \Lambda$, where Λ is defined by

$$\Lambda := \{\, \lambda \in \mathbb{R}^s \mid \lambda_i \geq 0 \quad \text{for } i \in I_2 \,\}. \tag{5.9}$$

It is direct to obtain the following:

$$
\begin{aligned}
\nabla_x L(x,\lambda) =& \nabla f(x) - \sum_{i=1}^{s} \lambda_i \nabla h_i(x), \\
\nabla_{xx} L(x,\lambda) =& \nabla^2 f(x) - \sum_{i=1}^{s} \lambda_i \nabla^2 h_i(x), \\
\nabla_\lambda L(x,\lambda) =& -(h_1(x),\dots,h_s(x)),
\end{aligned} \tag{5.10}
$$

$$
\widehat{N}_\Lambda(\lambda) = \left\{ (y_1,\dots,y_s) \;\middle|\; \begin{array}{ll} y_i \le 0 & \text{for } i \in I_2 \text{ with } \lambda_i = 0 \\ y_i = 0 & \text{for all other } i \end{array} \right\}.
$$

Denote by

$$
M(x_0) := \{\, \lambda \in \Lambda \mid \nabla_\lambda L(x_0,\lambda) \in \widehat{N}_\Lambda(\lambda) \,\}, \tag{5.11}
$$

$$
F(x_0) := \left\{ v \in K_D(x_0) \;\middle|\; \begin{array}{lll} \langle \nabla f(x_0), v\rangle = & 0 & \\ \langle \nabla h_i(x_0), v\rangle = & 0 & \text{for } i \in I_1 \\ \langle \nabla h_i(x_0), v\rangle \ge & 0 & \text{for } i \in I_2' \end{array} \right\}. \tag{5.12}
$$

If D is assumed to be differentiably regular, we can prove a new Lagrange multiplier rule incorporating an abstract constraint more general than a polyhedron.

THEOREM 5.2 (optimality conditions in Lagrange multiplier form). *Let x_0 be a feasible solution to $\mathcal{P}$ defined by* (1.1). *Assume that the corresponding conditions of Theorem* 5.1 *hold and D is differentiably regular for first-order result and strong differentiably regular for second-order result.*

I. *The necessary conditions for x_0 to be a local minimum are the following.*

(i). *There exists a nonzero vector $\lambda_0 \in M(x_0)$, defined by* (5.11) *such that $\langle \nabla_x L(x_0,\lambda), v\rangle \ge 0$ for all $v \in K_D(x_0)$;*

(ii). *For every $v \in K_D(x_0)$ such that $\langle \nabla_x L(x_0,\lambda), v\rangle = 0$, then $\langle v, \nabla^2_{xx} L(x_0,\lambda_0)v\rangle + \langle \nabla_x L(x_0,\lambda_0), u\rangle \ge 0$ for all $u \in K_D^2(x_0,v)$.*

II. *A feasible point x_0 is a strong local minimum if the following are true.*

(i). *There exists a nonzero vector $\lambda_0 \in M(x_0)$ such that $\langle \nabla_x L(x_0,\lambda), v\rangle > 0$ for all $v \in K_D(x_0)$.*

(ii). *For every $v \in K_D(x_0)$ such that $\langle \nabla_x L(x_0,\lambda), v\rangle \ge 0$, and if the equality holds for such a v, $\langle v, \nabla^2_{xx} L(x_0,\lambda_0)v\rangle + \langle \nabla_x L(x_0,\lambda_0), u\rangle > 0$ for all $u \in K_D^2(x_0,v)$.*

Proof. The special case without an abstract constraint $x \in D$ is easy to prove by applying Lemma 2.4 to Theorem 5.1. Because D is (strong) differentiably regular, by the *first-order* (*second-order*) *basic condition* H_1 (H_2), the special case applies to $A = C \cap D$, which is a set determined by a number of functions (some of which determine C, and others determine D), i.e.,

$$
D := \{\, x \in \mathbf{X} \mid h_i(x) = 0 \text{ for } i \in I_3 \text{ and } h_i(x) \ge 0 \text{ for } i \in I_4 \,\}, \tag{5.13}
$$

here $I_3 := \{s+1,\dots,p\}$ and $I_4 := \{p+1,\dots,q\}$. We denote by I'_4 and I''_4, subsets of I_4, the index set of active inequality constraints and the index set of inactive inequality

constraints. The corresponding Lagrange function is

$$L^*(x,\lambda^*) = f(x) - \sum_{i=1}^{s} \lambda_i^* h_i(x) - \sum_{i=s+1}^{q} \lambda_i^* h_i(x), \tag{5.14}$$

with $\lambda^* := (\lambda_1^*, \dots, \lambda_q^*)$ defined by (5.9) (s and I_2 in (5.9) are actually q and $I_2 \cup I_4$ here). The special case concludes $-\nabla_x L^*(x_0, \lambda_0^*) = 0$, which implies first-order result because

$$\nabla_x L(x_0, \lambda_0^*) = \nabla f(x_0) - \sum_{i=1}^{s} \lambda_i^* \nabla h_i(x_0) = \sum_{i=s+1}^{q} \lambda_i^* \nabla h_i(x_0) =: -w \in \widehat{N}_D(x_0)$$

from Corollary 3.4. The second-order necessary result is true, because the special case concludes

$$\left\langle \left(\nabla_{xx}^2 f(x_0) - \sum_{i=1}^{s} \lambda_i^* \nabla_{xx}^2 h_i(x_0) - \sum_{i=s+1}^{q} \lambda_i^* \nabla_{xx}^2 h_i(x_0) \right) v, v \right\rangle \geq 0,$$

and for any $u \in K_D^2(x_0, v)$, from (3.15) in Theorem 3.6, we have

$$\left\langle \sum_{i=s+1}^{q} \lambda_i^* \nabla_{xx}^2 h_i(x_0) v, v \right\rangle \geq - \left\langle \sum_{i=s+1}^{q} \lambda_i^* \nabla h_i(x_0), u \right\rangle = \langle w, u \rangle = -\langle \nabla_x L(x_0, \lambda_0), u \rangle.$$

For the second-order sufficiency, we know

$$\langle v, \nabla_{xx}^2 f(x) v \rangle - \left\langle v, \sum_{i=1}^{s} \lambda_i^* \nabla_{xx}^2 h_i(x_0) v \right\rangle > \langle w, u \rangle$$

for all $u \in K_D^2(x_0, v)$, which means,

$$\langle v, \nabla_{xx}^2 f(x) v \rangle - \left\langle v, \sum_{i=1}^{s} \lambda_i^* \nabla_{xx}^2 h_i(x_0) v \right\rangle + \left\langle \nabla f(x_0) - \sum_{i=1}^{s} \lambda_i^* \nabla h_i(x_0), u \right\rangle > 0.$$

Therefore, $\langle v, \nabla_{xx}^2 f(x) v \rangle + \langle \nabla f(x_0), u \rangle > 0$ for any $u \in K_D^2(x_0, v)$ such that $\langle v, \nabla_{xx}^2 h(x) v \rangle + \langle \nabla h(x_0), u \rangle \in \Lambda^*$ defined by (3.2), and the proof is finished. □

It is interesting to note that, for the first-order conditions in the above result, a closed convex set D is also fine as an abstract constraint set.

It is necessary to mention that the first-order necessary condition for $\mathcal{P}$ without an abstract constraint has appeared in Halkin [12], proved by using an implicit function theorem, and later in Di [8] and Di and Poliquin [9] under a slightly stricter constraint qualification: linear independency of $\nabla h_1(x_0), \dots, \nabla h_s(x_0)$, proved by a similar approach that we used here. It is worth mentioning that Łędzewicz-Kowalewska [14, 15] has proved first-order necessary condition under the assumptions that constraint functions are Fréchet differentiable around x_0 and inequality constraint functions are continuously differentiable around x_0.

If D is a closed convex set, then Theorem 5.1 improves Theorems 9.1 and 9.2 in Borwein [3], where the similar second-order necessary conditions and second-order sufficiency conditions are obtained for problems without incorporating inequality constraints under the assumption that the equality constraint functions h_i are strictly differentiable at x_0, twice Fréchet differentiable at x_0, and the objective function f is twice differentiable at x_0.

It is necessary to mention that a second-order necessary condition in terms of Ursescu tangency has appeared in Cominetti [5] for the problem

$$\min\{ f(x) \ : \ x \in D \quad \text{and} \quad h(x) \in C \},$$

where C and D are closed and convex, and f, h are assumed twice differentiable at some local optimal x_0 (and hence continuously differentiable around x_0 in Cominetti's sense). It is important to note that the notion of second-order Lipschitz, introduced in the thesis of Milosz [17] and the note by Cominetti and Correa [6], has been used to develop second-order optimality condition for nondifferentiable functions, and the result is very interesting; see Ioffe [13].

Lagrange multiplier rule for nonsmooth problems appears in the form of set inclusion involving sets of certain kinds of generalized differentials. However, applying to a minimization problem with functions differentiable at some local optimum x_0 (not strictly differentiable), which is still a nonsmooth problem, Lagrange multiplier rule in the form of set inclusion is not as conclusive as Theorem 5.2. For example, if h_i has Fréchet derivative $\nabla h_i(x_0)$ at x_0, then $\nabla h_i(x_0)$ is an element in the Clarke generalized gradient $\partial h_i(x_0)$, which is not necessarily a singleton. (It is a singleton if h is strictly differentiable.)

The optimality conditions for *max-type* nonsmooth problems were developed in Ben-Tal and Zowe [2] for smooth functions. This kind of nonsmooth optimization problem can be described to minimize

$$f(x) := \sum_{i=1}^{m} h_i(f_i(x)), \quad \text{where} \quad f_i(x) := \max_{j=1,\dots,s_i} f_{i,j}(x).$$

A more specific example is to minimize $f(x) := \max_{j=1,\dots,s} f_j(x)$. These problems are also described as typical examples of composite convex nonsmooth problems (Examples 1.2 and 1.7) in Rockafellar [18]. By applying the theorems of this section to the above max-type problems, the assumptions on all functions are reduced to "continuous around and differentiable at x_0"; The details are worked out in another paper and the results are interesting.

It is clear that our optimality conditions can be modified and applied to *Pareto* optimization problems.

We conclude this paper by stressing that, although many different kinds of constraint qualifications have been proposed since the 1940s, perhaps an appropriate constraint qualification for problem $\mathcal{P}$ defined by (1.1) is (4.1) for the first-order, and (4.11) for the second-order, according to the results developed in this article.

Acknowledgments. It is a pleasure to thank Dr. René Poliquin for his support and encouragement to carry out this research. His initial comments about second-order necessary condition and incorporating an abstract constraint like a polyhedron were very positive in reaching the current version of this paper. Thanks to the Department of Mathematics, University of Alberta, for its support and research environment.

REFERENCES

[1] I.-P. Aubin and H. Frankowska, *Set-Valued Analysis*, Birkhäuser Boston, Cambridge, MA, 1990.

[2] A. Ben-Tal and J. Zowe, *Necessary and sufficient optimality conditions for a class of nonsmooth minimization problems*, Math. Programming, 24 (1982), pp. 70–91.

[3] J. M. BORWEIN, *Stability and regular points of inequality system*, J. Optim. Theory Appl., 48 (1986), pp. 9–52.
[4] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.
[5] R. COMINETTI, *Metric regularity, tangent set, and second-order optimality conditions*, Appl. Math. Optim., 21 (1990), pp. 265–287.
[6] R. COMINETTI AND R. CORREA, *Sur une dérivée du second ordre en analyse non différentiable*, C.R. Acad Sci. Paris, 303 (1986), pp. 861–864.
[7] B.D. CRAVEN, *Mathematical Programming and Control Theory*, Chapman and Hall Ltd, London, 1978.
[8] S. DI, *The Trust Region Approach for Conic models and Calmness of Optimization Problems*, Ph.D. thesis, University of Alberta, Canada, 1992.
[9] S. DI AND R. POLIQUIN, *The contingent cone to a set defined by equality and inequality constraints at a (Fréchet) differentiable point*, J. Optim. Theory Appl., 189 (1994), pp. 502–513.
[10] T.M. FLETT, *Differential Analysis*, Cambridge University Press, Cambridge, 1980, pp. 277–278.
[11] M. GUIGNARD, *Generalized Kuhn–Tucker conditions for mathematical programming problems in a Banach space*, SIAM J. Control, 7 (1969), pp. 232–241.
[12] H. HALKIN, *Implicit functions and optimization problems without continuous differentiability of the data*, SIAM J. Control, 12 (1974), pp. 229–236.
[13] A. IOFFE, *On some recent developments in the theory of second order optimality conditions*, in Optimization, Lecture Notes in Mathematics, 1405, S. Dolecki, ed., Springer-Verlag, New York, 1988, pp. 229–236.
[14] U. ŁĘDZEWICZ-KOWALEWSKA, *Euler–Lagrange equation in the case of nonlinear equality constraints*, J. Optim. Theory Appl., 71 (1991), pp. 549–568.
[15] ———, *A necessary condition for the extremal problems under Gâteaux differentiability*, J. Math. Anal. Appl., 134 (1988), pp. 158–169.
[16] U. ŁĘDZEWICZ-KOWALEWSKA AND P. G. WALCZAK, *On the Lusternik theorem for nonsmooth operators*, 1992, preprint.
[17] T. MILOSZ, *Thesis*, Institute of Mathematics, Polish Academy of Sciences, Poland, 1988.
[18] R. T. ROCKAFELLAR, *First- and second-order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 307 (1988), pp. 75–108.
[19] ———, *Nonsmooth Optimization*, Lecture Notes at University of Washington, Seattle, manuscript.
[20] M. STUDNIARSKI, *Necessary and sufficient conditions for isolated local minima of nonsmooth functions*, SIAM J. Control Optim., 24 (1986), pp. 1044–1049.
[21] D. E. WARD AND J. M. BORWEIN, *Nonsmooth calculus in finite dimensions*, SIAM J. Control Optim., 25 (1987), pp. 1312–1340.
[22] E. ZEIDLER, *Nonlinear Functional Analysis and its Applications* I, Springer-Verlag, London, 1985.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 198–211, February 1996
© 1996 Society for Industrial and Applied Mathematics
011

DINI DERIVATIVES OF THE MARGINAL FUNCTION OF A NON-LIPSCHITZIAN PROGRAM*

D. E. WARD†

Abstract. Upper and lower bounds are established for the Dini directional derivatives of the marginal function of a parametric mathematical program. In this program, the equality constraint functions are assumed to be strictly differentiable, but the objective and inequality constraint functions can belong to a large class of non-Lipschitzian functions. A nonsmooth version of the Mangasarian–Fromovitz constraint qualification is also assumed. The main tool in the proofs of these bounds is the calculus of tangent cones.

Key words. marginal function, differential stability, tangent cone, directional derivative, subdifferential

AMS subject classifications. primary 90C31; secondary 49J52, 90C30

1. Introduction.

We consider the parametric mathematical program

$$\text{(P(s))} \qquad v(s) := \min\{f(x) \mid g_i(x) \le s_i,\ i \in J,\ h(x) = s_{m+1},\ x \in C\},$$

where f, $g_i : \mathbb{R}^n \to (-\infty, +\infty]$, $h : \mathbb{R}^n \to \mathbb{R}^p$, $J := \{1, \dots, m\}$, C is a nonempty subset of $\mathbb{R}^n$, and $s = \{s_1, \dots, s_{m+1}) \in \mathbb{R}^{m+p}$. The sensitivity of the *marginal function* v to changes in the parameter s is an important and much-studied issue in optimization (see, for instance, [7]).

Although v is generally nondifferentiable, bounds on

$$v^+(s;d) := \limsup_{t \to 0^+} (v(s+td) - v(s))/t$$

and

$$v_+(s;d) := \liminf_{t \to 0^+} (v(s+td) - v(s))/t,$$

the upper and lower *Dini directional derivatives* of v at s can often be calculated [2, 4, 5, 7–10, 18–20, 22]. For example, in the case in which f, g_i, and h are continuously differentiable, $C = \mathbb{R}^n$, and the Mangasarian–Fromovitz constraint qualification is satisfied at solutions of $(P(0))$, Gauvin and Tolle [9] have established upper bounds for $v^+(0;d)$ and lower bounds for $v_+(0;d)$ in terms of support functions of sets of Karush–Kuhn–Tucker (KKT) multipliers for $(P(0))$.

The development of a calculus and optimization theory for locally Lipschitzian functions [5, 11] has opened the way for more general results of this type. Using a constraint qualification and multiplier set involving Clarke generalized gradients, Auslender [2] has extended the Gauvin–Tolle bounds to the situation where f and g_i are locally Lipschitzian, h is continuously differentiable, and $C = \mathbb{R}^n$. Clarke [5, §6.5] has gone even further in this direction, proving similar results under the assumption that f, g_i, and h are locally Lipschitzian and C is closed.

Today, a calculus for generalized directional derivatives and subdifferentials is firmly in place for large classes of non-Lipschitzian functions (e.g., [1, 5, 14, 16, 23]), and it is natural to ask whether non-Lipschitzian versions of the Gauvin–Tolle bounds can be derived with this calculus. We have made some progress in this direction in [22], where we consider an inequality-constrained program in which f and g_i have a sort of "directionally Lipschitzian" property.

* Received by the editors June 28, 1993; accepted for publication August 12, 1994.
† Department of Mathematics and Statistics, Miami University, Oxford, OH 45056-1641 (dw86mthf@miamiu.muohio.edu).

In this paper, we improve upon the results of [22] in two ways. First, we include a strictly differentiable equality constraint function h in addition to the inequality constraints of [22]. Second, our assumptions on f and g_i no longer demand that these functions possess the directionally Lipschitzian property of [22]. These improvements are made possible by a simplified method of proof that reduces the derivation of bounds for $v^+(0;d)$ and $v_+(0;d)$ to the problem of calculating tangent cones to intersections of sets. This latter problem can be readily solved with the help of the tangent cone calculus from [1] and [23].

An outline of this paper now follows. In §2, we review the definitions of tangent cones and their associated directional derivatives, along with the concept of "strong general position" for convex cones, which will help us in stating our main results. We also apply a metric regularity theorem of Borwein [4] to prove a technical lemma that will enable us to work with the equality constraint in (P(s)). In §3, we use the calculus of tangent cones to deduce bounds on $v^+(0;s)$ and $v_+(0;s)$ in primal form; i.e., in terms of directional derivatives of f, g_i, and h and tangent cones of C. Then in §4, we review generalized KKT conditions for $(P(0))$, define corresponding multiplier sets, and use Lagrangian duality to establish bounds for $v^+(0;s)$ and $v_+(0;s)$ in terms of support functions of these multiplier sets. In our upper bound for $v^+(0;s)$, we use a subdifferential "smaller" than the Clarke subdifferential and are thus able to sharpen the corresponding bound from [2] as well as extend it beyond the locally Lipschitzian case. We prove our lower bound for $v_+(0;s)$ under conditions weaker than those given in [2] and [22]. By comparing our upper and lower bounds, we also obtain conditions implying directional differentiability of v.

We pause now to review some essential definitions and notation. Throughout this paper, $\|\cdot\|$ will denote the Euclidean norm on $\mathbb{R}^n$. For $x \in \mathbb{R}^n$ and $\varepsilon > 0$, we set $B(x,\varepsilon) := \{y \in \mathbb{R}^n \mid \|y - x\| \leq \varepsilon\}$; and for $S \subset \mathbb{R}^n$, we define the *distance function*

$$\operatorname{dist}(x \mid S) := \inf\{\|y - x\| : y \in S\}$$

and the *indicator function*

$$i_S(x) := \begin{cases} 0 & \text{if } x \in S, \\ +\infty & \text{else.} \end{cases}$$

We denote the *interior* of S by int S, the *relative interior* of S by ri S, and the *closure* of S by cl S. The *recession cone* of S is the set

$$S^\infty := \{y \mid x + ty \in S \quad \forall x \in S \quad \forall t \geq 0\}.$$

We say that S is *closed near* $x \in S$ if there exists $\varepsilon > 0$ such that $S \cap B(x,\varepsilon)$ is closed. For a cone S, we define the (negative) *polar cone* $S^\circ := \{y \in \mathbb{R}^n \mid \langle y, x\rangle \leq 0 \ \forall x \in S\}$, where $\langle \cdot, \cdot \rangle$ denotes the usual inner product on $\mathbb{R}^n$.

For a function $f : \mathbb{R}^n \to \overline{\mathbb{R}} := [-\infty, +\infty]$, the *epigraph* of f is the set epi $f := \{(x,r) \mid f(x) \leq r\}$, and the *effective domain* of f is defined by dom $f := \{x \mid f(x) < +\infty\}$. If dom $f \neq \emptyset$ and f never takes on the value $-\infty$, then f is said to be *proper*. If epi f is closed near $(x, f(x))$, then f is said to be *strictly lower semicontinuous* (LSC) at x [18].

2. Preliminaries. Just as the derivative of a function can be viewed in terms of a local linear approximation to its graph, so can many directional derivatives be described in terms of local conical approximations, called *tangent cones*, to epigraphs. In this paper, we think of a tangent cone A as a set-valued mapping $(S,x) \mapsto A(S,x)$ such that for all $S \subset \mathbb{R}^n$ and $x \in S$, $A(S,x)$ is a (possibly empty) cone. We say that A has a certain property (e.g., A is closed or A is convex) if $A(S,x)$ has that property for all $S \subset \mathbb{R}^n$ and $x \in S$.

In this paper, we have use for three tangent cones: the *adjacent cone*, defined for $S \subset \mathbb{R}^n$ and $x \in S$ by

$$k(S,x) := \{y \mid \forall \varepsilon > 0,\ \exists \lambda > 0 \text{ such that } \forall t \in (0,\lambda),\ \exists v \in B(y,\varepsilon) \text{ with } x + tv \in S\};$$

its recession cone

$$k^{\infty}(S,x) := \{y \mid k(S,x) + y \subset k(S,x)\};$$

and the *Clarke tangent cone*

$$\begin{aligned} T(S,x) := \{y \mid \forall \varepsilon > 0,\ \exists \lambda > 0 \text{ such that } \forall t \in (0,\lambda),\\ \forall z \in S \cap B(x,\lambda),\ \exists v \in B(y,\varepsilon) \text{ with } z + tv \in S\}. \end{aligned}$$

All three of these tangent cones are closed and k^{∞} and T are convex. Also, note that $T \subset k^{\infty} \subset k$. For more on the properties of these cones, see, for example, [1, 4–6, 14, 16, 17, 21, 23].

For a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a tangent cone A, the *A directional derivative* of f at x in the direction y is defined by

$$f^A(x;y) := \inf\{r \mid (y,r) \in A(\operatorname{epi} f,\ (x, f(x)))\}.$$

If A is closed and $(0,1) \in A(\operatorname{epi} f,\ (x,f(x)))^{\infty}$, then

$$\operatorname{epi} f^A(x;\cdot) = A(\operatorname{epi} f,\ (x,f(x))). \tag{2.1}$$

(See [6, Thm. 3.1].) This is the case, in particular, for $A = k$, k^{∞}, and T.

It is often possible to express $f^A(x;y)$ as a generalized limit of difference quotients [1, 6, 14, 16, 23]. For example, we can write

$$\begin{aligned} f^k(x;y) &= \limsup_{t \to 0^+} \inf_{v \to y} (f(x+tv) - f(x))/t \\ &:= \sup_{\varepsilon > 0} \inf_{\lambda > 0} \sup_{t \in (0,\lambda)} \inf_{v \in B(y,\varepsilon)} (f(x+tv) - f(x))/t. \end{aligned}$$

The T directional derivative is also known as the *upper subderivative*. It is extensively discussed in [1, 16, 17, 23, 5, §2.9].

If f is locally Lipschitzian near x, then $f^k(x;y)$, $f^{k^{\infty}}(x;y)$, and $f^T(x;y)$ are finite for all y; in fact, $f^k(x;\cdot) = f^+(x;\cdot)$, and $f^T(x;\cdot) = f^{\circ}(x;\cdot)$, where $f^{\circ}(x;\cdot)$ is the *Clarke directional derivative* of f [1, Chap. 6] and [5, Chap. 2].

In our main results, we assume that the domains of the upper subderivatives of f and g_i have sufficient intersection to allow us to apply the tangent cone "intersection theorems" of [1] and [23]. In describing what we mean by "sufficient intersection" in §3, we utilize the following definition.

DEFINITION 2.1. (a) *For $C \subset \mathbb{R}^p$, define*

$$\Delta^n C := \{(x_1, \ldots, x_n) \mid x_i \in C,\ x_1 = x_2 = \cdots = x_n\}.$$

(b) *Let $C_1, \ldots, C_n$ be nonempty convex cones in $\mathbb{R}^p$. These cones are said to be in* strong general position *if*

$$\Delta^n \mathbb{R}^p - \prod_{i=1}^{n} C_i = \mathbb{R}^{np}. \tag{2.2}$$

The concept of cones being in strong general position is discussed in [23] and [24], where several equivalent ways of writing (2.2) are presented. We mention here in particular that

$$\bigcap_{i=1}^{m} C_i - C_{m+1} = \mathbb{R}^p \qquad \forall m = 1, \ldots, n-1 \tag{2.3}$$

is equivalent to (2.2). Another equivalent of (2.2) is

$$\forall x_i \in \mathbb{R}^p,\ i = 1, \ldots, n, \quad \bigcap_{i=1}^{n} (C_i - x_i) \neq \emptyset,$$

which is often used in [1]. It is worth noting that (2.2) implies

$$\bigcap_{i=1}^{n} \operatorname{ri} C_i \neq \emptyset$$

and is implied by

$$\exists i \text{ such that } \quad C_i \cap \bigcap_{j \neq i} \operatorname{int} C_j \neq \emptyset.$$

In addition, observe that if $C_1, \ldots, C_n$ are in strong general position, then so are C_i, $i \in L$, for any nonempty $L \subset \{1, \ldots, n\}$.

We next mention a simple fact that will also help us in formulating the hypotheses of our main theorems in §3.

LEMMA 2.2. *Let $A : \mathbb{R}^n \to \mathbb{R}^p$ be linear, $C \subset \mathbb{R}^n$. The following are equivalent:*

(i) $A(C) = \mathbb{R}^p$,

(ii) $A(\mathbb{R}^n) = \mathbb{R}^p$ *and* $A^{-1}(0) - C = \mathbb{R}^n$.

Proof. Suppose that (i) holds. Then $A(\mathbb{R}^n) = \mathbb{R}^p$. Let $y \in \mathbb{R}^n$. By (i), there exists $z \in C$ such that $Az = -Ay$. Call $w = z + y$. Then $Aw = 0$, and $y = w - z \in A^{-1}(0) - C$. Hence (ii) holds.

On the other hand, suppose that (ii) is true, and let $y \in \mathbb{R}^p$. Then there exist $z \in \mathbb{R}^n$ and $x \in C$, $w \in A^{-1}(0)$, such that $Az = y$ and $z = x - w$. Hence $y = Ax - Aw = Ax \in A(C)$, and so (i) holds. □

Throughout this paper, we will assume that the equality constraint function h has the following differentiability property.

DEFINITION 2.3. *The function $h : \mathbb{R}^n \to \mathbb{R}^p$ is said to be* strictly differentiable *at $x \in \mathbb{R}^n$ if there exists a linear mapping $\nabla h(x) : \mathbb{R}^n \to \mathbb{R}^p$ such that for all $y \in \mathbb{R}^n$,*

$$\lim_{(w,v,t) \to (x,y,0^+)} (h(w + tv) - h(w))/t = \nabla h(x) y.$$

The properties of strictly differentiable functions are discussed in [5, §2.2]. In particular, strictly differentiable functions are locally Lipschitzian, and continuously differentiable functions are strictly differentiable.

We next present a technical lemma that roughly plays the role traditionally occupied by the implicit function theorem in discussions of differential stability. This lemma is a consequence of the following theorem, a special case of a versatile "inversion theorem" of Borwein [4, Thm. 2.1].

THEOREM 2.4. [4] *Let $S \subset \mathbb{R}^n$, and let $H : \mathbb{R}^n \to \mathbb{R}^m$ be strictly differentiable at $x_0 \in S \cap H^{-1}(0)$. Suppose that S is closed near x_0 and*

$$\nabla H(x_0)\, T(S, x_0) = \mathbb{R}^m. \tag{2.4}$$

Then there exist $L > 0$, $\delta > 0$ *such that for all* $x \in B(x_0, \delta) \cap S$ *and all* $u \in B(0, \delta)$,

$$\operatorname{dist}(x \mid H^{-1}(u) \cap S) \leq L \operatorname{dist}(H(x) \mid u). \tag{2.5}$$

LEMMA 2.5. *Posit the assumptions of Theorem* 2.4, *and let* $s \in \mathbb{R}^m$. *Then*

(a) $k(S, x_0) \cap \nabla H(x_0)^{-1}(s) = \{y \mid \forall \varepsilon > 0,\ \exists \lambda > 0$ *such that* $\forall t \in (0, \lambda),\ \exists y(t) \in B(y, \varepsilon)$ *with* $x_0 + ty(t) \in S \cap H^{-1}(ts)\}$.

(b) $T(S, x_0) \cap \nabla H(x_0)^{-1}(s) \subset \{y \mid \forall \varepsilon > 0, \exists \lambda > 0$ *such that* $\forall x \in B(x_0, \lambda) \cap S$, $\forall t \in (0, \lambda),\ \exists y(t) \in B(y, \varepsilon)$ *with* $x + ty(t) \in S \cap H^{-1}(H(x) + ts)\}$.

Proof. Denote the set on the right-hand side of (a) by Γ, and let $\varepsilon > 0$ be given. By Theorem 2.4, there exist $L > 0$, $\delta \in (0, \varepsilon)$ such that for all $x \in B(x_0, \delta)$ and $u \in B(0, \delta)$, (2.5) holds. To prove (a), let $y \in k(S, x_0)$ with $\nabla H(x_0)y = s$. Then there exists $\lambda > 0$ such that for all $t \in (0, \lambda)$, there exists $w(t) \in B(y, \delta/2)$ with $x_0 + tw(t) \in S \cap B(x_0, \delta)$, $ts \in B(0, \delta)$, and

$$\|H(x_0 + tw(t)) - H(x_0) - ts\|/t \leq \delta/4L.$$

Let $t \in (0, \lambda)$. Choose $w(t)$ as above. Then by (2.5), there exists $z(t) \in H^{-1}(ts) \cap S$ with

$$\begin{aligned} \|x_0 + tw(t) - z(t)\|/t &\leq L\|H(x_0 + tw(t)) - H(x_0) - ts\|/t + \varepsilon/4 \\ &\leq L\delta/4L + \varepsilon/4 < \varepsilon/2. \end{aligned}$$

Call $y(t) := (z(t) - x_0)/t$. Then $x_0 + ty(t) = z(t) \in S$, $H(x_0 + ty(t)) = ts$, and

$$\|y(t) - y\| \leq \|z(t) - x_0 - tw(t)\|/t + \|w(t) - y\| < \varepsilon.$$

Hence $y \in \Gamma$. The opposite inclusion in (a) is easy to prove.

To prove (b), let $y \in T(S, x_0)$ with $\nabla H(x_0)y = s$. Then there exists $\lambda > 0$ such that for all $t \in (0, \lambda)$ and $x \in B(x_0, \lambda) \cap S$, there exists $w(t) \in B(y, \delta/2)$ with

(i) $x + tw(t) \in S \cap B(x_0, \delta)$,

(ii) $H(x) + ts \in B(0, \delta)$ (since H is continuous at x_0),

(iii) $\|H(x + tw(t)) - H(x) - ts\|/t \leq \delta/4L$.

Let $t \in (0, \lambda)$, $x \in B(x_0, \lambda) \cap S$. Choose $w(t)$ satisfying (i), (iii). By (2.5), there exists $z(t) \in H^{-1}(H(x) + ts) \cap S$ such that

$$\begin{aligned} \|x + tw(t) - z(t)\|/t &\leq L\|H(x + tw(t)) - H(x) - ts\|/t + \varepsilon/4 \\ &\leq L\delta/4L + \varepsilon/4 < \varepsilon/2. \end{aligned}$$

Call $y(t) := (z(t) - x)/t$. Then $x + ty(t) = z(t) \in S$, $H(x + ty(t)) = H(x) + ts$, and

$$\|y(t) - y\| \leq \|z(t) - x - tw(t)\|/t + \|w(t) - y\| < \varepsilon.$$

Hence y is an element of the set on the right-hand side of (b). □

3. Bounds in primal form. In (P(s)), let $C(s) := \{x \in C \mid g_i(x) \leq s_i,\ i \in J,\ h(x) = s_{m+1}\}$ and $\Omega(s) := \{x \in C(s) \mid f(x) = v(s)\}$. Throughout this section and the next, we let $x_0 \in \Omega(0)$ and define $I(x_0) := \{i \in J \mid g_i(x_0) = 0\}$. We will often assume that x_0 satisfies the following conditions.

Assumption 3.1. (a) g_i is continuous at x_0 for all $i \in J \backslash I(x_0)$.

(b) h is strictly differentiable at x_0 and $\nabla h(x_0)\mathbb{R}^n = \mathbb{R}^p$.

(c) C is closed near x_0; f and g_i, $i \in I(x_0)$ are strictly LSC at x_0.

(d) $f^k(x_0; \cdot)$, $g_i^k(x_0; \cdot)$, $i \in I(x_0)$ are proper.

(e) dom $f^T(x_0;\cdot)$, dom $g_i^T(x_0;\cdot)$, $i \in I(x_0)$, $\nabla h(x_0)^{-1}(0)$, and $T(C,x_0)$ are in strong general position.

Remark 3.2. (a) We observe that Assumption 3.1(b) implies that $A(h^{-1}(0),x_0) = \nabla h(x_0)^{-1}(0)$ for all tangent cones A with $T \subset A \subset K$ by [23, Thm. 2.4, Cor. 6.15]. In addition, since Assumption 3.1(e) implies that $\nabla h(x_0)^{-1}(0) - T(C,x_0) = \mathbb{R}^n$ (by (2.3)), it follows from Lemma 2.2 that $\nabla h(x_0)\, T(C,x_0) = \mathbb{R}^p$.

(b) The conditions of Assumption 3.1 are satisfied in a wide variety of circumstances. For example, if f and each g_i are Lipschitzian near x_0 and C is closed, then (a), (c), and (d) hold. In addition, (b) and (e) reduce to $\nabla h(x_0)(T(C,x_0)) = \mathbb{R}^p$, since dom $f^T(x_0;\cdot) =$ dom $g_i^T(x_0;\cdot) = \mathbb{R}^n$ in this case.

Another important special case is that in which C is epi-Lipschitzian and f and g_i, $i \in I(x_0)$ are directionally Lipschitzian. (A set C is said to be *epi-Lipschitzian* at x_0 if the open tangent cone

$$IT(S,x) := \{y \mid \exists \lambda > 0 \text{ such that } \forall\, t \in (0,\lambda)\, \forall\, z \in S \cap B(x,\lambda)\, \forall\, v \in B(y,\lambda),\ z+tv \in S\}$$

is nonempty; and a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is said to be *directionally Lipschitzian* at x_0 if epi f is epi-Lipschitzian at $(x_0, f(x_0))$ [18, 19].) In this case, int dom $f^T(x_0;\cdot)$ and int dom $g_i^T(x_0;\cdot)$ are nonempty, and Assumption 3.1(e) will hold if

$$\text{int dom } f^T(x_0;\cdot) \cap \bigcap_{I(x_0)} \text{int dom } g_i^T(x_0;\cdot) \cap IT(C,x_0) \cap \nabla h(x_0)^{-1}(0) \neq \emptyset.$$

A thorough study of directionally Lipschitzian functions and epi-Lipschitzian sets is carried out in [19]; in particular, $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is directionally Lipschitzian at x_0 whenever f is LSC and dom $f^T(x_0;\cdot)$ is not contained in a subspace of $\mathbb{R}^n$ of dimension less than n.

It is also possible for Assumption 3.1 to be satisfied even if f and g_i are not locally Lipschitzian or directionally Lipschitzian. An example of LSC functions f_1 and f_2 for which $f_i^k(x_0;\cdot)$, $i = 1,2$ are proper and dom $f_1^T(x_0;\cdot)$, dom $f_2^T(x_0;\cdot)$ are in strong general position, but neither f_1 nor f_2 is directionally Lipschitzian at x_0, is given in [23, Remark 3.7(a)].

We now prove the first of our main results, which is an upper bound for the upper Dini derivative of v.

THEOREM 3.3. *Suppose that Assumption* 3.1 *is satisfied at* x_0 *and*

$$\text{(3.1)}\quad \text{dom } f^{k^\infty}(x_0;\cdot) \cap \nabla h(x_0)^{-1}(0) \cap k^\infty(C,x_0) \cap \bigcap_{I(x_0)} \{y \mid g_i^{k^\infty}(x_0;y) < 0\} \neq \emptyset.$$

Then for all $s \in \mathbb{R}^{m+p}$,

(3.2)

$$v^+(0;s) \leq \inf\{f^k(x_0;d) \mid g_i^k(x_0;d) \leq s_i,\ i \in I(x_0), \nabla h(x_0)d = s_{m+1},\ d \in k(C,x_0)\} < +\infty.$$

Proof. Let $s \in \mathbb{R}^{m+p}$, and denote the expression on the right-hand side of (3.2) by $\varphi(s)$. We first show that $\varphi(s) < +\infty$. By (b) and (e) of Assumption 3.1 and the fact that $T \subset k$, there exists $z \in \operatorname{dom} f^k(x_0;\cdot) \cap k(C,x_0) \cap \bigcap_{I(x_0)} \operatorname{dom} g_i^k(x_0;\cdot)$ such that $\nabla h(x_0)z = s_{m+1}$. Let y be an element of the set in (3.1). Then for $t > 0$ large enough, we have for all $i \in I(x_0)$ that $g_i^k(x_0; z+ty) \leq s_i$ as well as

$$\nabla h(x_0)(z+ty) = s_{m+1} \quad \text{and} \quad z+ty \in \operatorname{dom} f^k(x_0;\cdot) \cap k(C,x_0).$$

Hence $\varphi(s) < +\infty$.

To demonstrate that $v^+(0;s) \leq \varphi(s)$, suppose that $d \in k(C,x_0) \cap \nabla h(x_0)^{-1}(s_{m+1})$ with $g_i^k(x_0;d) \leq s_i$, $i \in I(x_0)$, and $f^k(x_0;d) \leq r$. Let $\varepsilon > 0$ be given, let y be an element of the set in (3.1), and suppose that $\alpha \geq f^{k^\infty}(x_0;y)$. Choose $\eta > 0$ such that $\max\{\|\eta y\|, |\eta\alpha|\} < \varepsilon/2$. Then $d+\eta y \in k(C,x_0) \cap \nabla h(x_0)^{-1}(s_{m+1})$ and $f^k(x_0; d+\eta y) \leq r + \eta\alpha$. In addition, $g_i^k(x_0; d+\eta y) < s_i$ for $i \in I(x_0)$, so there exists $\delta \in (0, \varepsilon/2)$ with $g_i^k(x_0; d+\eta y) \leq s_i - \delta$ for all $i \in I(x_0)$.

Now for $x \in \mathbb{R}^n$, $(r_0, \ldots, r_m) \in \mathbb{R}^{m+1}$, define

$$D_0 := \{(x, r_0, \ldots, r_m) \mid f(x) \leq r_0\},$$

$$D_i := \{(x, r_0, \ldots, r_m) \mid g_i(x) \leq r_i\},\ i \in J,$$

$$D_{m+1} := \{(x, r_0, \ldots, r_m) \mid x \in C\},$$

and call $S := \bigcap_{i=0}^{m+1} D_i$, $z_0 := (x_0, f(x_0), 0, \ldots, 0)$. By Assumption 3.1(a), $k(D_i, z_0) = \mathbb{R}^{n+m+1}$ for each $i \in J \backslash I(x_0)$. Taking into account (2.1), we obtain

$$(d+\eta y,\ r+\eta\alpha,\ s_1-\delta, \ldots, s_m-\delta) \in \bigcap_{i=0}^{m+1} k(D_i, z_0).$$

We observe that Assumption 3.1(e) implies that $T(D_i, z_0)$, $i = 0, \ldots, m+1$ are in strong general position. By [1, Cor. 4.3.6], it follows that $(d+\eta y,\ r+\eta\alpha,\ s_1-\delta, \ldots, s_m-\delta) \in k(S, z_0)$.

Next, define $H : \mathbb{R}^{n+m+1} \to \mathbb{R}^p$ by $H(x, r_0, \ldots, r_m) := h(x)$. Then $(d+\eta y,\ r+\eta\alpha,\ s_1-\delta,\ s_m-\delta) \in k(S, z_0) \cap \nabla H(z_0)^{-1}(s_{m+1})$. By Assumption 3.1(e),

$$\nabla H(z_0)^{-1}(0) - \bigcap_{i=0}^{m+1} T(D_i, z_0) = \mathbb{R}^{n+m+1},$$

and so it follows from Lemma 2.2 that $\nabla H(z_0)(\bigcap_{i=0}^{m+1} T(D_i, z_0)) = \mathbb{R}^p$. Again applying [1, Cor. 4.3.6], we conclude that $\nabla H(z_0)\, T(S, z_0) = \mathbb{R}^p$. By Lemma 2.5(a), there exists $\lambda > 0$ such that for all $t \in (0, \lambda)$, there exists $d(t) \in B(d+\eta y, \delta) \subset B(d, \varepsilon)$ satisfying the following conditions:

(i) $\quad (f(x_0 + td(t)) - f(x_0))/t \leq r + \eta\alpha + \delta,$

(ii) $\quad g_i(x_0 + td(t)) \leq ts_i \ \forall i \in J,$

(iii) $\quad x_0 + td(t) \in C,$

(iv) $\quad h(x_0 + td(t)) = ts_{m+1}.$

Finally, since (ii)–(iv) imply that $x_0 + td(t) \in C(ts)$, we have by (i) that for all $t \in (0, \lambda)$,

$$(v(ts) - v(0))/t \leq (f(x_0 + td(t)) - f(x_0))/t \leq r + \eta\alpha + \varepsilon/2 \leq r + \varepsilon.$$

Since ε is arbitrary, we conclude that $v^+(0;s) \le r$, and (3.2) holds. □

It is also possible to obtain a lower bound on $v_+(0;s)$ in terms of the Clarke tangent cone of C and the upper subderivatives of f and g_i. For this bound, we will assume a Mangasarian–Fromovitz-type constraint qualification along the lines of (3.1) along with a "tameness" condition as in [10, 18, 22].

DEFINITION 3.4. *Problem* $(P(0))$ *is said to be* tame *if* $v(0)$ *is finite and for any sequence* $\{s_j\} \to 0$ *in* $\mathbb{R}^{m+p}$, *there exists* $\{x_j\} \to x$ *for some* $x \in \Omega(0)$ *with* $x_j \in \Omega(s_j)$ *for* j *sufficiently large.*

THEOREM 3.5. *Suppose that* $(P(0))$ *is tame. At each* $x_0 \in \Omega(0)$, *assume that Assumption* 3.1 *is satisfied and*

$$\text{(3.3)} \qquad \operatorname{dom} f^T(x_0;\cdot) \cap \nabla h(x_0)^{-1}(0) \cap T(C,x_0) \cap \bigcap_{I(x_0)} \{y \mid g_i^T(x_0;y) < 0\} \neq \emptyset.$$

Then, for each $s \in \mathbb{R}^{m+p}$, *there exists* $x \in \Omega(0)$ *such that*

$$\text{(3.4)} \qquad \begin{aligned} v_+(0;s) \ge -\inf\{\, & f^T(x;d) \mid g_i^T(x;d) \le -s_i,\ i \in I(x), \\ & \nabla h(x)d = -s_{m+1},\ d \in T(C,x)\} > -\infty. \end{aligned}$$

Proof. Let $s \in \mathbb{R}^{m+p}$. Choose $\{t_j\} \to 0^+$ such that

$$v_+(0;s) = \lim_{j\to\infty} (v(t_j s) - v(0))/t_j.$$

Since $(P(0))$ is tame, there exists $x \in \Omega(0)$ and $\{x_j\} \to x$ with $x_j \in \Omega(t_j s)$ for j large enough. Then

$$v_+(0;s) = \lim_{j\to\infty} (f(x_j) - f(x))/t_j.$$

Let $\varepsilon > 0$, $\eta > 0$ be given, and denote the right-hand side of (3.4) by $\varphi(s)$. As in the proof of Theorem 3.3, Assumption 3.1 and (3.3) imply that $\varphi(s) > -\infty$. Suppose that $\varphi(s)$ is finite. Then there exists $z \in T(C,x)$ with $\nabla h(x)z = -s_{m+1}$, $g_i^T(x;z) \le -s_i$ for all $i \in I(x)$, and $f^T(x;z) \le -\varphi(s) + \varepsilon/2$. By (3.3), there exists $y \in \operatorname{dom} f^T(x;\cdot) \cap T(C,x) \cap \nabla h(x)^{-1}(0)$ with $g_i^T(x;y) < 0$ for all $i \in I(x)$. Set $d := z + \eta y$.

Next, define H and the sets S and D_i, $i = 0,\dots,m+1$ as in the proof of Theorem 3.3. Call $z_0 := (x, f(x), 0, \dots, 0)$, $\beta_0 := f^T(x;d)$, $\beta_i := g_i^T(x;d)$ for $i \in I(x)$, and let $\beta_i \in \mathbb{R}$ for $i \in J\backslash I(x)$. Then $(d,\beta_0,\dots,\beta_m) \in \bigcap_{i=0}^{m+1} T(D_i, z_0)$. As in the proof of Theorem 3.3, we may apply [1, Cor. 4.3.6] to conclude that $(d,\beta_0,\dots,\beta_m) \in T(S,z_0) \cap \nabla H(z_0)^{-1}(-s_{m+1})$. Hence by Lemma 2.5(b), there exists $(d_j,\beta_{0j},\dots,\beta_{mj}) \to (d,\beta_0,\dots,\beta_m)$ such that $(x_j, f(x_j), g_1(x_j),\dots,g_m(x_j)) + t_j(d_j,\beta_{0j},\dots,\beta_{mj}) \in S$ and

(i) $$h(x_j + t_j d_j) = h(x_j) - t_j\, s_{m+1} = 0.$$

It follows that for j large enough, we have

(ii) $$(f(x_j + t_j d_j) - f(x_j))/t_j \le \beta_{0j} \le \beta_0 + \varepsilon/2,$$

(iii) $$\begin{aligned} (g_i(x_j + t_j d_j) - g_i(x_j))/t_j &\le \beta_{ij} \\ &\le \beta_i + (-s_i - \beta_i)\,\forall\, i \in I(x) \end{aligned}$$

(iv) $$x_j + t_j d_j \in C,$$

(v) $$g_i(x_j + t_j d_j) < 0 \quad \forall\, i \in J\backslash I(x).$$

Now (i), (iii), (iv), and (v) imply that $x_j + t_j d_j \in C(0)$ for j sufficiently large, and so

$f(x_j + t_j d_j) \geq f(x)$. Hence by (ii) and the fact that $f^T(x;\cdot)$ is sublinear,

$$\begin{aligned}(f(x) - f(x_j))/t_j &\leq (f(x_j + t_j d_j) - f(x_j))/t_j \\ &\leq \beta_0 + \varepsilon/2 \\ &\leq f^T(x;z) + \eta f^T(x;y) + \varepsilon/2 \\ &\leq -\varphi(s) + \eta f^T(x;y) + \varepsilon.\end{aligned}$$

Therefore, since ε and η are arbitrary, $v_+(0;s) \geq \varphi(s)$ if $\varphi(s)$ is finite.

Finally, suppose that $\varphi(s) = +\infty$, and let $r \in \mathbb{R}$. Then there exists $z \in T(C,x)$ with $\nabla h(x)z = -s_{m+1}$, $g_i^T(x;z) \leq -s_i$ for all $i \in I(x)$, and $f^T(x;z) \leq -r$. Arguing as above, we can show that $v_+(0;s) \geq r$. Since r is arbitrary, we conclude that $v_+(0;s) = +\infty$, and again $v_+(0;s) \geq \varphi(s)$. □

4. Lagrange multipliers and bounds in dual form. The bounds in Theorems 3.3 and 3.5 can also be expressed in "dual form" in terms of appropriate Lagrange multipliers for $(P(0))$. To define these multipliers, we will need the concept of the subdifferential associated with a tangent cone.

DEFINITION 4.1. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be finite at $x \in \mathbb{R}^n$, and let A be a tangent cone. The* A subdifferential *of f at x is the set*

$$\partial^A f(x) := \{x^* \in \mathbb{R}^n \mid \langle x^*, y\rangle \leq f^A(x;y) \text{ for all } y \in \mathbb{R}^n\}.$$

We will be primarily interested in A subdifferentials for convex tangent cones A. If A is convex and $f^A(x;\cdot)$ is proper, then $\partial^A f(x) = \partial(f^A(x;\cdot))(0)$, where $\partial(f^A(x;\cdot))(0)$ is the ordinary subdifferential of the convex function $f^A(x;\cdot)$ at 0. In discussing Lagrange multipliers, we need to consider expressions of the form $\partial(\lambda f^A(x;\cdot))(0)$ for $\lambda \geq 0$. If $\lambda > 0$, then $\partial(\lambda f^A(x;\cdot))(0) = \lambda \partial^A f(x)$. If $\lambda = 0$, we follow the convention that $\lambda f^A(x;y) = i_{\operatorname{dom} f^A(x;\cdot)}(y)$, so that if $f^A(x;\cdot)$ is proper, then $\partial(\lambda f^A(x;\cdot))(0) = (\partial^A f(x))^\infty$.

For a tangent cone A and $x_0 \in \Omega(0)$, we define the multiplier set

$$M(A,x_0) := \Big\{(\lambda_1,\ldots,\lambda_m,\mu) \in \mathbb{R}^{m+p} \mid \lambda_i \geq 0,\ \lambda_i = 0\ \forall i \in J\backslash I(x_0),$$

$$0 \in \partial^A f(x_0) + \sum_{I(x_0)} \lambda_i \partial^A g_i(x_0) + \mu^t \nabla h(x_0) + (A(C,x_0))^\circ\Big\},$$

where if $\lambda_i = 0$ for some $i \in I(x_0)$, $\lambda_i \partial^A g_i(x_0)$ is to be interpreted as $(\partial^A g_i(x_0))^\infty$. We next prove a KKT theorem which shows that $M(A,x_0)$ is nonempty if $A \subset k$ is a closed convex tangent cone and the Mangasarian–Fromovitz-type constraint qualification (3.1) is satisfied. In the proof, we utilize the fact that if A and D are tangent cones and $D \subset A$, then $f^A(x;\cdot) \leq f^D(x;\cdot)$ and $\partial^A f(x) \subset \partial^D f(x)$.

THEOREM 4.2. *Let $x_0 \in \Omega(0)$ satisfy Assumption* 3.1, *and suppose that $A \subset k$ is a closed convex tangent cone. Then there exist $\lambda_i \geq 0$, $i \in \{0\} \cup I(x_0)$, not all equal zero, and $\mu \in \mathbb{R}^p$, such that*

$$0 \in \lambda_0 \partial^A f(x_0) + \sum_{I(x_0)} \lambda_i \partial^A g_i(x_0) + \mu^t \nabla h(x_0) + (A(C,x_0))^\circ. \tag{4.1}$$

If in addition (3.1) *holds, then there exist $\lambda_i \geq 0$, $i \in I(x_0)$, and $\mu \in \mathbb{R}^p$ such that*

$$0 \in \partial^A f(x_0) + \sum_{I(x_0)} \lambda_i \partial^A g_i(x_0) + \mu^t \nabla h(x_0) + (A(C,x_0))^\circ. \tag{4.2}$$

Proof. For $S \subset \mathbb{R}^n$, $x \in S$, define $D(S,x) := \operatorname{cl}(A(S,x) + k^\infty(S,x))$. Observe that D is a nonempty, closed convex tangent cone with $A \subset D$ and $k^\infty \subset D$. In addition, we have $D \subset k$, since

$$D(S,x) \subset \operatorname{cl}(k(S,x) + k^\infty(S,x)) = \operatorname{cl} k(S,x) = k(S,x) \quad \forall\, S \subset \mathbb{R}^n,\ x \in S.$$

Next, let $\gamma := h^{-1}(0) \cap C$, and define

$$F(x) := \max\{f(x) + i_\gamma(x) - f(x_0),\ \max_{1 \le i \le m} g_i(x)\}.$$

Since $x_0 \in \Omega(0)$, x_0 is a minimizer of F. So for all $y \in \mathbb{R}^n$,

$$\begin{aligned} 0 \le F^k(x_0;y) &\le \max\{(f + i_\gamma)^k(x_0;y),\ \max_{I(x_0)} g_i^k(x_0;y)\} \\ &\quad \text{(by Assumption 3.1(e) and [1, Thm. 6.1.9, Prop. 6.3.2])} \\ &\le \max\{f^k(x_0;y) + i_{\nabla h(x_0)^{-1}(0) \cap k(C,x_0)}(y),\ \max_{I(x_0)} g_i^k(x_0;y)\} \\ &\quad \text{(by Assumption 3.1(b), (e) and [23, Cor. 6.15, 6.16])} \\ &\le \max\{f^D(x_0;y) + i_\Gamma(y),\ \max_{I(x_0)} g_i^D(x_0;y)\}, \end{aligned}$$

where $\Gamma := \nabla h(x_0)^{-1}(0) \cap D(C,x_0)$.

Now let $q := |I(x_0)| + 1$, and define $S_1 := \{z \in \mathbb{R}^q \mid z \le 0\}$, $S_2 := \{z \in \mathbb{R}^q \mid \exists y \in \mathbb{R}^n$ with $(f^D(x_0;y) + i_\Gamma(y), \dots, g_i^D(x_0;y) \dots) \le z\}$, where $\le$ denotes the coordinatewise ordering on $\mathbb{R}^q$. Note that S_1 and S_2 are convex sets with $\operatorname{int} S_1 \cap S_2 \ne \emptyset$. By a standard separation theorem and Assumption 3.1(d), there exist $\lambda_i \in \mathbb{R}$, $i \in \{0\} \cup I(x_0)$, not all equal zero, such that for all $z = (z_i) \in S_1$ and $y \in \mathbb{R}^n$,

$$\sum \lambda_i z_i \le 0 \le \lambda_0(f^D(x_0;y) + i_\Gamma(y)) + \sum_{I(x_0)} \lambda_i g_i^D(x_0;y). \tag{4.3}$$

By Assumption 3.1(e), $\operatorname{dom} f^D(x_0;\cdot) \cap \Gamma \cap \bigcap_{I(x_0)} \operatorname{dom} g_i^D(x_0;\cdot) \ne \emptyset$. If follows from the first inequality in (4.3) that each $\lambda_i \ge 0$. The second inequality in (4.3) implies that

$$0 \in \partial\left(\lambda_0 f^D(x_0;\cdot) + \sum_{I(x_0)} \lambda_i g_i^D(x_0;\cdot) + i_\Gamma(\cdot)\right)(0).$$

Since $T \subset D$, Assumption 3.1(e) implies that $\operatorname{dom} f^D(x_0;\cdot)$, Γ, and $\operatorname{dom} g_i^D(x_0;\cdot)$, $i \in I(x_0)$ are in strong general position. By the calculus of subdifferentials [15, Thm. 23.8, Cor. 16.4.2, 16.3.2], there exists $\mu \in \mathbb{R}^p$ such that

$$\begin{aligned} 0 &\in \lambda_0 \partial^D f(x_0) + \sum_{I(x_0)} \lambda_i \partial^D g_i(x_0) + \mu^t \nabla h(x_0) + (D(C,x_0))^\circ, \\ &\subset \lambda_0 \partial^A f(x_0) + \sum_{I(x_0)} \lambda_i \partial^A g_i(x_0) + \mu^t \nabla h(x_0) + (A(C,x_0))^\circ. \end{aligned}$$

Thus (4.1) holds. Finally, suppose that (3.1) holds and $\lambda_0 = 0$ in (4.3). Let y be an element of the set in (3.1). Since $k^\infty \subset D$, (4.3) implies that $0 \le \sum_{I(x_0)} \lambda_i g_i^{k^\infty}(x_0;y) < 0$, a contradiction. Hence $\lambda_0 > 0$, and (4.2) holds. □

Theorem 4.2 shows that for any closed convex tangent cone $A \subset k$, the generalized KKT inclusion (4.2) holds under a constraint qualification (3.1) that is independent of A. Tangent

cones A that can be used in (4.1) and (4.2) include k^∞, T, and the prototangent cone of Penot [14]. Theorem 4.2 for $A := T$ sharpens [23, Thm. 4.7], in which (3.3) is used instead of (3.1). For an example in which (3.1) is satisfied but (3.3) is not, see [12, Ex. 4.4].

We can also give conditions, as in [13], which guarantee that $M(A, x_0)$ is closed and bounded.

PROPOSITION 4.3. *Suppose that Assumption* 3.1 *and* (3.1) *are satisfied at* $x_0 \in \Omega(0)$, *and let* $A \subset k$ *be a closed convex tangent cone. Assume that* $\operatorname{dom} f^A(x_0;\cdot)$, $A(C, x_0)$, $\operatorname{dom} g_i^A(x_0;\cdot)$, $i \in I(x_0)$, *and* $\nabla h(x_0)^{-1}(0)$ *are in strong general position, and that*

$$\operatorname{dom} f^A(x_0;\cdot) \cap \nabla h(x_0)^{-1}(0) \cap A(C, x_0) \cap \bigcap_{I(x_0)} \{y \mid g_i^A(x_0; y) < 0\} \neq \emptyset. \tag{4.4}$$

Then $M(A, x_0)$ *is nonempty, closed, and bounded.*

Proof. The fact that $M(A, x_0) \neq \emptyset$ follows from Theorem 4.2. One can easily show, just as in [22, Prop. 2.2], that $M(A, x_0)$ is closed. To show that $M(A, x_0)$ is bounded, let $(\lambda_1, \ldots, \lambda_m, \mu) \in M(A, x_0)$. Then with $\mu = (\mu_1, \ldots, \mu_p)$ and $h(x) = (h_1(x), \ldots, h_p(x))$, we have for all $y \in \mathbb{R}^n$ that

$$0 \leq f^A(x_0; y) + \sum_{I(x_0)} \lambda_i g_i^A(x_0; y) + \sum_{i=1}^{p} \mu_i \nabla h_i(x_0) y + i_{A(C,x_0)}(y).$$

Let $L := \{i \mid \mu_i > 0\}$. By our general position hypothesis and Lemma 2.2, there exists $y_L \in \operatorname{dom} f^A(x_0;\cdot) \cap \bigcap_{I(x_0)} \operatorname{dom} g_i^A(x_0;\cdot) \cap A(C, x_0)$ such that $\nabla h_i(x_0) y_L = -1$ for $i \in L$ and $\nabla h_i(x_0) y_L = 1$ for $i \notin L$. Choose an element y_0 in the set in (4.4). Then there exist $\alpha_L > 0$ and t sufficiently small in $(0, 1)$ such that for all $i \in I(x_0)$,

$$g_i^A(x_0; ty_L + (1-t)y_0) \leq t g_i^A(x_0; y_L) + (1-t) g_i^A(x_0; y_0) \leq -\alpha_L.$$

Call $y := ty_L + (1-t)y_0$. Since $\nabla h(x_0) y_0 = 0$, we have $\nabla h_i(x_0) y = t \nabla h_i(x_0) y_L$, and so $\mu_i \nabla h_i(x_0) y = -t \mid \mu_i \mid$ for $1 \leq i \leq p$. Since A is convex, $y \in A(C, x_0) \cap \operatorname{dom} f^A(x_0;\cdot)$. Hence

$$\begin{aligned} 0 &\leq f^A(x_0; y) + \sum_{I(x_0)} \lambda_i g_i^A(x_0; y) + \sum_{i=1}^{P} \mu_i \nabla h_i(x_0) y \\ &\leq f^A(x_0; y) - \alpha_L \sum_{I(x_0)} \lambda_i - t \sum_{i=1}^{p} \mid \mu_i \mid, \end{aligned}$$

and so

$$\sum_{I(x_0)} \lambda_i + \sum_{i=1}^{p} \mid \mu_i \mid \leq f^A(x_0; y) / \min\{\alpha_L, t\}.$$

Since there are only a finite number of possible choices for L, it follows that $M(A, x_0)$ is bounded. □

We now present the main results of this section, which are bounds for $v^+(0; s)$ and $v_+(0; s)$ in dual form.

THEOREM 4.4. (a) *Suppose that Assumption* 3.1 *and* (3.1) *are satisfied at* $x_0 \in \Omega(0)$. *Then for all* $s \in \mathbb{R}^{m+p}$,

$$v^+(0; s) \leq -\min\{\langle (\lambda, \mu), s \rangle \mid (\lambda, \mu) \in M(k^\infty, x_0)\} < +\infty. \tag{4.5}$$

(b) *Suppose that* $(P(0))$ *is tame and Assumption* 3.1 *and* (3.3) *are satisfied at each* $x_0 \in \Omega(0)$. *Then for each* $s \in \mathbb{R}^{m+p}$, *there exists* $x \in \Omega(0)$ *such that*

$$v_+(0;s) \geq -\max\{\langle(\lambda,\mu),s\rangle \mid (\lambda,\mu) \in M(T,x)\} > -\infty. \tag{4.6}$$

Proof. Let $s \in \mathbb{R}^{m+p}$, and consider the convex program

(P1)

$$\alpha := \inf\{f^{k^\infty}(x_0;d) \mid g_i^{k^\infty}(x_0;d) \leq s_i,\ i \in I(x_0),\ \nabla h(x_0)d = s_{m+1},\ d \in k^\infty(C,x_0)\}.$$

By Theorem 3.3, $v^+(0;s) \leq \alpha$. As in the proof of Theorem 3.3, we can use Assumption 3.1(e) and (3.1) to show that $\alpha < +\infty$. For $(\lambda,\mu) \in \mathbb{R}^{m+p}$ and $y \in \mathbb{R}^n$, define the Lagrangian function

$$L(\lambda,\mu,y) := f^{k^\infty}(x_0;y) + \sum_{I(x_0)} \lambda_i g_i^{k^\infty}(x_0;y) + \mu^t \nabla h(x_0) y.$$

In terms of this Lagrangian, the Lagrangian dual of (P1) is

$$\beta := \sup_{\substack{\mu \in \mathbb{R}^p \\ \lambda_i \geq 0}} \inf_{y \in k^\infty(C,x_0)} L(\lambda,\mu,y) - \sum_{I(x_0)} \lambda_i s_i - \mu^t s_{m+1}. \tag{D1}$$

If $(\lambda,\mu) \in M(k^\infty,x_0)$, then $0 \leq L(\lambda,\mu,y)$ for all $y \in k^\infty(C,x_0)$, and

$$\inf_{y \in k^\infty(C,x_0)} L(\lambda,\mu,y) = 0.$$

On the other hand, if $(\lambda,\mu) \notin M(k^\infty,x_0)$, then $L(\lambda,\mu,y) < 0$ for some $y \in k^\infty(C,x_0)$, and so $L(\lambda,\mu,ty) \to -\infty$ as $t \to +\infty$. Hence $\beta = \sup\{-\langle(\lambda,\mu),s\rangle \mid (\lambda,\mu) \in M(k^\infty,x_0)\}$, and β is finite by Proposition 4.3.

Now let $d \in \operatorname{dom} f^{k^\infty}(x_0;\cdot)$ be feasible for (P1), and let y be an element of the set in (3.1). Then for all $t > 0$, $d + ty \in k^\infty(C,x_0) \cap \operatorname{dom} f^{k^\infty}(x_0;\cdot)$, $g_i^{k^\infty}(x_0;d+ty) < s_i$ for all $i \in I(x_0)$, and $\nabla h(x_0)(d+ty) = s_{m+1}$. In addition, $\nabla h(x_0)k^\infty(C,x_0) = \mathbb{R}^p$ by Assumption 3.1 and Lemma 2.2, and so a constraint qualification is satisfied for (P1). By Lagrangian duality (see, for example, [3, Thm. 6.2.4]), $\alpha = \beta$ and β is attained for some $(\lambda,\mu) \in M(k^\infty,x_0)$. Therefore (4.5) holds. The proof of (4.6) is similar. □

The upper bound in (4.5) sharpens the bound given in [2, Thm. 3.2] as well as extending it beyond the locally Lipschitzian case. This is demonstrated by [22, Ex. 3.1]. The lower bound in (4.6) extends [2, Thm. 3.4] beyond the locally Lipschitzian case. It also improves the result given in [22, Thm. 4.1] by adding equality constraints to (P(s)) and replacing assumption (17) of [22] with the weaker hypothesis Assumption 3.1(e).

We can combine parts (a) and (b) of Theorem 4.4 to give conditions guaranteeing the existence of

$$v'(0;s) = \lim_{t \to 0^+} (v(ts) - v(0))/t.$$

COROLLARY 4.5. *Suppose that* $(P(0))$ *is tame, that* (3.3) *and Assumption* 3.1 *are satisfied at each* $x_0 \in \Omega(0)$, *and that* $M(T,x_0)$ *is a singleton at each* $x_0 \in \Omega(0)$. *Then for each* $s \in \mathbb{R}^{m+p}$, $v'(0;s)$ *exists and there exists* $x \in \Omega(0)$ *with* $M(T,x) = \{(\lambda,\mu)\}$ *and*

$$v'(0;s) = -\langle(\lambda,\mu),s\rangle. \tag{4.7}$$

Proof. By Theorem 4.4, there exists $x \in \Omega(0)$ such that (4.6) holds. By hypothesis, $M(T,x)$ is a singleton; say $M(T,x) = \{(\lambda,\mu)\}$. Since $T \subset k^\infty$ and (3.3) holds at x, (3.1) also holds at x. By Theorem 4.2, $M(k^\infty,x)$ is nonempty, and since $M(k^\infty,x) \subset M(T,x)$, it follows that $M(k^\infty,x) = \{(\lambda,\mu)\}$. By (4.5) and (4.6), we have for each $s \in \mathbb{R}^{m+p}$ that

$$-\langle(\lambda,\mu),s\rangle \le v_+(0;s) \le v^+(0;s) \le -\langle(\lambda,\mu),s\rangle,$$

and so (P1) holds. □

In many nonsmooth programs, $M(T,x)$ will contain more than one element, and Corollary 4.5 will not be applicable. However, it is easy to construct examples in which some nonsmoothness is present and Corollary 4.5 can be used. We illustrate below with a simple example for which Corollary 4.5 is relevant but the corresponding results of [2, 5, 9, 22] do not apply.

Example 4.6. In (P(s)), let $C = \mathbb{R}^3$ and $J = \{1\}$, $p = 1$. Define $f : \mathbb{R}^3 \to \mathbb{R}$ by $f(x,y,z) = |x|^{1/2} - y + z$, $g_1 : \mathbb{R}^3 \to \mathbb{R}$ by $g_1(x,y,z) = y$, and $h : \mathbb{R}^3 \to \mathbb{R}$ by $h(x,y,z) = z$. It is easy to calculate that $\Omega(0) = \{(0,0,0)\}$, and $(P(0))$ is tame. Here f is not Lipschitzian near $x_0 := (0,0,0)$, and $\operatorname{dom} f^T(x_0;\cdot) = \{0\}\times\mathbb{R}\times\mathbb{R}$, $\operatorname{dom} g_1^T(x_0;\cdot) = \mathbb{R}^3$, and $\nabla h(x_0)^{-1}(0) = \mathbb{R}\times\mathbb{R}\times\{0\}$. Thus Assumption 3.1 holds, and the direction $(0,-1,0)$ belongs to the set in (3.3). Finally, $M(k^\infty,x_0) = M(T,x_0) = \{(1,-1)\}$, and so for $s = (s_1,s_2)$, $v'(0;s) = -s_1 + s_2$ by Corollary 4.5.

5. Conclusion. Using the calculus of tangent cones, we have extended the Gauvin–Tolle bounds on the upper and lower Dini directional derivatives of the marginal function v to a nonsmooth setting. Since there is a corresponding calculus for second-order tangent sets [1], the methods of this paper should also be applicable in establishing bounds on second-order directional derivatives of v. We will pursue this in a forthcoming paper.

A comparison of this paper and [5, §6.5] suggests an interesting question: Is it possible to provide KKT conditions and bounds on $v^+(0;s)$ and $v_+(0;s)$ for programs with locally Lipschitizian equality constraint functions and non-Lipschitzian objective and inequality constraint functions?

Acknowledgments. The author would like to thank Professor Marcin Studniarski for helpful discussions on the subject of this paper. He is also grateful for the perceptive comments of a referee.

REFERENCES

[1] J.-P. Aubin and H. Frankowska, *Set-Valued Analysis*, Birkhäuser Boston, Cambridge, MA, 1990.
[2] A. Auslender, *Differentiable stability in non-convex and non-differentiable programming*, Math. Programming Stud., 10 (1979), pp. 29–41.
[3] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley, New York, 1979.
[4] J. M. Borwein, *Stability and regular points of inequality systems*, J. Optim. Theory Appl., 48 (1986), pp. 9–52.
[5] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.
[6] K.-H. Elster and J. Thierfelder, *Abstract cone approximations and generalized differentiability in nonsmooth optimization*, Optimization, 19 (1988), pp. 315–341.
[7] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, New York, 1983.
[8] J. Gauvin and R. Janin, *Directional behaviour of optimal solutions in nonlinear mathematical programming*, Math. Oper. Res., 13 (1988), pp. 629–649.
[9] J. Gauvin and J. M. Tolle, *Differential stability in nonlinear programming*, SIAM J. Control Optim., 15 (1977), pp. 294–311.
[10] B. Gollan, *On the marginal function in nonlinear programming*, Math. Oper. Res., 9 (1984), pp. 208–221.
[11] J.-B. Hiriart-Urruty, *On optimality conditions in nondifferentiable programming*, Math. Programming, 14 (1978), pp. 73–86.

[12] R. R. MERKOVSKY AND D. E. WARD, *General constraint qualifications in nondifferentiable programming*, Math. Programming, 47 (1990), pp. 389–405.
[13] V. H. NGUYEN, J.-J. STRODIOT, AND R. MIFFLIN, *On conditions to have bounded multipliers in locally Lipschitz programming*, Math. Programming, 18 (1980), pp. 100–106.
[14] J.-P. PENOT, *Variations on the theme of nonsmooth analysis: another subdifferential*, in Nondifferential Optimization: Motivations and Applications, V. F. Demyanov and D. Pallaschke, eds., Springer-Verlag, Berlin, 1985, pp. 41–54.
[15] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[16] ———, *Directionally Lipschitzian function and subdifferential calculus*, Proc. London Math. Soc., 39 (1979), pp. 331–355.
[17] ———, *Generalized directional derivatives and subgradients of nonconvex functions*, Canad. J. Math., 32 (1980), 257–280.
[18] ———, *Lagrange multipliers and subderivatives of optimal value functions in nonlinear programming*, Math. Programming Stud., 17 (1982), pp. 28–66.
[19] ———, *Directional differentiability of the optimal value function in a nonlinear programming problem*, Math. Programming Stud., 21, (1984), pp. 213–226.
[20] A. SEEGER, *Second order directional derivatives in parametric optimization problems*, Math. Oper. Res., 13 (1988), pp. 124–139.
[21] C. URSESCU, *Tangent sets' calculus and necessary conditions for extremality*, SIAM J. Control Optim., 20 (1982), pp. 563–574.
[22] D. E. WARD, *Differential stability in non-Lipschitzian optimization*, J. Optim. Theory Appl., 73 (1992), pp. 101–120.
[23] D. E. WARD AND J. M. BORWEIN, *Nonsmooth calculus in finite dimensions*, SIAM J. Control Optim., 25 (1987), pp. 1312–1340.
[24] C. ZALINESCU, *On convex sets in general position*, Linear Algebra Appl., 64 (1985), pp. 191–198.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 212–226, February 1996
© 1996 Society for Industrial and Applied Mathematics
012

A GENERALIZED CONVEXITY AND VARIATIONAL INEQUALITY FOR QUASI-CONVEX MINIMIZATION*

PHAN THIEN THACH† AND MASAKAZU KOJIMA‡

Abstract. We present a variational-type necessary and sufficient condition for the optimality in a quasi-convex minimization, in which a vanishing differential is not a sufficient condition. Using this condition we show that path-following methods can, in principle, be applied to solving a quasi-convex minimization. Also, we give a connection between an optimality condition and a duality scheme using a minimax principle.

Key words. generalized convexity, variational inequality, quasi-convex minimization

AMS subject classifications. 49N15, 49J52, 90C26

1. Introduction.

It is well known that optimality criteria in mathematical programming problems can be formulated in the forms of variational inequalities, based on which the interconnection between various areas was established. In unconstrained convex programs a vanishing subdifferential is necessary and sufficient for the optimality. In constrained convex programs, this principle appears in the Karush–Kuhn–Tucker (KKT) condition. A fundamental hypothesis behind this condition is convexity. A convex function has two important properties: the lower level sets are convex and local minima are global (briefly, L = G). One can extend the KKT condition to pseudoconvex programs in which these properties are satisfied (e.g., Tuy [45], Mangasarian [19, 20]). In quasi-convex programs a vanishing differential is not sufficient for the optimality. Therefore, one must use an appropriate generalization of subdifferentials and related concepts such as quasi conjugations. A variety of notions were introduced (e.g., Greenberg and Pierskalla [9, 10], Crouzeix [3, 5], Oettli [25, 26], Martinez-Legaz [21, 23], Singer [36–38], Passy and Prisman [27, 28], Penot and Volle [29, 30]). For a survey we refer to a paper of Penot and Volle [30].

In fact, there are several approaches to a generalization. A unified approach uses the observation that a quasi-convex function is the supremum of a family of quasi-affine functions which is the composition of linear functions and nondecreasing real extended-value single-variable functions. By using a subclass, K, of nondecreasing real extended-value single-variable functions one is able to develop a generalized subdifferential for quasi-convex functions (see, e.g., Penot and Volle [30]). Another interesting approach uses the observation that the level sets of a quasi-convex function on R^n are the intersections of halfspaces in R^n and each halfspace in R^n corresponds to a linear function on R^{n+1}. Thus, each quasi-convex function on R^n corresponds to a zero-degree homogeneous quasi-convex function on R^{n+1}. Therefore, using an extra parameter one also is able to define a generalized subdifferential for quasi-convex functions (Passy and Prisman [28]). Since these approaches are dealing with quite general quasi-convex functions, there exist unavoidable obstacles in several technical problems. The first obstacle is the contradiction between the simplicity and the generalized subdifferentiability. For instance, if the class K is covered by increasing functions, then a quasi-convex function is not K-subdifferentiable at the vectors

* Received by the editors November 18, 1992; accepted for publication (in revised form) April 12, 1994.

† Department of Industrial Engineering and Management, Tokyo Institute of Technology, Tokyo, Japan (phan@titisna.is.titech.ac.jp).

‡ Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan.

which are local minima but not global. If the class K covers the set of all increasing functions, then the set of K-subdifferentials is unbounded and contains zero in its boundary. This is technically difficult and seemingly inappropriate to obtain the variational inequalities. The second obstacle is that the generalized subdifferentials and the quasi conjugation involve an extra parameter. So, the dualization is not constructed in the usual dual space.

In order to avoid these obstacles one must restrict the class of quasi-convex functions. In this paper we investigate a subclass consisting of quasi-convex functions that attain the minimum value at the origin. Let us consider a quasi-convex program

$$\min\{f(x) : x \in D\}, \tag{1}$$

where f is a quasi-convex function, and D a closed convex set in R^n. If we have prior information, namely, a vector $z \in R^n$ such that

$$f(z) \leq \min\{f(x) : x \in D\},$$

then problem (1) is equivalent to

$$\min\{f^*(x) : x \in D^*\},$$

where $D^* = D - z$, and the function $f^*(x) = \max\{f(z), f(z+x)\}$ is quasi-convex and attains the minimum value at the origin. Our restriction to a quasi-convex function attaining the minimum value at the origin is thus practical if the prior information is available.

In [42] a generalized KKT condition is presented for a quasi-convex program. Nevertheless, this condition is sufficient but not necessary for the optimality. This is irregular but reasonable, since the property L=G does not hold for general quasi-convex programs. In this paper we show that for the functions satisfying the property L=G the generalized KKT condition is both necessary and sufficient. Therefore, the path-following methodology can, in principle, be used, although these functions do have nonextremal values at critical points according to usual differentials. Moreover, we prove that for the upper semicontinuous (briefly, USC) functions the property L=G is equivalent to the explicit quasi convexity, and we can check the validity of this property using a practical criterion (Lemma 1).

The paper is organized as follows. In §2 we present the so-called essential quasi convexity and its relationship with the explicit quasi convexity. In §3 we present a generalized KKT condition. In §4 we prove its equivalence to a generalized variational inequality. Using this variational inequality we show that the path-following methods can be applied to quasi-convex minimization problems. In §5 we provide a duality scheme using the minimax approach and establish a connection between the optimality condition and the existence of a saddle point. Finally we give concluding remarks in §6.

2. A generalized convexity. Let f be a real extended-value function. Define

$$\operatorname{dom}(f) = \left\{x : f(x) < \sup_{x \in R^n} f(x)\right\}.$$

This set is called the domain of f. The domain of a quasi-convex function is a convex set. In Mangasarian [20], a function f is said to be explicitly quasi-convex (on $\operatorname{dom}(f)$) if f is quasi-convex and for every $x \in \operatorname{dom}(f)$, $y \in \operatorname{dom}(f)$, $f(x) < f(y)$, $\lambda \in [0, 1)$

one has $f(x+\lambda(y-x)) < f(y)$. This notion of explicit quasi convexity (sometimes called strict quasi convexity, or functional quasi convexity) was used in a great deal of the literature (see, e.g., Hanson [11], Martos [24], Mangasarian [20]). It is well known that any local minimum on the domain of an explicitly quasi-convex function is global minimum (see, e.g., Mangasarian [20]).

Let f be a quasi-convex function. We consider the following properties of f:

(A1) any local minimizer of f on an arbitrary convex set $D \subseteq \mathrm{dom}(f)$ is a global minimizer of f on D;

(A2) any local minimizer of f on an arbitrary segment $[x,y] \subseteq \mathrm{dom}(f)$ is a global minimizer of f on $[x,y]$;

(A3) any local minimizer of f on $\mathrm{dom}(f)$ is a global minimizer.

LEMMA 1. *If f is* USC, *quasi-convex, and*

$$f(0) = \min\{f(x) : x \in R^n\},$$

then the property (A3) *is equivalent to one of the following properties:*

(A3′) *f is explicitly quasi-convex;*

(A3″) *for every $x \in \mathrm{dom}\ (f)$ such that $f(x) > f(0)$, one has*

$$f(x) > f(\lambda x) \quad \forall \lambda \in (0,1).$$

Proof. (A3′) $\Rightarrow$ (A3). See Mangasarian [20, p. 139].

(A3) $\Rightarrow$ (A3″). Let $x \in \mathrm{dom}(f)$ such that $f(x) > f(0)$. Since f satisfies (A3) and $f(x) > f(0)$, x is not a local minimizer; hence

$$x \in \mathrm{cl}\{y : f(y) < f(x)\}. \tag{2}$$

Since f is USC, the set $\{y : f(y) < f(x)\}$ is open. It follows from (2) that

$$[0,x) \subseteq \mathrm{int}(\mathrm{cl}(\{y : f(y) < f(x)\})) = \{y : f(y) < f(x)\}$$

(Rockafellar [33, p. 47]). Therefore, $f(x) > f(\lambda x)$ for any $\lambda \in [0,1)$.

(A3″) $\Rightarrow$ (A3′). Let x and y be vectors in $\mathrm{dom}(f)$ such that $f(x) < f(y)$. Suppose that there is $\lambda \in (0,1)$ such that $f((1-\lambda)x + \lambda y) = f(y)$. Then,

$$f((1-\theta)x + \theta y) = f(y) \quad \forall \theta \in [\lambda, 1].$$

Since $\{z : f(z) < f(y)\}$ is open, convex and the intersection of $\{z : f(z) < f(y)\}$ and the segment $[(1-\lambda)x + \lambda y, y]$ is empty, there is a linear function $\langle v, . \rangle$ strictly separating $\{z : f(z) < f(y)\}$ and $[(1-\lambda)x + \lambda y, y]$:

$$\langle v, z \rangle < 1 \quad \forall z : f(z) < f(y), \tag{3}$$

$$\langle v, (1-\theta)x + \theta y \rangle \geq 1 \quad \forall \theta \in [\lambda, 1]. \tag{4}$$

Since $f((1-\theta)x + \theta y) = f(y) > f(x) \geq f(0)$, it follows from (A3″) that

$$f(t((1-\theta)x + \theta y)) < f((1-\theta)x + \theta y) \quad \forall t \in (0,1)\ \forall \theta \in [\lambda, 1];$$

hence by (3)

$$\langle v, t((1-\theta)x + \theta y) \rangle < 1 \quad \forall t \in (0,1)\ \forall \theta \in [\lambda, 1].$$

This together with (4) implies

$$\langle v, (1-\theta)x + \theta y\rangle = 1 \quad \forall \theta \in [\lambda, 1].$$

Therefore, the hyperplane $\{z : \langle v, z\rangle = 1\}$ entirely contains the line $\{(1-t)x + ty : t \in R\}$. So, $\langle v, x\rangle = 1$. We arrive at a contradiction with (3) and $f(x) < f(y)$. □

Using Lemma 1 we obtain the following theorem.

THEOREM 1. *Let f be a quasi-convex function.*

(i) *f is explicitly quasi-convex* $\Leftrightarrow$ (A1) $\Leftrightarrow$ (A2) $\Rightarrow$ (A3);

(ii) *If in addition f is* USC *then one has*

f is explicitly quasi-convex $\Leftrightarrow$ (A1) $\Leftrightarrow$ (A2) $\Leftrightarrow$ (A3).

Proof. The proof of assertion (i) can be found in Mangasarian [20, p. 139]. In order to prove assertion (ii) we need to show that if f is USC then the property (A3) implies the explicit quasi convexity. Let f be a USC quasi-convex function satisfying (A3). Let $x, y \in \mathrm{dom}(f)$ such that $f(x) < f(y)$. Set

$$g(z) := \max\{f(x), f(x+z)\}.$$

The function g is USC, quasi-convex, and it satisfies (A3) and

$$g(0) = \min\{g(z) : z \in R^n\} = f(x) < f(y) = g(y-x).$$

By Lemma 1 this implies that

$$g(y-x) > g(\lambda(y-x)) \quad \forall \lambda \in (0,1).$$

Therefore,

$$\Rightarrow f(y) > f(x + \lambda(y-x)) \quad \forall \lambda \in (0,1).$$

So, f is explicitly quasi-convex. □

A quasi-convex function satisfying property (A3) will be called essentially quasi-convex. The essential quasi convexity is slightly weaker than the explicit quasi convexity, and in the class of USC functions they are equivalent.

In what follows we demonstrate that the class of essentially quasi-convex functions is invariant according to a quasi conjugation [41]. For a given real extended-value function f on R^n, a quasi conjugate of f, denoted by f^H, is a function on R^n defined as follows:

$$f^H(v) = \begin{cases} -\inf\{f(x) : \langle v, x\rangle \geq 1\} \text{ if } v \neq 0, \\ -\sup\{f(x) : x \in R^n\} \text{ if } v = 0. \end{cases}$$

The function $(f^H)^H$ is called the quasi biconjugate of f and is denoted by f^{HH}. Denote by Φ the class of functions f such that $f = f^{HH}$. Since f^{HH} is quasi-convex and [41]

$$f^{HH}(0) = \inf\{f^{HH}(x) : x \in R^n \setminus \{0\}\},$$

a function $f \in \Phi$ is quasi-convex and

$$f(0) = \inf\{f(x) : x \in R^n \setminus \{0\}\}.$$

THEOREM 2. *Assume $f \in \Phi$. Then, f is essentially quasi-convex if and only if f^H is essentially quasi-convex.*

Proof. Let $f \in \Phi$. Suppose that f is not essentially quasi-convex, i.e., there is a vector $x \in \text{dom}(f)$ such that $f(0) < f(x)$ and $x \notin \text{cl}\{y : f(y) < f(x)\}$. Since $\text{cl}\{y : f(y) < f(x)\}$ is closed convex and contains 0, there is a vector v such that

$$\langle v, x\rangle > 1, \tag{5}$$

$$\langle v, y\rangle < 1 \quad \forall y : f(y) < f(x). \tag{6}$$

This implies that

$$f^H(v) = -\inf\{f(y) : \langle v, y\rangle \geq 1\} = -f(x).$$

It follows from (5) and (6) that there is a neighborhood V of v such that $\langle u, x\rangle > 1$ for all $u \in V$. Let $u \in V$. One has

$$f^H(u) = -\inf\{f(y) : \langle u, y\rangle \geq 1\} \geq -f(x) = -f^H(v).$$

Therefore, v is a local minimizer of f^H. Since

$$f^H(0) = -\sup_{x \in R^n} f(x) < -f(x) = f^H(v) < -f(0) = -f^{HH}(0) = \sup_{v \in R^n} f^H(v),$$

v belongs to $\text{dom}(f^H)$ and it is not a global minimizer of f^H. Therefore, f^H is not essentially quasi-convex. Conversely if f^H is not essentially quasi-convex, then by the similar arguments as above, f^{HH} is not essentially quasi-convex. Since $f = f^{HH}$, this implies that f is not essentially quasi-convex. □

3. A generalized KKT condition. Let f be a real extended-value function on R^n. A vector v is called a quasi subdifferential of f at x if

$$\langle v, x\rangle = 1 \quad \text{and} \quad f(x) \leq f(y) \quad \forall y : \langle v, y\rangle \geq 1.$$

Denote by $\partial^H f(x)$ the set of quasi subdifferentials of f at x. If $\partial^H f(x) \neq \emptyset$ then f is quasi-subdifferentiable at x, and if $\partial^H f(x) \neq \emptyset$ for all $x \in X$ then f is quasi-subdifferentiable on X.

It is obvious that

$$\begin{aligned} & v \in \partial^H f(x) \\ \Leftrightarrow\ & \langle v, x\rangle = 1 \ \text{ and } \ f(x) = \inf\{f(y) : \langle v, y\rangle \geq 1\} \\ \Leftrightarrow\ & \langle v, x\rangle = 1 \ \text{ and } \ f^H(v) = -f(x). \end{aligned}$$

Furthermore, if $f \in \Phi$, then

$$\begin{aligned} & v \in \partial^H f(x) \\ \Leftrightarrow\ & \langle v, x\rangle = 1 \ \text{ and } \ f^H(v) = -f(x) \\ \Leftrightarrow\ & \langle v, x\rangle = 1 \ \text{ and } \ f^H(v) = -f^{HH}(x) \\ \Leftrightarrow\ & x \in \partial^H(f^H)(v). \end{aligned}$$

If f is a real value convex function, then for any x such that $f(x) > f(0)$ the set $\partial^H f(x)$ is nonempty, compact, convex, and [42]

$$\partial^H f(x) = \left\{ \frac{v}{\langle v, x\rangle} : v \in \partial f(x) \right\}. \tag{7}$$

Example 3.1. Let h_i, $i = 1,\dots,n$ be USC real extended-value single-variable increasing functions and f a function on R^n defined as follows:

$$f(x) = \max\{h_i(0), h_i(x_i) : i = 1,\dots,n\} \quad \forall x = (x_1,\dots,x_n) \in R^n.$$

Then, f is USC, essentially quasi-convex. For any x such that $\infty > f(x) > f(0)$ denote by $I(x)$ the set of indices i such that $f(x) = h_i(x_i)$. Then,

$$\partial^H f(x) = \left\{ \sum_{i\in I(x)} \frac{\theta_i}{x_i} e_i : \sum_{i\in I(x)} \theta_i = 1, \theta_i \geq 0,\ i \in I(x) \right\},$$

where e_i is the ith unit vector in R^n.

Denote by Ψ the set of quasi-convex functions satisfying

$$f(0) = \inf\{f(x) :\ x \in R^n \setminus \{0\}\}. \tag{8}$$

We consider now the following program:

$$\min\{f(x) : x \in D\}, \tag{9}$$

where $f \in \Psi$, and D is a closed convex set. Define

$$\ker(f) = \{x : f(x) = f(0)\}.$$

It follows from (8) that $\ker(f)$ is the set of global minimizers of f on R^n and it is a convex set. If $\ker(f) \cap D \neq \emptyset$ then

$$\min\{f(x) : x \in D\} = \min\{f(x) : x \in R^n\},$$

and every vector in the convex set $\ker f \cap D$ is a solution of the program (9). If $\operatorname{dom}(f) \cap D = \emptyset$, then f is constant on D, and every vector in D is a solution of the program (9). We are however interested in the following case:

$$\ker(f) \cap D = \emptyset, \quad \operatorname{dom}(f) \cap D \neq \emptyset. \tag{10}$$

If the program (9) is solvable, i.e., the function f has a minimum value on D, then the condition (10) is equivalent to

$$\min\{f(x) : x \in R^n\} < \min\{f(x) : x \in D\} < \sup\{f(x) : x \in R^n\}.$$

If f is USC then f is quasi-subdifferentiable on $R^n \setminus \ker(f)$, hence on D, and the set of quasi subdifferentials $\partial^H f(x)$ is convex, compact for every $x \in R^n \setminus \ker(f)$ [42]. Denote by $N(x, D)$ the normal cone of D at x:

$$N(x, D) = \{v \in R^n :\ \langle v, y - x\rangle \leq 0\ \forall y \in D\}.$$

THEOREM 3. *If $f \in \Psi$ is* USC, *essentially quasi-convex then a vector $x \in D$ is optimal to the program* (9) *if and only if*

$$0 \in \partial^H f(x) + N(x, D). \tag{11}$$

Proof. The condition (11) is sufficient for the optimality [42]. It remains to prove that it is a necessary condition. Suppose that $x \in D$ is an optimal solution. Then, $\{y : f(y) < f(x)\} \cap D = \emptyset$. So, there is a vector v such that

$$\langle v, y\rangle < 1 \quad \forall y : f(y) < f(x), \tag{12}$$

$$\langle v, y\rangle \geq 1 \quad \forall y \in D. \tag{13}$$

Since $x \in D$, it follows from (13) that $\langle v, x\rangle \geq 1$. Since f is essentially quasi-convex, x belongs to $\mathrm{cl}\{y : f(y) < f(x)\}$; hence it follows from (12) that $\langle v, x\rangle \leq 1$. Thus, $\langle v, x\rangle = 1$. This together with (12) implies

$$f(x) = \inf\{f(y) : \langle v, x\rangle \geq 1\}.$$

Therefore, $v \in \partial^H f(x)$. Moreover, $v \in -N(x, D)$ because of (13) and $\langle v, x\rangle = 1$. □

Let D be given by the following inequalities:

$$D = \{x : f_i(x) \leq 0, \quad i = 1, \ldots, m\},$$

where f_i, $i = 1, \ldots, m$ are real value convex functions on R^n. If the Slater's condition is satisfied

$$\exists x : \quad f_i(x) < 0 \quad \forall i = 1, \ldots, m,$$

then for every $x \in D$ there are nonnegative numbers $\lambda_1, \ldots, \lambda_m$ such that

$$N(x, D) = \mathrm{cone}\left(\sum_{i=1}^{m} \lambda_i \partial f_i(x)\right), \quad \lambda_i f_i(x) = 0, \quad i = 1, \ldots, m,$$

where $\mathrm{cone}(X)$ is the cone generated by X (Rockafellar [33]). From this together with Theorem 3 we can obtain a generalized KKT condition

$$0 \in \partial^H f(x) + \sum_{i=1}^{m} \lambda_i \partial f_i(x), \tag{14}$$

$$\lambda_i f_i(x) = 0, \quad i = 1, \ldots, m, \tag{15}$$

$$\lambda_i \geq 0, \quad f_i(x) \leq 0, \quad i = 1, \ldots, m. \tag{16}$$

Example 3.2. Let f be given as in Example 3.1 and f_i is differentiable for any $i = 1, \ldots, m$. Then, the condition (14)–(16) becomes the following system of equations and inequations:

$$0 = \sum_{i=1}^{n} \frac{\theta_i}{x_i} e_i + \sum_{i=1}^{m} \lambda_i \nabla f_i(x), \tag{17}$$

$$\lambda_i f_i(x) = 0, \quad f_i(x) \leq 0, \quad \lambda_i \geq 0, \quad i = 1, \ldots, m, \tag{18}$$

$$\sum_{i=1}^{n} \theta_i = 1, \quad \theta_i \geq 0, \quad \theta_i(f(x) - h_i(x_i)) = 0, \quad i = 1, \ldots, n, \tag{19}$$

where $\nabla f_i(x)$ denotes the gradient of f_i at x. Note that the system (17)–(19) is slightly different from the KKT condition even in a case where the functions h_i, $i = 1, \ldots, n$, are differentiable. In such a case, (17) is replaced by

$$0 = \sum_{i=1}^{n} \theta_i \nabla h_i(x_i) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x).$$

At a "stationary" point x such that $\nabla h_i(x_i) = 0$ $(i = 1, \ldots, n)$, the KKT condition is satisfied, but x may not be a minimizer, since h_i is nonconvex. Therefore, the generalized KKT condition (17)–(19) allows us to avoid the stationary points that are not minima.

Example 3.3. Suppose that function f_i is differentiable for any $i = 1, \ldots, m$, and function f is the quasi conjugate of a given function g: $f(x) = g^H(x)$. This situation may appear in quasi-convex dual programs which are discussed in §5. One has

$$\begin{aligned}\partial^H f(x) &= \{v : \langle v, x\rangle = 1,\ f^H(v) = -f(x)\} \\ &= \{v : \langle v, x\rangle = 1,\ g(v) = -f(x)\}.\end{aligned}$$

The condition (14)–(16) can be represented as follows:

$$0 = v + \sum_{i=1}^{m} \lambda_i \nabla f_i(x), \tag{20}$$

$$\lambda_i f_i(x) = 0,\ \lambda_i \geq 0,\ f_i(x) \leq 0, \quad i = 1, \ldots, m, \tag{21}$$

$$\langle v, x\rangle = 1,\ g(v) = -f(x). \tag{22}$$

Set $\lambda_i^+ = \max\{0, \lambda_i\}$, $\lambda_i^- = \max\{0, -\lambda_i\}$, $i = 1, \ldots, m$. Then, the system (14)–(16) can be represented as the following inclusion (Kojima [16]):

$$0 \in H(x, \lambda_1, \ldots, \lambda_m), \tag{23}$$

where H is a point-to-set mapping from R^{n+m} to R^{n+m} defined as follows:

$$H(x, \lambda_1, \ldots, \lambda_m) = \left\{ (v, \lambda_1^- - f_1(x), \ldots, \lambda_m^- - f_m(x)) : v \in \partial^H f(x) + \sum_{i=1}^{m} \lambda_i^+ \partial f_i(x) \right\}.$$

4. A generalized variational inequality and applications. As a consequence of Theorem 3, we can obtain a generalized variational inequality for the optimality in a quasi-convex minimization.

THEOREM 4. *If $f \in \Psi$ is* USC, *essentially quasi-convex, and it satisfies* (8), *then a vector $x \in D$ is optimal to program* (9) *if and only if x is a solution of the following generalized variational inequality:*

$$\min_{u \in \partial^H f(x)} \langle u, x - y\rangle \leq 0 \quad \forall y \in D. \tag{24}$$

Proof. One has

$$\begin{aligned}&x \text{ solves } (9) \\ \Leftrightarrow\ &0 \in \partial^H f(x) + N(x, D) \quad \text{(Theorem 3)} \\ \Leftrightarrow\ &\partial^H f(x) \cap -N(x, D) \neq \emptyset \\ \Leftrightarrow\ &\min_{u \in \partial^H f(x)} \sup_{y \in D} \langle u, x - y\rangle \leq 0 \quad \text{(since } \partial^H f(x) \text{ is compact)} \\ \Leftrightarrow\ &\sup_{y \in D} \min_{u \in \partial^H f(x)} \langle u, x - y\rangle \leq 0 \quad \text{(since } \partial^H f(x) \text{ is convex).}\end{aligned}$$

Therefore, x solves (9) if and only if x solves (24). □

If f is not essentially quasi-convex then the generalized variational inequality (24) is not necessary, and the set of solutions $x \in D$ satisfying (24) may be a subset of the set of optimal solutions of the program (9).

Remark. If $D = \{x : x \geq x_0\}$ where x_0 is a vector in R^n, and the quasi-subdifferential mapping $x \mapsto \partial^H f(x)$ is singleton on D, then it is easy to check that the variational inequality (24) becomes

$$\langle \partial^H f(x), x - x_0 \rangle = 0, \quad \partial^H f(x) \geq 0.$$

So, the set of optimal solutions is exactly the set of solutions of the following complementarity problem:

$$\langle \partial^H f(x), x - x_0 \rangle = 0, \quad \partial^H f(x) \geq 0, \quad x \geq x_0.$$

Note that the quasi-subdifferential mapping $x \mapsto \partial^H f(x)$ is not necessarily monotone in this complementarity problem:

Using the variational inequality (24) we are going to demonstrate that path-following methods could be used to find an optimal solution to the quasi-convex program (9) (e.g., Eaves [6, 7], Kojima [16, 17], Robinson [31, 32], Tuy [46], Yamamoto [48, 49]).

Let us consider a general variational inequality

$$\inf_{u \in g(x)} \langle u, x - y \rangle \leq 0 \quad \forall y \in D, \tag{25}$$

where D is a compact convex set, g is an arbitrary point-to-set mapping from $D \subseteq R^n$ to R^n. The general variational inequality (25) has a solution in D if g is a hemicontinuous (or closed) mapping and for any $x \in D$ the set $g(x)$ is convex and contained in a compact set K (Hartman and Stampacchia [12], Eaves [6], Karamardian [15]). Let $\pi(\cdot)$ be the projection mapping on D:

$$\pi(y) = \operatorname{argmin}\{\| x - y \| : x \in D\}.$$

The inequality (25) is then equivalent to the following inclusion:

$$u - \pi(u) \in g(\pi(u)),$$

or equivalently,

$$u \in E(u) := g(\pi(u)) + \pi(u). \tag{26}$$

Let B be a closed ball large enough so that $K + D \subseteq B$. Then, E is from B to B. By Kakutani's theorem it has a fixed point [14].

In order to use the fixed point theory to solve our inclusion

$$u \in \partial^H f(\pi(u)) + \pi(u), \tag{27}$$

we need to show that the quasi-subdifferential mapping $x \mapsto \partial^H f(x)$ is hemicontinuous and for any $x \in D$ the set of quasi subdifferentials $\partial^H f(x)$ is contained in a compact set K. Since $f^H(v) \leq -f(x)$ for all x such that $\langle v, x \rangle = 1$, one has

$$\begin{aligned} \partial^H f(x) &= \{v : \langle v, x \rangle = 1, f^H(v) = -f(x)\} \\ &= \{v : \langle v, x \rangle = 1\} \cap \{v : f^H(v) \leq -f(x)\}. \end{aligned}$$

Since f is USC, one has [43]

$$\{v : f^H(v) \leq -f(x)\} = \{y : f(y) < f(x)\}^0,$$

where A^0 denotes the polar of A. Therefore,

$$\partial^H f(x) \subseteq \{y : f(y) < f(x)\}^0.$$

It follows from $\inf\{f(x) : x \in D\} > f(0)$ that

$$\partial^H f(x) \subseteq K := \{y : f(y) < \alpha\}^0 \quad \forall x \in D,$$

where α is a number satisfying

$$\inf\{f(x) : x \in D\} > \alpha > f(0).$$

Since the level set $\{y : f(y) < \alpha\}$ is open and it contains 0, K is compact. In the following theorem we study the hemicontinuity of the quasi-subdifferential mapping $x \mapsto \partial^H f(x)$.

THEOREM 5. *Let $f \in \Psi$ be a* USC *function. If f is least semicontinuous* (LSC) *at $x \notin \ker(f)$ then $\partial^H f(\cdot)$ is hemicontinuous at x, i.e.,*

$$\{x_n\} \to x, \quad \{v_n\} \to v, \quad v_n \in \partial^H f(x_n) \Rightarrow v \in \partial^H f(x).$$

Proof. One has

$$v_n \in \partial^H f(x_n) \Leftrightarrow \langle v_n, x_n \rangle = 1, \quad f^H(v_n) \leq -f(x_n).$$

Since f is USC, f^H is LSC [43]. This together with the continuity of f at x implies

$$\langle v, x \rangle = 1, \ f(v) \leq \liminf_{n\to\infty} f^H(v_n) \leq \lim_{n\to\infty} -f(x_n) = -f(x).$$

So, $v \in \partial^H f(x)$. □

In the remainder of this section we discuss a homotopy of the inclusion (24). We demonstrate that the essential quasi convexity provides the nonsingularity of the homotopy inclusion (Theorem 6 below).

If the function f is convex and subdifferentiable on D, then instead of (27) we can use the usual subdifferential inclusion:

$$u - \pi(u) \in \partial f(\pi(u)).$$

Denote by $U(t)$, $t \in [0,1]$, the set of solutions of the linear homotopy inclusion

$$u - \pi(u) \in t\partial f(\pi(u)) + (1-t)(\pi(u) - x_0),$$

where $x_0 \in D$. Then, the path $\{\pi(U(t)), t \in [0,1]\}$ from x_0 to the set of optimal solutions is nonsingular, i.e., $\pi(U(t))$ is singleton for any $t \in [0,1)$, because $\pi(U(t))$ contains a unique minimizer of the strictly convex function $tf(x) + (1-t)\frac{\|x-x_0\|^2}{2}$ on D. Nevertheless, this type of linear homotopy may not provide a monotone path when f is essentially quasi-convex, since the function $tf(x) + (1-t)\frac{\|x-x_0\|^2}{2}$ may not be quasi-convex. In what follows we use another homotopy which is based on a retraction (Yamamoto [49]). For a family $\{D_t : t \in [0,1]\}$ of compact convex subsets of R^n, a continuous mapping $r_t : R^n \to R^n$ is a retraction onto D_t if $\{r_t(x) : x \in R^n\} \subseteq D_t$ and $r_t(x) = x$ for all $x \in D_t$. The projection mapping $\pi_t(\cdot)$ onto D_t is a retraction. We shall use this projection mapping and a family $\{D_t : t \in [0,1]\}$ which is an expansion from a singleton vector $x_0 \in \operatorname{int}(D)$ to D. Suppose

$$D = \{x : d(x) \leq 0\},$$

where $d(\cdot)$ is a real value convex function and the constraint qualification

$$\exists x: \quad d(x) < 0$$

is satisfied. Let x_0 be an arbitrary vector in $\text{dom}(f) \cap \text{int}(D)$. For every $t \in [0,1]$, define

$$D_t = \{x : (1-t) \parallel x - x_0 \parallel^2 + td(x) \leq 0\}.$$

It is obvious that

$$\begin{aligned} &D_0 = \{x_0\}, \\ &D_1 = D, \\ &D_0 \subseteq D_t \subseteq D_{t'} \subseteq D \quad \forall 0 \leq t \leq t' \leq 1. \end{aligned}$$

Since $(1-t) \parallel x - x_0 \parallel^2 + td(x)$ is strictly convex for any $t \in [0,1)$, the set D_t is strictly convex for any $t \in [0,1)$, i.e.,

$$x_1 \in D_t,\ x_2 \in D_t,\ x_1 \neq x_2,\ \lambda \in (0,1) \Rightarrow \lambda x_1 + (1-\lambda)x_2 \in \text{int}(D_t).$$

In the following theorem we prove the uniqueness of a minimizer of an essentially quasi-convex function on a strictly convex set.

THEOREM 6. *Let $f \in \Psi$ be a* USC *function. Then, f is essentially quasi-convex if and only if for every strictly convex set C such that*

$$C \cap \ker(f) = \emptyset, \quad C \cap \text{dom}(f) \neq \emptyset \tag{28}$$

f has at most a minimizer on C.

Proof. Suppose that there exist a strictly convex set C satisfying (28) and two minimizers x_1, x_2, $(x_1 \neq x_2)$ of f on C. Then, $(x_1 + x_2)/2$ is a minimizer of f on C as well. It follows from (28) that

$$\sup_{x \in R^n} f(x) > f((x_1 + x_2)/2) > f(0).$$

If f is essentially quasi-convex, then by Lemma 1 one has

$$f((x_1 + x_2)/2) > f(\lambda(x_1 + x_2)/2) \quad \forall \lambda \in [0,1).$$

We arrive at a contradiction with the fact that $(x_1+x_2)/2 \in \text{int}(C)$ and $(x_1+x_2)/2$ is a minimizer of f on C. Therefore, f is not essential quasi-convex. Suppose conversely that f is not essentially quasi-convex. Then, there is a local minimizer $x \in \text{dom}(f)$ that is not a global minimizer. So, there is a ball C centered at x such that $C \cap \{y : f(y) < f(x)\} = \emptyset$. This implies that x is a minimizer of f on the strictly convex set C. Since $f(\lambda x) \leq f(x)$ for any $\lambda \in [0,1]$, it follows that f has more than one minimizer on the strictly convex set C. □

Let π_t be the projection mapping on D_t and consider the following homotopy of the inclusion (27):

$$u \in E_t(u) := \partial^H f(\pi_t(u)) + \pi_t(u). \tag{29}$$

Since $x_0 \in \text{int}(D)$, the mapping $\pi_t(u)$ is hemicontinuous of (t,u) on $[0,1] \times R^n$. Therefore, if f is continuous on D then by Theorem 5, $\partial^H f(\cdot)$ is hemicontinuous of x on D and hence $E_t(u)$ is hemicontinuous of (t,u) on $[0,1] \times R^n$. Set

$$X(t) := \pi_t(U),$$

where U is the set of solutions of (29). If f is essentially quasi-convex, then by Theorem 4 $X(t)$ is the set of minimizers of f on D_t. Therefore, by Theorem 6, $X(t)$ is singleton for every $t \in [0,1)$ because

$$x_0 \in D_t \Rightarrow D_t \cap \mathrm{dom}(f) \neq \emptyset,$$
$$D_t \subseteq D \Rightarrow D_t \cap \ker(f) = \emptyset.$$

Thus, the path $\{X(t),\ t \in [0,1]\}$ from $X(0) = x_0$ to the set of optimal solutions of the program (9) is nonsingular.

5. A duality by minimax. In this section we present a connection between the generalized KKT condition and a quasi-convex duality by minimax. Let us consider a program

$$\min\{f(x) : x \in D\}, \tag{30}$$

where $f \in \Phi$ (i.e., $f^{HH} = f$), and D is a closed convex set in R^n satisfying

$$\lambda x \in D \quad \forall x \in D \quad \forall \lambda \geq 1. \tag{31}$$

Assumption (31) can be made without loss of generality. In fact, we can replace D by $D^* := \{\lambda x \in D : x \in D, \lambda \geq 1\}$, which satisfies this assumption. Since $f(\lambda x) \geq f(x)$ for any $\lambda \geq 1$, the minimum value of f on D is equal to the minimum value of f on D^*. Denote by D^H the reverse polar of D:

$$D^H = \{y :\ \langle y, x\rangle \geq 1 \quad \forall x \in D\}.$$

Since f has a minimum value at 0, if $0 \in D$ then the program (27) is trivial with an immediate optimal solution 0. Therefore, we can assume that $0 \notin D$. Since D satisfies (31), one has $D^{HH} = D$ (Tind [44]). We associate the program (30) with the following bifunction $F : R^n \times R^n \to R \cup \{\pm\infty\}$:

$$F(x,y) = f(x) + \delta(\langle x, y\rangle \mid \{t : t \geq 1\}),$$

where $\delta(\cdot \mid T)$ is the indicator function of T whose value is equal to 0 on T and to $+\infty$ off T. Let us consider the following minimax program:

$$\inf_{x \in R^n} \sup_{y \in D^H} F(x,y). \tag{32}$$

Since $D = (D^H)^H$, one has

$$\sup_{y \in D^H} F(x,y) = \begin{cases} f(x) & \text{if } x \in D, \\ +\infty & \text{if } x \notin D. \end{cases}$$

Therefore, the minimax program (32) is exactly the program (30). We call the following program

$$\sup_{y \in D^H} \inf_{x \in R^n} F(x,y), \tag{33}$$

the dual program of the program (30). For fixed y one has

$$\inf_{x \in R^n} F(x,y) = \inf_x \{f(x) : \langle y, x\rangle \geq 1\} = -f^H(y).$$

So, the dual (33) can be rewritten as follows:

$$\sup_{y \in D^H} -f^H(y), \tag{34}$$

or, equivalently,

$$\inf_{y \in D^H} f^H(y). \tag{35}$$

By a minimax principle we have the weak duality:

$$\inf_{x \in D} f(x) \geq \sup_{y \in D^H} -f^H(y). \tag{36}$$

The primal-dual pair (30), (35) is voluntary, since the dual of the dual (35) is exactly the primal:

$$\inf_{x \in D^{HH}} f^{HH}(x) \Leftrightarrow \inf_{x \in D} f(x).$$

If the bifunction $F(x, y)$ has a saddle point on $R^n \times D^H$, then we have the strong duality:

$$\inf_{x \in D} f(x) = \sup_{y \in D^H} -f^H(y). \tag{37}$$

THEOREM 7. *The following assertions are equivalent.*

(i) (x^*, y^*) *is a saddle point of* $F(\cdot, \cdot)$ *on* $R^n \times D^H$;

(ii) x^* *is a solution of the primal program* (30), y^* *is a solution of the dual program* (34), *and* $\inf(30) = \sup(34)$.

Proof. Suppose that (x^*, y^*) is a saddle point, i.e., $(x^*, y^*) \in R^n \times D^H$ and

$$\inf_{x \in R^n} F(x, y^*) = F(x^*, y^*) = \sup_{y \in D^H} F(x^*, y). \tag{38}$$

Since if $x^* \notin D$ then $\sup_{y \in D^H} F(x^*, y) = \infty$, it follows that $x^* \in D$. This together with (38) implies $F(x^*, y^*) = f(x^*)$ and $F(x^*, y^*) = -f^H(y^*)$. This together with the weak duality implies that x^* solves (30) and y^* solves (34). Conversely suppose that x^* solves (30), y^* solves (34) and $\inf(30) = \sup(34)$. Then,

$$\sup_{y \in D^H} F(x^*, y) = f(x^*) = -f^H(y^*) = \inf_{x \in R^n} F(x, y^*).$$

So, (x^*, y^*) is a saddle point. □

THEOREM 8. *If the generalized* KKT *condition* (11) *for the program* (30) *is satisfied at* $x^* \in D$, *then* (x^*, y^*) *is a saddle point for any* $y^* \in \partial^H f(x^*) \cap -N(x^*, D)$.

Proof. Suppose that the generalized KKT condition (11) is satisfied at $x^* \in D$ in the program (30). Then, x^* is optimal to the primal program (30). Let $y^* \in \partial^H f(x^*) \cap -N(x^*, D)$. It follows from $y^* \in \partial^H f(x^*)$ that $-f^H(y^*) = f(x^*)$ and $\langle y^*, x^* \rangle = 1$. Since $y^* \in -N(x^*, D)$, we have $\langle y^*, x - x^* \rangle \geq 0$ for all $x \in D$, or equivalently, $\langle y^*, x \rangle \geq 1$ for all $x \in D$; hence $y \in D^H$. By the weak duality (36) this implies that y^* solves the dual (34) and $\inf(30) = \sup(34)$. Thus, (x^*, y^*) is a saddle point. □

COROLLARY 1. *If* f *is* USC, *essentially quasi-convex and the program* (30) *is solvable, then there is a saddle point.*

Proof. This corollary is immediately obtained from Theorems 3 and 8. □

6. Concluding remarks. We presented a connection between a mathematical program and a variational inequality based on a generalized convexity: essential quasi convexity. The variational inequality was defined via the so-called quasi subdifferential. For the quasi subdifferentiability we need the upper semicontinuity. A weaker property such as the even quasi convexity would be enough for the quasi subdifferentiability (see, e.g., Passy and Prisman [28]). But the upper semicontinuity is essential for the compactness of the set of quasi subdifferentials, which enables us to obtain the variational inequality. In order to solve the variational inequality, we can use the usual homotopy approach in which the nonsingularity is guaranteed by the essential quasi convexity. We also presented a connection between the generalized KKT condition and a quasi-convex duality in which no extra parameter is involved.

REFERENCES

[1] M. Atteia and A. El Qortobi, *Quasiconvex duality*, Optimization and Optimal Control, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 30 (1981), pp. 3–8.

[2] F. H. Clarke, *Generalized gradient and applications*, Trans. Amer. Math. Soc., 205 (1975), pp. 247–262.

[3] J.-P. Crouzeix, *Conjugacy in quasi-convex analysis*, Convex Analysis and Its Applications, Lecture Notes in Economics and Math. Systems, Springer-Verlag, Berlin, 144 (1977), pp. 66–99.

[4] ———, *Continuity and differentiability properties of quasi-convex functions on* R^m, General Concavity in Optimization and Economics, S. Schaible, ed., Academic Press, New York, 1981, pp. 109–129.

[5] J.-P. Crouzeix and J. A. Ferland, *Algorithms for generalized fractional programming*, Math. Programming, 52 (1991), pp. 191–207.

[6] B. C. Eaves, *On the basic theorem of complementarity*, Math. Programming, 1 (1971), pp. 68–75.

[7] ———, *Computing Kakutani fixed points*, SIAM J. Appl. Math., 21 (1971), pp. 236–244.

[8] C. B. Garcia and W. I. Zangwill, *Pathways to Solution Fixed Points and Equilibria*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[9] H. P. Greenberg and W. P. Pierskalla, *Surrogate mathematical programming*, Oper. Res., 18 (1970), pp. 924–939.

[10] ———, *Quasiconjugate functions and surrogate duality*, Cahiers Centre Etudes Rech. Oper., 15 (1973), pp. 437–448.

[11] M. A. Hanson, *Bounds for functionally convex optimal control problems*, J. Math. Anal. Appl., 8 (1964), pp. 84–89.

[12] P. Hartman and G. Stampacchia, *On some nonlinear differential-functional equations*, Acta Mathematica, 115 (1966), pp. 271–310.

[13] J.-B. Hiriart-Urruty, *Generalized differentiability, duality and optimization for problems dealing with differences of convex functions*, Lecture Notes in Economic and Math. Systems, J. Ponstain, ed., Springer-Verlag, Berlin, 256 (1984), pp. 37–70.

[14] S. Kakutani, *A generalization of Brower's fixed point theorem*, Duke Math. J., 8 (1941), pp. 457–459.

[15] S. Karamardian, *The complementary problem*, Math. Programming, 2 (1971), pp. 107–129.

[16] M. Kojima, *A complementarity pivoting approach to parametric nonlinear programming*, Math. Oper. Res., 4 (1979), pp. 464–477.

[17] ———, *Strongly stable stationary solutions in nonlinear programs*, Analysis and computation of fixed points, S. M. Robinson, ed., Academic Press, New York, 1980, pp. 93–138.

[18] M. Kojima, N. Megiddo, T. Noma, and A. Yosishe, *A unified approach to interior point algorithms for linear complementarity problems*, Lecture Notes in Computer Science 538, Springer-Verlag, 1991.

[19] O. L. Mangasarian, *Pseudo-convex functions*, SIAM J. Control, 3 (1965), pp. 181–190.

[20] ———, *Nonlinear programming*, McGraw-Hill, New York, New York, 1969.

[21] J. E. Martinez-Legaz, *Un concepto generalizato de conjugation, applicacion a las funciones quasi-convexas*, Thesis, Barcelona, 1981.

[22] ———, *A generalized concept of conjugation*, Optimization Theory and Applications, Lecture Notes in Pure and Appl. Math., A. Auslender, J.-B. Hiriart-Urruty, and W. Oetlli, eds.,

Marcel Dekker, New York, 1983, pp. 45–89.
[23] J. E. MARTINEZ-LEGAZ, *Quasiconvex duality theory by generalized conjugation methods*, Optimization, 19 (1988), pp. 603–652.
[24] B. MARTOS, *The direct power of adjacent vertex programming methods*, Management Science, 12 (1965), pp. 241–252.
[25] W. OETTLI, *Optimality condition involving generalized convex mappings*, Generalized Concavity in Optimization and Economics, S. Schaible and W. T. Ziemba, eds., Academic Press, New York, 1981, pp. 227–238.
[26] ———, *Optimality conditions for programming problems involving multivalued mappings*, Applied Mathematics, B. Korte, ed., North-Holland Publishing Company, 1982, pp. 196–226.
[27] U. PASSY AND E. Z. PRISMAN, *Conjugacy in quasi-convex programming*, Math. Programming, 30 (1984), pp. 121–146.
[28] ———, *A convex-like duality scheme for quasi-convex programs*, Math. Programming, 32 (1985), pp. 278–300.
[29] J.-P. PENOT AND M. VOLLE, *Another duality scheme for quasi-convex problems*, Trends in Math. Optim., K. H. Hoffman, J.-B. Hiriart-Urruty, C. Lemarechal, and J. Zowe, eds., Birkhauser-Verlag, Basel, Boston, 1988, pp. 259–275.
[30] ———, *On quasi-convex duality*, Math. Oper. Res., 4 (1990), pp. 597–625.
[31] S. M. ROBINSON, *Generalized equations and their solutions, Part* I: *Basic theory*, Math. Prog. Studies, 10 (1979), pp. 128–141.
[32] ———, *Generalized equations*, Mathematical Programming, The State of the Art Bonn 1982, A. Bachem, M. Grotschel, and B. Korte, eds., 1983, pp. 346–367.
[33] R. T. ROCKAFELLAR, *Convex analysis*, Princeton Academic Press, Princeton, NJ, 1970.
[34] S. SCHAIBLE, *Fractional programming,* II: *On Dinkelbach's algorithm*, Management Science, 22 (1976), pp. 868–873.
[35] S. SCHAIBLE AND W. T. ZIEMBA, EDS., *General concavity in optimization and economics*, Academic Press, New York, 1981.
[36] I. SINGER, *Conjugate functional and level sets*, Nonlinear Anal. Theory, Methods Appl., 8 (1984), pp. 313–320.
[37] ———, *Generalized convexity, functional hulls and applications*, Selected Topics in Operations Research and Mathematical Economics, Proc. 8th Symp. Oper. Res., Karlsuhe, August 1983, Lecture Notes in Economics and Math. Systems, 226, Springer-Verlag, New York, pp. 49–79, 1984.
[38] ———, *A general theory of dual optimization problems*, J. Math. Anal. Appl., 116 (1986), pp. 77–130.
[39] J. STOER AND C. WITZGALL, *Convexity and optimization in finite dimensions* I, Springer-Verlag, Berlin, 1970.
[40] Y. TANAKA, M. FUKUSHIMA, AND T. IBARAKI, *On generalized pseudoconvex functions*, J. Math. Anal. Appl., 144 (1989), pp. 342–355.
[41] P. T. THACH, *Quasiconjugates of functions, duality relationship between quasiconvex minimization under a reverse convex constraint and quasiconvex maximization under a convex constraint, and applications*, J. Math. Anal. Appl., 159 (1991), pp. 299–322.
[42] ———, *A nonconvex duality with a zero gap and applications*, SIAM J. Optim., 4 (1994), pp. 44–64.
[43] ———, *Global optimality criteria and a duality with a zero gap in nonconvex optimization problems*, SIAM J. Math. Anal., 24 (1993), pp. 1537–1556.
[44] J. TIND, *Blocking and antiblocking sets*, Math. Programming, 6 (1971), pp. 157–166.
[45] H. TUY, *Sur les inegalites lineaires*, Colloquium Mathematicum, 13 (1964), pp. 107–123.
[46] ———, *Pivotal methods for computing equilibrium points: Unified approach and new restart algorithm*, Math. Programming, 16 (1979), pp. 210–227.
[47] Y. YAMAMOTO, *A new variable dimension algorithm for the fixed point problem*, Math. Programming, 25 (1983), pp. 329–342.
[48] ———, *A variable dimension fixed point algorithm and the orientation of simplices*, Math. Programming, 30 (1984), pp. 301–312.
[49] ———, *A unifying model based on retraction for fixed point algorithms*, Math. Programming, 28 (1984), pp. 192–197.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 227–249, February 1996
© 1996 Society for Industrial and Applied Mathematics
013

RESTRICTED STEP AND LEVENBERG–MARQUARDT TECHNIQUES IN PROXIMAL BUNDLE METHODS FOR NONCONVEX NONDIFFERENTIABLE OPTIMIZATION*

KRZYSZTOF C. KIWIEL†

Abstract. Two methods are given for minimizing locally Lipschitzian upper semidifferentiable functions. They employ extensions of restricted step (trust region) and Levenberg–Marquardt techniques that are widely used in other contexts. Extensions to linearly constrained optimization are discussed. Preliminary numerical experience is reported.

Key words. nondifferentiable (nonsmooth) optimization, mathematical programming, linearly constrained optimization, bundle methods, trust region methods, restricted step methods, locally Lipschitz functions, stationary points

AMS subject classifications. 65K05, 90C30

1. Introduction.

This paper considers the nondifferentiable optimization (NDO) problem of minimizing a (possibly nonconvex) locally Lipschitzian function $f : \mathbb{R}^N \to \mathbb{R}$. We assume that at each $x \in \mathbb{R}^N$ we can compute $f(x)$ and an arbitrary subgradient $g(x) \in \partial f(x)$. This problem abounds with applications and has been treated in many books and papers; see, e.g., [Cla83, Fle87, HaN88, Kiw85b, Lem89, LeM78, MäN92, Sch89, ScZ92, Sho79, StN88, Wol75] and the references therein.

The pioneering works of Lemarechal [Lem75] and Wolfe [Wol75] on NDO bundle methods for convex minimization started intensive research on extensions to the nonconvex case; see, e.g., [Bih84b, DiG80, Gol77, Kiw85b, Kiw85a, Lem80, LSB81, Mif77, Mif82, MGN87, PoM85, PMW83, Sch89, ScZ92, StN88] and the reviews in [HUL93, Kiw89c, Lem86, Lem89]. Yet significant computational advances [Kiw90, Sch89, ScZ92] are quite recent and are limited essentially to the convex case.

This paper attempts to extend these advances by presenting a restricted step (trust region) method and a Levenberg–Marquardt method. We establish a general framework for proving global convergence of such methods and their various modifications. We also try to offer a broader perspective by re-examining some ideas implicit in [Kiw85a, Kiw85b, Lem80, Mif77, PMW83, Sch89, Wol75] in order to identify techniques that ensure convergence and are likely to be efficient in practice. In particular, we feel that, since handling nonconvexity always requires some locality concepts in the x-space [DiG80, Gol77, Kiw85b, LSB81, Mif77, Mif82, PMW83], explicit consideration of trust region techniques should enhance practical efficiency of bundle methods. Our paper seems to be the first constructive attempt in this direction. Hence we consider mainly general concepts, leaving much room for (hopefully) efficient implementations.

We should add that Schramm and Zowe [Sch89, ScZ92] present another extension to the nonconvex case of a technique related to that of Kiwiel [Kiw90], but do not consider any trust region explicitly and give much weaker convergence results.

To save space, we shall describe extensions to linearly constrained NDO only, since for nonlinear constraints one can easily incorporate our techniques in the feasible point

* Received by the editors January 14, 1994; accepted for publication (in revised form) August 26, 1994. This research was supported by the Polish Academy of Sciences.
† Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland (kiwiel@ibspan.waw.pl).

methods of Kiwiel [Kiw85b], Mifflin [Mif77, Mif82], and Polak, Mayne, and Wardi [PMW83] and the exact penalty algorithms of Polak, Mayne, and Wardi [PMW83].

The paper is organized as follows. The restricted step algorithm is derived in §2 and analyzed in §3. Several modifications are discussed in §4. Section 5 describes subgradient aggregation (which is crucial for large-scale applications [Sch89, ScZ92]). Linearly constrained NDO is discussed in §6. The Levenberg–Marquardt method is introduced in §7. Our implementation of the methods is described in §8. Preliminary numerical experience with our algorithms is reported in §9. Section 10 concludes the paper.

We use the following notation. We denote by $\langle\cdot,\cdot\rangle$ and $|\cdot|$, respectively, the usual inner product and norm in $\mathbb{R}^N$. Both subscripts and superscripts are used to denote different vectors. Our general reference on NDO is found in [Cla83]. We say that $f:\mathbb{R}^N\to\mathbb{R}$ is locally Lipschitzian if for any bounded set $B\subset\mathbb{R}^N$ there exists a constant $L<\infty$ such that $|f(x)-f(y)|\le L|x-y|\ \forall x,y\in B$. The subdifferential (generalized gradient) of f at x is $\partial f(x)=\operatorname{co}\{\lim\nabla f(x^i): x^i\to x \text{ and } \nabla f(x^i) \text{ exists } \forall x^i\}$, where ∇f denotes the gradient of f and co denotes the convex hull. We say that $\bar{x}\in\mathbb{R}^N$ is stationary for f if $0\in\partial f(\bar{x})$, because $0\in\partial f(\hat{x})$ is a necessary condition for $\hat{x}$ to minimize f. For any $x\in\mathbb{R}^N$ and $\epsilon\ge 0$ the Goldstein ϵ-subdifferential [Gol77] of f at x is the set $\partial f(x;\epsilon)=\operatorname{co}\{\partial f(y):|y-x|\le\epsilon\}$. $\partial f(\cdot;\cdot)$ is locally bounded and upper semicontinuous (i.e., $x^i\to x$, $\epsilon^i\to\epsilon$, $p^i\in\partial f(x^i;\epsilon^i)$ and $p^i\to p$ imply $p\in\partial f(x;\epsilon)$; see, e.g., [Kiw85b, p. 159]). For convex f, $\partial f(x;\epsilon)$ is, in general, different from the usual ϵ-subdifferential in the sense of convex analysis, which is defined by $\partial_\epsilon f(x)=\{p\in\mathbb{R}^N: f(y)\ge f(x)+\langle p,y-x\rangle-\epsilon \quad \forall y\in\mathbb{R}^N\}$. We let $1{:}k$ denote $1,2,\dots,k$.

2. The restricted step method. We need some notation for introducing our restricted step method.

Let $\bar{f}(x;y)=f(y)+\langle g(y),x-y\rangle$ denote the value at x of the *linearization* of f computed at y, and $\alpha(x,y)=|f(x)-\bar{f}(x;y)|$ its (absolute) *error* at x. To deal with nondifferentiability of f, at iteration k we may use a piecewise linear (polyhedral) model of f of the form

$$\check{f}^k(x)=\max\{\,f(x^k)-\alpha(x^k,y^j)+\langle g(y^j),x-x^k\rangle : j\in J^k\,\},\tag{2.1}$$

where x^k is the current approximation to a minimizer of f, and the terms $f(x^k)-\alpha(x^k,y^j)+\langle g(y^j),x-x^k\rangle$ may be interpreted as (possibly shifted) linearizations of f accumulated at some *trial points* y^j of past iterations $j\in J^k\subset\{1{:}k\}$. Choosing a *weight* $u^k>0$ for the *search direction*

$$d^k=\arg\min\{\,\check{f}^k(x^k+d)+u^k|d|^2/2 : d\in\mathbb{R}^N\,\},\tag{2.2}$$

the method produces the next $x^{k+1}=x^k+t_L^k d^k$ and $y^{k+1}=x^k+t_R^k d^k$ by finding the left and right *stepsizes* $0\le t_L^k\le t_R^k>0$. The criterion for a *descent step* to x^{k+1}

$$f(x^{k+1})\le f(x^k)+m_L t_L^k v^k,\tag{2.3}$$

where $0<m_L<1$ and

$$v^k=\check{f}^k(x^k+d^k)-f(x^k)\tag{2.4}$$

is the (negative) *predicted descent* (for $t_L^k=1$), ensures that the actual descent is substantial if t_L^k is "sufficiently large." Otherwise, a *null step* with $t_L^k=0$ ($x^{k+1}=x^k$)

and y^{k+1} (i.e., t_R^k) chosen so that

$$\langle g(y^{k+1}), d^k \rangle - \alpha(x^{k+1}, y^{k+1}) \geq m_R v^k \tag{2.5}$$

with $0 < m_R < 1$ ensures that the next model $\check{f}^{k+1}$ with $k+1 \in J^{k+1}$ differs significantly from $\check{f}^k$, since $\check{f}^{k+1}(x^k+d^k) - f(x^k) \geq m_R v^k > v^k = \check{f}^k(x^k+d^k) - f(x^k)$.

The choice of J^k and u^k in (2.1)–(2.2) hinges on the following observation. Due to nonconvexity, $\check{f}^k$ is a useful local approximation to f only when (2.1) involves y^j close to x^k. Hence J^k may need to be reduced via *locality resets* that drop nonlocal past subgradients [Kiw85a, Kiw85b, Mif77, PMW83]. Furthermore, this observation suggests that $x^k + d^k$ should be restricted to some *trust region* $\{x : |x - x^k| \leq \rho^k\}$ with radius $\rho^k > 0$ (in which $\check{f}^k$ should be close enough to f for a descent step to occur). Consider, therefore, the restricted subproblem

$$\min\{\, \check{f}^k(x^k + d) : |d| \leq \rho^k \,\}. \tag{2.6}$$

Notice that varying $u^k = u$ in (2.2) yields a *Levenberg–Marquardt trajectory* [Fle87, p. 103], $\check{d}^k(u)$ say, which is continuous and piecewise linear in $1/u$, with $|\check{d}^k(u)|$ nonincreasing in u, $|\check{d}^k(u)| \downarrow 0$ as $u \uparrow \infty$, $\check{d}^k(u)$ minimizing $\check{f}^k$ for small u if $\operatorname{Arg\,min} \check{f}^k \neq \emptyset$, and $|\check{d}^k(u)| \uparrow \infty$ as $u \downarrow 0$ otherwise [Kiw95b, ScZ88]. Hence, as in [Kiw89b], an approximate solution d^k to (2.6) can easily be found by solving (2.2) with changed u^k until $|d^k| \approx \rho^k$, since $\check{d}^k(u)$ solves (2.6) with ρ^k replaced by $|\check{d}^k(u)|$. (In fact (2.6) could be solved *exactly* [Kiw95b], but this is not necessary.)

Our techniques for choosing ρ^k and u^k involve two components: a general theoretical framework for ensuring convergence, and specific (possibly heuristic) rules that are efficient in practice. The choice of u^k in (2.2) must balance the two conflicting needs: to improve $\check{f}^k$ by generating y^{k+1} near to x^k, and to step far away from x^k towards a minimizer of f when $\check{f}^k$ turns out to be close to f. As always, learning is costly but unavoidable, so a good method should converge quickly by learning just enough about f on the way. Thus our rules should, for instance, (as in the smooth case [DeS89]) ensure that ρ^k can decrease in a region where f is highly nonconvex and/or varies rapidly, and can increase in the region where f is better behaved; also ρ^k should decrease and then increase when the algorithm passes close to an almost stationary point of f. Our choices of ρ^k and u^k will be safeguarded to ensure that ρ^k can tend to zero and u^k to ∞ only if x^k approaches a stationary point of f. In turn, our heuristics will be similar to those used in other contexts [DeS83, DeS89, DNR87, Fle87, Mor83].

Our safeguards will require additional quantities. Let us rewrite (2.1) as

$$\check{f}^k(x) = f(x^k) + \max\{\, \langle g^j, x - x^k \rangle - \alpha_j^k : j \in J^k \,\}, \tag{2.7}$$

where $g^j = g(y^j)$ and $\alpha_j^k = \alpha(x^k, y^j)$. By (2.2) and (2.4), (d^k, v^k) solves the QP subproblem

$$\begin{array}{ll} \text{minimize} & v + u^k |d|^2/2 \quad \text{over all } (d, v) \in \mathbb{R}^{N+1} \\ \text{satisfying} & -\alpha_j^k + \langle g^j, d \rangle \leq v \quad \forall j \in J^k \end{array} \tag{2.8}$$

with Lagrange multipliers, say $\lambda_j^k \geq 0$, $j \in J^k$. As in [Kiw90], from the optimality condition $0 \in \partial \check{f}^k(x^k + d^k) + u^k d^k$ for (2.2) (or from the Karush–Kuhn–Tucker conditions for (2.8)) we deduce that

$$p^k + u^k d^k = 0, \tag{2.9}$$

$$-v^k = u^k|d^k|^2 + \alpha_p^k = |p^k|^2/u^k + \alpha_p^k, \tag{2.10}$$

with

$$(p^k, \alpha_p^k, 1) = \sum_{j \in J^k} \lambda_j^k (g^j, \alpha_j^k, 1), \tag{2.11}$$

$\alpha_p^k \geq 0$ and $v^k \leq 0$ because $\alpha_j^k \geq 0$. Moreover, if $J_s^k \subset J^k$ contains $\{j \in J^k : \lambda_j^k \neq 0\}$ then $\check{f}_s^k$, defined by (2.1) with J^k substituted by J_s^k, may replace $\check{f}^k$ in (2.2) and (2.4). This will justify dropping inactive constraints of (2.8), i.e., setting $J^{k+1} = J_s^k \cup \{k+1\}$ to save storage without impairing convergence.

By (2.11), $p^k \in \partial f(x^k; \hat{a}^k)$ with $\hat{a}^k = \max\{|y^j - x^k| : \lambda_j^k \neq 0, j \in J^k\}$. Without storing y^j, the method will compute an overestimate $a^k = \max\{s_j^k : \lambda_j^k \neq 0, j \in J^k\} \geq \hat{a}^k$ from the quantities $s_j^k = |y^j - x^j| + \sum_{i=j}^{k-1} |x^{i+1} - x^i| \geq |y^j - x^k|$, so that

$$p^k \in \partial f(x^k; a^k). \tag{2.12}$$

Moreover, if f is convex then $p^k \in \partial_{\alpha_p^k} f(x^k)$ [Kiw90], and hence

$$V^k \equiv |p^k| + \alpha_p^k \geq f(x^k) - \min\{f(x) : |x - x^k| \leq 1\}, \tag{2.13}$$

since $\min_{|x-x^k|\leq 1} f(x) \geq \min_{|x-x^k|\leq 1} f(x^k) + \langle p^k, x - x^k \rangle - \alpha_p^k = f(x^k) - |p^k| - \alpha_p^k$. Thus we may *measure the stationarity* of x^k by the quantity

$$\epsilon_p^k \equiv \max\{ m_p(|p^k| + \alpha_p^k), a^k \} = \max\{ m_p V^k, a^k \}, \tag{2.14}$$

where $m_p > 0$ is a fixed scaling coefficient (other forms of ϵ_p^k are given in §4).

We may now state our prototype restricted step method in general terms. The remarks following it and the convergence analysis of §3 will motivate its rules in more detail and will suggest several useful modifications in §4.

ALGORITHM 2.1.

Step 0 (Initialization). Select an initial point $x^1 \in \mathbb{R}^N$, a final optimality tolerance $\epsilon_{\text{opt}} \geq 0$, positive line search parameters m_L, m_R, m_v and $\bar{t} \leq 1$ satisfying $m_L + m_v < m_R < 1$, positive contraction parameters $\kappa_\delta < 1$ and $\kappa_\rho < 1$, a stationarity weight $m_p > 0$, a lower bound $u_{\min} > 0$, an initial weight $u^1 \geq u_{\min}$, an initial trust region radius $\rho^1 > 0$, an initial stationarity target $\delta^1 > 0$ and the maximum number of stored subgradients $M_g \geq N + 3$. Set $y^1 = x^1$, $J^1 = \{1\}$, $f_1^1 = f(y^1)$, $g^1 = g(y^1)$ and $s_1^1 = 0$. Set $\delta^0 = \delta^1$. Set the counters $k = 1$, $l = 0$ and $k(0) = 1$ ($k(l) - 1$ will denote the iteration number of the lth descent step).

Step 1 (Direction finding). Find the solution (d^k, v^k) of (2.8) and its multipliers λ_j^k such that the set $\hat{J}^k = \{j \in J^k : \lambda_j^k \neq 0\}$ satisfies $|\hat{J}^k| \leq N + 1$. Compute $|p^k|$ and α_p^k by (2.11). Set $a^k = \max\{s_j^k : \lambda_j^k \neq 0\}$ and ϵ_p^k by (2.14).

Step 2 (Stopping criterion). If $\epsilon_p^k \leq \epsilon_{\text{opt}}$, terminate; otherwise, continue.

Step 3 (Stationarity target updating). If $\epsilon_p^k \leq \delta^k$, replace δ^k by $\kappa_\delta \delta^k$.

Step 4 (Resetting test). If $m_p(|p^k| + \alpha_p^k) > a^k$, go to Step 6.

Step 5 (Locality reset). Delete from J^k one or more elements j with $\lambda_j^k > 0$ and largest $s_j^k > 0$ (or just $s_j^k > \delta^k$) and go to Step 1.

Step 6 (Trust region enforcement). If $|d^k| > \rho^k$, choose $u \in [1.1u^k, 10u^k]$, set $u^k = u$ and go to Step 1.

Step 7 (Unit stepsize testing). If $f(x^k + d^k) \le f(x^k) + m_L v^k$, set $x^{k+1} = y^{k+1} = x^k + d^k$, $t_L^k = t_R^k = 1$, $k(l+1) = k+1$, increase l by 1 and go to Step 9. Otherwise, either go to Step 8 or optionally, if

$$\max\{|f(x^k + d^k) - f(x^k)|, \alpha(x^k, x^k + d^k)\} > |p^k| + \alpha_p^k, \tag{2.15}$$

then choose $\rho \in [0.1\rho^k, 0.9\rho^k]$, $u \in [1.1u^k, 10u^k]$, set $\rho^k = \rho$, $u^k = u$ and go to Step 1.

Step 8 (Line search). By a line search procedure as discussed below, find two stepsizes $0 \le t_L^k \le t_R^k \le 1$ such that $x^{k+1} = x^k + t_L^k d^k$ and $y^{k+1} = x^k + t_R^k d^k$ satisfy (2.3) and *either a descent step is taken*: $t_L^k = t_R^k > 0$ and either $t^k \ge \bar{t}$ or $\alpha(x^k, x^k + t^k d^k) > m_v |v^k|$ for some $t^k \in [t_L^k, 10t_L^k]$, *or a null step occurs*: $t_L^k = 0$ and (2.5) holds. If $t_L^k > 0$, set $k(l+1) = k+1$ and increase l by 1.

Step 9 (Bundle updating). Select a set J_s^k such that $\hat{J}^k \subset J_s^k \subset J^k$, $|J_s^k| \le M_g - 1$ and $k(l) \in J^{k+1} = J_s^k \cup \{k+1\}$. Set $g^{k+1} = g(y^{k+1})$, $f_{k+1}^{k+1} = \bar{f}(x^{k+1}; y^{k+1})$, $f_j^{k+1} = f_j^k + \langle g^j, x^{k+1} - x^k \rangle$, $j \in J_s^k$, $s_{k+1}^{k+1} = |x^{k+1} - y^{k+1}|$, $s_j^{k+1} = s_j^k + |x^{k+1} - x^k|$, $j \in J_s^k$.

Step 10 (Weight and trust region updating). Do one of the following:

(i) If $\delta^k = \delta^{k-1}$ and $t_L^k = 0$, set $u^{k+1} = u^k$ and select $\rho^{k+1} \le \rho^k$ satisfying

$$\rho^{k+1} = \rho^k \quad \text{if} \quad \rho^k < \kappa_\rho \delta^k, \quad \rho^{k+1} \in [\kappa_\rho \delta^k, \max\{\delta^k, 0.9\rho^k\}] \quad \text{otherwise.} \tag{2.16a}$$

(ii) If $\delta^k = \delta^{k(l-1)}$ and $t_L^k > 0$, select $u^{k+1} \in [u_{\min}, u^k]$ and

$$\rho^{k+1} \ge \min\{\rho^k, \kappa_\rho \delta^k\}. \tag{2.16b}$$

(iii) If $\delta^k < \delta^{k-1}$ and $t_L^k = 0$, select $u^{k+1} \ge u^k$ and $\rho^{k+1} > 0$.

(iv) If $\delta^k < \delta^{k(l-1)}$ and $t_L^k > 0$, select $u^{k+1} \ge u_{\min}$ and $\rho^{k+1} > 0$.

Step 11. Set $\delta^{k+1} = \delta^k$, increase k by 1 and go to Step 1.

A few general comments on the method are in order.

Step 1 may use the dual QP methods of [Kiw89a, Kiw94], which can efficiently solve sequences of related subproblems (2.8) with varying u^k. Note that (2.8) involves $\alpha_j^k = |f(x^k) - f_j^k|$ with $f_j^k = \bar{f}(x^k; y^j)$ from Step 9. Step 2 is justified by the stationarity estimates (2.12)–(2.14). Steps 4 and 5 localize the past subgradient information that is used for search direction finding; hence the role of α in theory is secondary (e.g., $\alpha \equiv 0$ could be used as in [Mif77, PMW83]). If Step 6 sets $u = (1+\epsilon)u^k$ then any cycle between Steps 1 and 6 ends with $\rho^k/(1+\epsilon) \le |d^k| \le \rho^k$ (if Step 5 is not executed in the cycle) [Kiw89b].

Step 7 is intended to avoid expensive line searches. The test (2.15) of [Kiw90] (cf. (2.13)) ensures that u^k can become large only if ϵ_p^k is small, i.e., x^k is almost stationary (with $\epsilon_p^k = m_p V^k > a^k$ after Step 4). Due to Step 7, instead of testing just one u^k at iteration k, one may also search along the Levenberg–Marquardt trajectory to minimize $f(x^k + \check{d}^k(u))$ approximately, e.g., by imposing the additional requirement $\langle g(x^{k+1}), d^k \rangle \ge m_R v^k$ if $t_L^k = 1$ as in [Sch89]. Such a *zigzag search* was first suggested in [Lem80]. Its possible disadvantage is that it may not use each new subgradient to improve the model $\check{f}^k$; this places it between a line search strategy and that of [Kiw90].

Our line search rules combine those in [Kiw85a] and [Kiw85b, §7.8]. If f is upper semidifferentiable [Bih84b], the following procedure can be used (see [Kiw85b, p. 336]).

PROCEDURE 2.2 (line search).

(i) Set $t_L = 0$ and $t = t_U = 1$.

(ii) If $f(x^k + td^k) \le f(x^k) + m_L t v^k$ set $t_L = t$, otherwise $t_U = t$.

(iii) If $t_L > 0$ and either $t_U \ge \bar{t}$ or $\max\{\alpha(x^k, x^k + t_L d^k), \alpha(x^k, x^k + t_U d^k)\} > m_v|v^k|$, set $t_L^k = t_R^k = t_L$ and return.

(iv) If $t \le \bar{t}$ and $\langle g(x^k + td^k), d^k\rangle - \alpha(x^k, x^k + td^k) \ge m_R v^k$ set $t_R^k = t$, $t_L^k = 0$ and return.

(v) Choose $t \in [t_L + 0.1(t_U - t_L), t_U - 0.1(t_U - t_L)]$ and go to (ii).

Step (v) can choose t via quadratic interpolation [Kiw85b, p. 105]. Note that no line search is necessary if f is convex, since then $t_R^k = \bar{t} = 1$ may be used [Kiw90]. As in [Kiw85a], we do not allow "short descent steps" (cf. [Kiw85b, p. 102]), since they would complicate our analysis, and (2.3) with $t_L^k < 0.01$ (say) would yield insignificant descent anyway. The line search could also expand an initial $t_L = 1$ to find $t_L > 1$ [Kiw85b, p. 122], [Mif82], but we prefer to increase ρ^k and decrease u^k in this case. Moreover, one may enrich the bundle at Step 9 with some additional subgradients calculated by the line search [Mif77].

Our implementation of Step 9 sets $J^{k+1} = J^k \cup \{k+1\}$ and then, if necessary, drops from J^{k+1} an index $j \in J^k \setminus \hat{J}^k$ with the largest error α_j^k. The requirement $k(l) \in J^{k+1}$ (and hence $M_g \ge N + 3$) only ensures that $J^k \ne \emptyset$ at Step 5, for all k.

The rules of Step 10 are intended to leave room for decreasing the trust region after a null (or "weak" descent) step, and increasing it otherwise, depending on the agreement between f and $\check{f}^k$ at y^{k+1}. Some specific choices of ρ^{k+1} and u^{k+1} will be discussed in §8. Criteria (2.16) safeguard against jamming due to arbitrarily fast shrinkage of trust regions. (Conversely, the term $0.9\rho^k$ in (2.16a) allows slow shrinkage.) The *stationarity target* δ^k is decreased at Step 3 if it is met by the stationarity measure ϵ_p^k. Hence $\delta^k < \delta^{k-1}$ at Step 10 means the algorithm is making good progress. Thus the safeguards hinge on a simple idea: "dangerous" decreases of ρ^k and increases of u^k are allowed if progress occurs as monitored by δ^k (§3 shows that $\delta^k \downarrow 0$ if $\{x^k\}$ is bounded). As will be seen from our analysis, one could develop many variations of (2.16) without impairing convergence.

3. Convergence. In this section we establish global convergence of the method. We assume, of course, that the tolerance $\epsilon_{\text{opt}} = 0$. Then (2.12) and (2.14) imply that upon termination $0 \in \partial f(x^k)$. Hence we may suppose that the algorithm does not terminate.

Although Step 1 may be executed more than once at each iteration, this is not reflected in our notation, which should be clear from the context. In particular, any cycle between Steps 1 and 6 or 7 must be finite; see the remark after the proof of Lemma 3.6.

Our aim will be to verify the assumptions of the following consequence of (2.12) and the local boundedness and upper semicontinuity of $\partial f(\cdot;\cdot)$, which is proved in [Kiw85b, p. 159].

LEMMA 3.1. *If* $\liminf_{k\to\infty} \max\{|x^k - \bar{x}|, |p^k|, a^k\} = 0$ *for some* $\bar{x}$ *then we have* $0 \in \partial f(\bar{x})$. □

We start with three technical results. Their proofs follow easily from the analysis in [Kiw90], and hence are omitted (see also [Kiw95a]). We only note that Lemma 3.3 hinges on the property $u^k \ge u_{\min}$ for all k.

LEMMA 3.2. *If* $\{f(x^k)\}$ *is bounded from below (e.g.,* $\{x^k\}$ *has an accumulation point) then* $\sum_{k=1}^{\infty} t_L^k |v^k| < \infty$. □

LEMMA 3.3. *If* $B \subset \mathbb{R}^N$ *is bounded then there exists* $C < \infty$ *such that if* $x^k \in B$ *then*: (i) $|d^k| \le C/\sqrt{u^k}$; (ii) $|g^{k+1}| \le C$; (iii) $\max\{|f(x^k + td^k) - f(x^k)|, \alpha(x^k, x^k +$

$td^k)\} \le Ct|d^k| \ \forall t \in [0,10]$. □

LEMMA 3.4. (i) *Let* $w^k \equiv u^k|d^k|^2/2 + \alpha_p^k = |p^k|^2/2u^k + \alpha_p^k$. *Then* $v^k \le -w^k \le v^k/2$.

(ii) *If* $x^{k+1} = x^k$, $u^{k+1} \ge u^k$ *and* $J^{k+1} \supset J_s^k \cup \{k+1\}$ *at Step* 1 *then* $0 \le w^{k+1} \le w^k - u^k|d^{k+1} - d^k|^2/2$. □

The next two lemmas are crucial for analyzing null steps.

LEMMA 3.5. *If* $x^k = x^{k(l)} = \bar{x}$ *for some fixed* l *and all* $k \ge k(l)$, *and no resets occur for all large* k, *then* $w^k \downarrow 0$ *and* $v^k \to 0$.

Proof. By the algorithm's rules and Lemma 3.4(ii), $u^{k+1} \ge u^k$ and $w^{k+1} \le w^k$ for all large k, and $|d^{k+1} - d^k| \to 0$. Let $\bar{v} = \limsup_{k\to\infty} v^k$ and $K \subset \{1,2,\ldots\}$ satisfy $v^k \xrightarrow{K} \bar{v}$. Let $k \ge k(l)$ and $\epsilon^k = \langle g^{k+1}, d^k\rangle - \alpha_{k+1}^{k+1} - v^k$. Then, by (2.1) and (2.4) with $x^{k+1} = x^k$ and $k+1 \in J^{k+1}$,

$$\begin{aligned}\epsilon^k &= \langle g^{k+1}, d^{k+1}\rangle - \alpha_{k+1}^{k+1} - v^k - \langle g^{k+1}, d^{k+1} - d^k\rangle \\ &\le v^{k+1} - v^k + |g^{k+1}||d^{k+1} - d^k|,\end{aligned}$$

so $\limsup\{\epsilon^k : k \in K\} \le 0$ by Lemma 3.3(ii). But (2.5) holds for all large k, so $\epsilon^k \ge m_R v^k - v^k = (1 - m_R)|v^k|$ with $0 < m_R < 1$ imply $\bar{v} = 0$. Then $w^k \downarrow 0$ and $v^k \to 0$ by Lemma 3.4. □

LEMMA 3.6. *If* $x^k = x^{k(l)} = \bar{x}$ *for some fixed* l *and all* $k \ge k(l)$ *then* $\delta^k \downarrow 0$.

Proof. (i) If Step 3 decreases δ^k by $\kappa_\delta < 1$ infinitely many times then $\delta^k \downarrow 0$. Hence we may suppose, for contradiction, that $\delta^k = \bar{\delta} > 0$ for all $k \ge k(l)$. Then at Step 6 $V^k = |p^k| + \alpha_p^k > \bar{\epsilon} \equiv \bar{\delta}/m_p > 0$ for all $k \ge k(l)$ by (2.14) with $\epsilon_p^k > \delta^k$ and $m_p V^k > a^k$.

(ii) With constant $\delta^k = \bar{\delta} > 0$, the rules of the method ensure that $\rho^k \ge \bar{\rho} > 0$ and $u^k \le \bar{u} < \infty$ for some fixed $\bar{\rho}$, $\bar{u}$ and all large k. Indeed, by Lemma 3.3(iii), the left side of (2.15) is at most $C|d^k| \le C\rho^k$ after Step 6, so Step 7 cannot decrease ρ^k if $C\rho^k < \bar{\epsilon} \le |p^k| + \alpha_p^k$, whereas Step 10(i) can decrease ρ^k at most to $\kappa_\rho \bar{\delta}$. Moreover, by (i) and (iii) of Lemma 3.3, Steps 6 and 7 cannot increase u^k if $C/\sqrt{u^k} \le \bar{\rho}$ and $C^2/\sqrt{u^k} \le \bar{\epsilon} \le V^k$, respectively.

(iii) Suppose that only finitely many resets occur. By Lemma 3.5, $w^k = |p^k|^2/2u^k + \alpha_p^k \to 0$ and, since $u^k \le \bar{u}$ for all large k, we get $V^k \to 0$ at Step 6, a contradiction.

(iv) Suppose that infinitely many resets occur. With constant $\delta^k = \bar{\delta}$ and $t_L^k = 0$, no step can increase ρ^k and Step 10(i) ensures that $\rho^k \le \bar{\delta}$ for all large k. Therefore, $s_{k+1}^{k+1} \le \bar{\delta}$ for all large k by the rules of Steps 6–9 with $t_R^k \le 1$ and $|y^{k+1} - x^k| \le \rho^k$. Then finitely many resets drop all earlier j with $s_j^k > \bar{\delta}$ (if any), and $a^k \le \bar{\delta}$ afterwards. Thus eventually we have $m_p V^k > \bar{\delta} \ge a^k$ at Step 3, but Step 5 is entered with $m_p V^k \le a^k$, a contradiction. □

The above proof may be used to verify that any cycle between Steps 1 and 7 is finite. Otherwise we would have $u^k \uparrow \infty$, $d^k \to 0$, $\delta^k \downarrow 0$ and $p^k \to 0$ with $p^k \in \partial \check{f}^k(x^k + d^k)$, so that $0 \in \partial \check{f}^k(x^k)$ and $d^k = 0$ in the loop, contradicting (2.15).

We may now analyze the case of infinitely many descent steps.

LEMMA 3.7. *Suppose* $\bar{x}$ *is an accumulation point of* $\{x^k\}$ *but* $\{x^k\}$ *does not converge. Then* $\liminf_{k\to\infty} \max\{|x^k - \bar{x}|, |p^k| + \alpha_p^k\} = 0$ *at Step* 6.

Proof. (i) For contradiction, suppose the assertion is false. Then there exist $\bar{\epsilon} > 0$, $\bar{k}$ and an infinite set $K = \{k : k \ge \bar{k}, |x^k - \bar{x}| \le \bar{\epsilon}\}$ such that $|p^k| + \alpha_p^k > \bar{\epsilon}$ for all $k \in K$.

(ii) Suppose $K' = \{k \in K : \alpha_p^k \geq \bar{\epsilon}/2,\ t_L^k > 0\}$ is infinite. By (2.10), $v^k \leq \bar{v} \equiv -\bar{\epsilon}/2$ for all $k \in K'$. Hence, with $t_L^k v^k \to 0$ from Lemma 3.2, the rules of Steps 7 and 8 yield $t_L^k \xrightarrow{K'} 0$ and $\alpha(x^k, x^k + t^k d^k) > m_v|\bar{v}|$ for all large $k \in K'$. But with $\{d^k\}_{k\in K}$ bounded by Lemma 3.3(i) (with $u^k \geq u_{\min}$), we have $t^k|d^k| \leq 10 t_L^k |d^k| \xrightarrow{K'} 0$ and Lemma 3.3(iii) implies that $\alpha(x^k, x^k + t^k d^k) \xrightarrow{K'} 0$, a contradiction. Therefore K' is finite.

(iii) By (i) and (ii), we have $|p^k| > \bar{\epsilon}/2$ and, hence, by (2.9) and (2.10), $-t_L^k v^k \geq t_L^k |d^k||p^k| \geq |x^{k+1} - x^k|\bar{\epsilon}/2$ for all large $k \in K$ with $t_L^k > 0$.

(iv) Since $\{x^k\}$ does not converge to its cluster $\bar{x}$, there exists $\epsilon > 0$ such that for each $k \in K$ there exists $n > k$ satisfying $|x^n - x^k| > \epsilon$ and $|x^i - \bar{x}| \leq \bar{\epsilon}$ for all $k \leq i < n$. Therefore, by the triangle inequality and (iii), we have $\epsilon < |x^n - x^k| \leq \sum_{i=k}^{n-1} |x^{i+1} - x^i| \leq \sum_{i=k}^{n-1} t_L^i |v^i| 2/\bar{\epsilon}$ with the right side being less than ϵ for large $k \in K$ from Lemma 3.2, a contradiction. □

LEMMA 3.8. *If $l \to \infty$ and $x^k \to \bar{x}$ then $\delta^k \downarrow 0$.*

Proof. Arguing by contradiction, i.e., assuming that eventually $\delta^k = \bar{\delta} > 0$, and using parts (i) and (ii) of the proof of Lemma 3.6 with ρ^k and $1/u^k$ decreased by at most 10 at Steps 6 and 7, we obtain $u^k \leq \bar{u}$ for all large k. Then (2.10) and $|p^k| + \alpha_p^k > \bar{\epsilon} = \bar{\delta}/m_p$ imply that $v^k \leq \bar{v} < 0$ for some $\bar{v}$ and all large k, which leads to a contradiction as in part (ii) of the proof of Lemma 3.7. Hence $\delta^k \downarrow 0$. □

We may now prove our principal result.

THEOREM 3.9. *Every accumulation point of $\{x^k\}$ is stationary for f.*

Proof. Use the rule of Step 3, note that $m_p(|p^k| + \alpha_p^k) > a^k$ at Step 6 for Lemma 3.7, combine Lemmas 3.6–3.8 and apply Lemma 3.1. □

In the convex case, the above result may be strengthened as follows.

THEOREM 3.10. *If f is convex then either $x^k \to \bar{x} \in \operatorname{Arg\,min} f$ or $\operatorname{Arg\,min} f = \emptyset$ and $|x^k| \to \infty$. In both cases $f(x^k) \downarrow \inf f$.*

Proof. One can easily extend the analysis in [Kiw90] to Algorithm 2.1. □

We close this section with a result motivated by practice.

COROLLARY 3.11. *If the level set $\{x \in \mathbb{R}^N : f(x) \leq f(x^1)\}$ is bounded and $\epsilon_{\text{opt}} > 0$, then the algorithm terminates in finitely many iterations.*

Proof. Lemmas 3.6–3.8 show that $\delta^k \downarrow 0$ if $k \to \infty$ and $\{x^k\}$ is bounded. □

4. Modifications. It is worthwhile to ponder the role of the various rules in our convergence analysis. This will suggest several modifications. Not all of them seem equally useful in practice (see §9), but here we wish to explore theoretical possibilities. In particular, we hope that our discussion will help the interested reader in re-examining some ideas from the literature.

In Step 3, δ^k can be decreased "nonlinearly," provided that infinitely many decreases yield $\delta^k \downarrow 0$. For instance, as in [Mif77] we may decrease δ^k to $\phi(\delta^k)$, where $\phi(t) = \beta t^q$ with $q \geq 1$ and $0 < \beta < 1/(\delta^1)^{q-1}$, or $\phi(t) = t(1 - \beta t)$ with $0 < \beta < 1/\delta^1$. In fact one may choose any $\delta \in (0, \phi(\delta^k)]$ and set $\delta^k = \delta$, e.g., $\delta = \min\{\epsilon_p^k, \phi(\delta^k)\}$ for any such ϕ.

The stationarity target updating may be moved from Step 3 to Step 9 (so that it occurs at most once per iteration) and it may use the "record" value of ϵ_p^k. Thus suppose Step 0 sets $\hat{\epsilon}_p^k = \infty$, Step 1 sets $\hat{\epsilon}_p^k$ to $\min\{\hat{\epsilon}_p^k, \epsilon_p^k\}$, Step 3 is deleted, but if $\hat{\epsilon}_p^k \leq \delta^k$ at Step 9 then δ^k is replaced by $\kappa_\delta \delta^k$. This modification may provide more δ^k-decreases, and hence more freedom in selecting ρ^{k+1} and u^{k+1}.

Step 10(i) may employ the following weaker version of (2.16a)

$$\begin{aligned} \rho^{k+1} &= \rho^k \quad \text{if} \quad \rho^k < \kappa_\rho \min\{\delta^k, \hat{\epsilon}_p^k\}, \quad \text{otherwise,} \\ \rho^{k+1} &\in [\kappa_\rho \min\{\delta^k, \hat{\epsilon}_p^k\}, \max\{\delta^k, 0.9\rho^k\}], \end{aligned}$$

with Step 10(ii) selecting $\rho^{k+1} \geq \min\{\rho^k, \kappa_\rho \delta^k, \kappa_\rho \hat{\epsilon}_p^k\}$.

Step 10(iii) may choose $u^k \geq u_{\min}$ (like Step 10(iv)). Then Lemma 3.5 holds if δ^k becomes constant, as required in the proof of Lemma 3.6.

The following modified resetting strategy is in the spirit of [Mif77]. Let $0 < \beta_1 < \beta_2$ be fixed. Replace (2.16a) by $\rho^{k+1} \in [\beta_1 \delta^k, \beta_2 \delta^k]$, let Step 11 drop from J^{k+1} all j with $s_j^{k+1} > \beta_2 \delta^k$, and replace Steps 3 and 4 by the following.

Step 3' (Stationarity measure updating). If $m_p V^k > \delta^k$, go to Step 6. Otherwise replace δ^k by $\kappa_\delta \delta^k$.

Step 4' (Resetting test). If $a^k \leq \beta_2 \delta^k$, go to Step 3'.

Note that in this modification $a^k/\beta_2 \leq \delta^k < m_p V^k$ at Step 6. In part (iii) of the proof of Lemma 3.6 observe that eventually J^{k+1} is not reduced at Step 11, since $s_{k+1}^{k+1} \leq \beta_2 \bar{\delta}$ for all large k (cf. part(iv)).

The above modification is quite rigid in maintaining $a^k \leq \beta_2 \delta^k$ (with $\delta^{k+1} \leq \delta^k$) and may thus drop too many subgradients after each descent step. It seems more natural to reduce the bundle only if the current "search radius" $|d^k|$ is much shorter than the "locality radius" a^k [Kiw85b, Chap. 4], [Kiw85a]. Consider, therefore, choosing $m_a > 0$ at Step 0 and replacing Step 4 by Step 4".

Step 4" (Resetting tests). If $|d^k| \geq m_a a^k$, or $m_p V^k > a^k \leq \min\{1.1\rho^k, \delta^k\}$ and $\delta^k = \delta^{k-1}$, go to Step 6; otherwise, continue.

These conditions (if violated) enable additional resets when $|d^k| < m_a a^k$. The proof of Lemma 3.6 is modified as follows. In part (i), $\max\{m_p V^k, a^k\} > \bar{\delta}$ at Step 3, and at Step 6 either $m_p V^k > a^k$ or $m_p V^k \leq a^k > \bar{\delta}$ and $|d^k| \geq m_a a^k$ implies $|p^k| \geq u^k m_a a^k \geq u_{\min} m_a \bar{\delta}$ by (2.9), so that $V^k \geq \bar{\delta} \min\{1/m_p, u_{\min} m_a\}$ in both cases. In part (iv), note that $\rho^k \downarrow \bar{\rho} > 0$, $s_{k+1}^{k+1} \leq \min\{1.1\bar{\rho}, \bar{\delta}\}$ and $a^k \leq \min\{1.1\rho^k, \bar{\delta}\}$ for all large k. Next, to use Lemma 3.7 in the proof of Theorem 3.9, recall that $|d^k| \geq m_a a^k$ implies $V^k \geq u_{\min} m_a a^k$.

It may seem difficult to choose a suitable value of m_p in (2.14) and the resetting test of Step 4 for comparing the "horizontal" variable a^k with the "vertical" quantity V^k of (2.13), which depends on the scaling of f. In general, we would like the method to decrease both V^k and a^k. In this multiobjective context we need some scalarizing function of V^k and a^k. Of course, (2.14) gives a simple example, but one may also consider $\epsilon_p^k = \max\{\phi_1(V^k), \phi_2(a^k)\}$, where each ϕ_i is a strict forcing function in the sense that it maps R_+ into itself, is strictly increasing, continuous at zero and $\phi_i(0) = 0$. Then Step 4 should use the resetting test $\phi_1(V^k) > \phi_2(a^k)$, and, if $\kappa_\rho \delta^k$ and δ^k in (2.16) are replaced by $\phi_2^{-1}(\kappa_\rho \delta^k)$ and $\phi_2^{-1}(\delta^k)$ respectively, one may easily verify the preceding results. Moreover, for the alternative definition of $\epsilon_p^k = \phi_1(V^k) + \phi_2(a^k)$, (2.16) should involve $\phi_2^{-1}(\kappa_\rho \delta^k/2)$ and $\phi_2^{-1}(\delta^k/2)$ instead of $\kappa_\delta \delta^k$ and δ^k if we wish to use the following version of Step 4".

Step 4"' (Resetting tests). If $|d^k| \geq m_a a^k$, or if $\phi_1(V^k) > \phi_2(a^k)$ and $a^k \leq \min\{1.1\rho^k, \phi_2^{-1}(\delta^k/2)\}$ and $\delta^k = \delta^{k-1}$, go to Step 6; otherwise, continue.

Again, δ^k replaces $\delta^k/2$ above if we use $\epsilon_p^k = \max\{\phi_1(V^k), \phi_2(a^k)\}$.

If the test $|d^k| > \rho^k$ of Step 6 is replaced by $|d^k| > (1+\beta)\rho^k$ with some fixed $\beta > 0$, then only $\delta^k/(1+\beta)$ should substitute δ^k in (2.16a) (to get $a^k \leq \bar{\delta}$ in (iv) of

the proof of Lemma 3.6). With $\beta \in [0.5, 1]$, d^k would correspond to usual inexact solutions of trust region subproblems [DeS83, Mor83] that save some QP work.

To enable additional ρ^k-decreases and u^k-increases without impairing convergence, one may use weaker forms of condition (2.15), e.g.,

$$\max\{|f(x^k+d^k)-f(x^k)|, \alpha(x^k, x^k+d^k), |g(x^k+d^k)||d^k|\} > \min\{\hat{\epsilon}_p^k, \delta^k\}/m_p \tag{4.1}$$

with $\hat{\epsilon}_p^k = \min_{j=1:k} \epsilon_p^k$; clearly, $\delta^k \geq \min\{\kappa_\delta \hat{\epsilon}_p^k, \delta^1\}$ but $\hat{\epsilon}_p^k \ll \delta^k$ is possible. (In particular, $|p^{k-1}| \max\{1, |x^k - x^{k-1}|\} + \alpha_p^{k-1}$ may be used as the right side of (4.1) as in [Sch89].) As in [Kiw90], if (4.1) holds at Step 7 then at Step 10(i) (with the same k) one may choose any $u^{k+1} \in [u^k, 10u^k]$ and $\rho^{k+1} \in [0.1\rho^k, \rho^k]$. Also the line search may be *interrupted* if (4.1) holds with d^k replaced by td^k ($t \leq 1$ being the trial stepsize), i.e., the algorithm may return to Step 1 with ρ^k and u^k changed as in Step 7; cf. Lemma 3.3(iii) and (ii) of the proof of Lemma 3.6. On the other hand, a descent step may occur if

$$\max\{|f(x^k+t^kd^k)-f(x^k)|, \alpha(x^k, x^k+t^kd^k)\} > \min\{V^k, m_v|v^k|\},$$

cf. the proof of Lemma 3.7.

Thus, as long as some form of (4.1) holds, one may search along the Levenberg–Marquardt trajectory by increasing u^k in (2.8). It can, however, be more efficient to modify this trajectory when new subgradient information becomes available. To this end, if (4.1) holds then from Step 7, instead of returning to Step 1, we may go to Step 9 with $t_L^k = 0$, $t_R^k = 1$, $x^{k+1} = x^k$ and $y^{k+1} = x^k + t_R^k d^k$ ($t_R^k = t$ if (4.1) holds with d^k replaced by td^k at Step 8). Such null steps might not ensure (2.5) (but they are ruled out for large k in (ii) of the proof of Lemma 3.6 due to u^k-increases, so the proof of Lemma 3.5 goes through).

Of course, our convergence analysis does not enforce a zigzag search if (4.1) occurs, and additional criteria may be employed. For example, a null step may be accepted if (2.5) holds, possibly with $\alpha(x^k, y^{k+1}) \leq -m_\alpha v^k$ or $\alpha(x^k, y^{k+1}) \leq m_\alpha \alpha_p^k$, where $m_\alpha > 0$.

The zigzag search may also expand $|d^k|$ when further descent is likely, e.g., $f(x^k + d^k) \leq f(x^k) + m_c v^k$ with $m_c \in (1/2, 1)$, or (2.3) holds but $\langle g(x^k+d^k), d^k\rangle < m_R v^k$. Then, for $\rho' = \rho^k$ and $u' = u^k$, Step 7 may choose $\rho'' \in (\rho', 10\rho']$, $\max\{0.1u', u_{\min}\} \leq u'' < u'$ and return to Step 1 with $\rho^k = \rho''$, $u^k = u''$ (unless $u^k = u_{\min}$). If the new $f(x^k+d^k) > f(x^k)+m_L v^k$, one may reset $\rho^k = \rho'$, $u^k = u'$ and accept the original step. Alternatively, one may interpolate in the bracket $[u'', u']$ to choose $u''' \in [u''+0.1(u'-u''), u'-0.1(u'-u'')]$, $\rho''' = \rho''$ (or $\rho' < \rho''' < \rho''$ if Step 6 is temporarily ignored), with u''' replacing u' or u'' in the next bracket. Simple safeguards (like $u'' \leq 0.9u'$) will prevent infinite looping, but it is worthwhile to observe that in the convex case such interpolations will yield (2.3) and $\langle g(x^{k+1}), d^k\rangle \geq m_R v^k$ in finitely many steps (barring roundoff errors, of course). Indeed, with $\check{d}(u)$, $\check{v}(u)$ corresponding to $u^k = u$ in (2.2), if $\langle g(x+d'), d'\rangle < m_R v'$ and $f(x+d'') > f(x) + m_L v''$, where $x = x^k$, $d' = \check{d}(u')$ etc., then in fact $f(x+d') < f(x) + m_R v'$ by the subgradient inequality, so that if $u' \downarrow \bar{u}$ and $u'' \uparrow \bar{u}$ then $f(x)+m_L\check{v}(\bar{u}) \leq f(x+\check{d}(\bar{u})) \leq f(x)+m_R\check{v}(\bar{u})$ by continuity, and this would contradict $0 < m_L < m_R$ and $\check{v}(\bar{u}) < 0$ (since either $\check{v}(u) < 0\ \forall u > 0$ or $\check{v} \equiv 0$ and the algorithm would terminate at Step 3). Possible resets do not invalidate this argument (which, incidentally, is simpler than that in [Sch89, ScZ88, ScZ92]). It also shows that interpolation may be used to satisfy *Goldstein's criteria*, i.e., (2.3)

and $f(x^{k+1}) \geq f(x^k) + m_R v^k$. Of course, it is restricted to the convex case, and in practice safeguards should be included anyway.

Note that by Lemma 3.4(i), the predicted decrease $-w^k$ based on the quadratic model $\tilde{f}^k + u^k|\cdot - x^k|^2/2$ is related to v^k (based on $\tilde{f}^k$). In fact, $-w^k$ may replace v^k at Steps 7 and 8. Then $\sum_k t_L^k w^k < \infty$ in Lemma 3.2, $\langle g^{k+1}, d^k\rangle - \alpha_{k+1}^{k+1} \geq -m_R w^k \geq m_R v^k$ in the proof of Lemma 3.5, $w^k \geq \bar{w} \equiv \bar{\epsilon}/2$ and $\alpha(x^k, x^k + t^k d^k) > m_v \bar{w}$ in (ii) and $t_L^k w^k \geq t_L^k |d^k||p^k|/2 \geq |x^{k+1} - x^k|\bar{\epsilon}/4$ in (iii) of the proof of Lemma 3.7. Moreover, combinations of v^k, $-w^k$ and $-u^k|d^k|^2/2$ can be used as in [Kiw90]. For example, (2.5) may be replaced by $\langle g^{k+1}, d^k\rangle - \alpha_{k+1}^{k+1} \geq -m_R w^k - u^k|d^k|^2/2$ that yields (2.5) with m_R replaced by $m_R' = (m_R+1)/2$ and $m_R w^k + u^k|d^k|^2/2 = m_R' u^k|d^k|^2 + m_R \alpha_p^k \leq -m_R' v^k$.

Finally, to enable more frequent acceptance of unit stepsizes, Steps 6 and 7 may be switched. Specifically, if $f(x^k + d^k) \leq f(x^k) + m_L v^k$ at Step 6 then one may proceed to Step 9 as in Step 7. This may, however, involve extra f-evaluations when unit stepsizes are rejected.

5. Subgradient aggregation. To trade off storage and work per iteration for speed of convergence, one may replace subgradient selection with aggregation as in [Kiw85b]. Then only one aggregate subgradient and $M_g \geq 2$ "ordinary" ones need be stored. To this end, suppose at iteration $k \geq 2$ we have an aggregate subgradient $(p^{k-1}, \tilde{f}_p^k)$ satisfying

$$(p^{k-1}, \tilde{f}_p^k, 1) = \sum_{j=1}^{k-1} \tilde{\lambda}_j^{k-1} (g^j, f_j^k, 1), \tag{5.1}$$

for some multipliers $\tilde{\lambda}_j^{k-1} \geq 0$, and we know some

$$\tilde{a}_p^k \geq \max\{\, s_j^k : \tilde{\lambda}_j^{k-1} \neq 0, \quad j = 1\!:\!k-1 \,\}, \tag{5.2}$$

so that $p^{k-1} \in \partial f(x^k; \tilde{a}_p^k)$ as in (2.12), and in the convex case $p^{k-1} \in \partial_\epsilon f(x^k)$ for $\epsilon = \tilde{\alpha}_p^k \equiv |f(x^k) - \tilde{f}_p^k|$ (analogous to $\alpha_j^k = |f(x^k) - f_j^k|$). For $k = 1$, let $(p^0, \tilde{f}_p^1, \tilde{a}_p^1) = (g^1, f_1^1, 0)$. At Step 1 append to (2.8) the aggregate constraint $-\tilde{\alpha}_p^k + \langle p^{k-1}, d\rangle \leq v$ with Lagrange multiplier $\lambda_p^k \geq 0$. Considering

$$(p^k, \alpha_p^k, f_p^k, 1) = \sum_{j \in J^k} \lambda_j^k (g^j, \alpha_j^k, f_j^k, 1) + \lambda_p^k (p^{k-1}, \tilde{\alpha}_p^k, \tilde{f}_p^k, 1) \tag{5.3}$$

as an extension of (2.11), we get (2.12) for $a^k = a_p^k$ with

$$a_p^k = \max\{\, s_j^k : \lambda_j^k \neq 0, j \in J^k; \quad \tilde{a}_p^k : \text{ if } \lambda_p^k \neq 0 \,\}. \tag{5.4}$$

Next, the aggregate subgradient is dropped at Step 5 if $a^k = \tilde{a}_p^k$, and Step 9 sets $\tilde{f}_p^{k+1} = f_p^k + \langle p^k, x^{k+1} - x^k\rangle$ and $\tilde{a}_p^{k+1} = a_p^k + |x^{k+1} - x^k|$. Letting $\tilde{\lambda}_k^k = \lambda_k^k$, $\tilde{\lambda}_j^k = \lambda_j^k + \lambda_p^k \tilde{\lambda}_j^{k-1}$, $j \in J^k \setminus \{k\}$, $\tilde{\lambda}_j^k = \lambda_p^k \tilde{\lambda}_j^{k-1}$, $j \notin J^k$, one gets (5.1) and (5.2) for k increased by 1. Also $\alpha_p^k \geq |f(x^k) - f_p^k|$ by (5.3) and the convexity of $|\cdot|$, and this suffices for proving Lemma 3.4(ii) as in [Kiw90]. (It also shows that resetting v^k to $-u^k|d^k|^2 - |f(x^k) - f_p^k|$ as in [Kiw85a, Kiw85b] is unnecessary.) Thus one may verify all the preceding convergence results.

6. Linearly constrained minimization. Following [Kiw90], one can easily extend our method to the problem of minimizing f on a nonempty closed convex subset S of $\mathbb{R}^N$. To this end, choose $x^1 \in S$ and add to $\check{f}^k$ in (2.2) the indicator δ_S of S (null on S, ∞ outside). Note that (2.8) is augmented with the constraint $x^k + d \in S$, and can be solved by the QP method of [Kiw89a] if S is described by finitely many linear inequalities or equalities (without any constraint qualification). Then $x^k + d^k \in S$ and, by convexity and by induction with $0 \le t_L^k \le t_R^k \le 1$, $x^k \in S$ and $y^k \in S$ for all k.

To establish convergence, we observe as in [Kiw90] that (2.9) and (2.10) hold with $p^k = p_f^k + p_S^k$ and $\alpha_p^k = \alpha_{fp}^k + \alpha_S^k \ge 0$, where $(p_f^k, \alpha_{fp}^k) = \sum_j \lambda_j^k (g^j, \alpha_j^k)$ as in (2.11), $\alpha_{fp}^k \ge 0$, $p_f^k \in \partial f(x^k; a^k)$ as in (2.12) and $p_S^k \in \partial_\epsilon \delta_S(x^k)$ with $\epsilon = \alpha_S^k \equiv \langle p_S^k, d^k \rangle \ge 0$. Hence $\epsilon_p^k = 0$ implies that $\bar{x} = x^k$ is *stationary for f on S*, i.e., $\bar{x} \in S$ and there exist $\bar{p}_f \in \partial f(\bar{x})$ and $\bar{p}_S \in \partial \delta_S(\bar{x})$ such that $\bar{p}_f + \bar{p}_S = 0$. Next, Lemma 3.1 may be replaced by the following extension of [Kiw86b, Lemma 3.2].

LEMMA 6.1. *If $x^k \xrightarrow{K} \bar{x}$, $p^k \xrightarrow{K} 0$, $a^k \xrightarrow{K} 0$ and $\alpha_S^k \xrightarrow{K} 0$ for some $\bar{x} \in S$ and $K \subset \{1, 2, \ldots\}$, then $0 \in \partial f(\bar{x}) + \partial \delta_S(\bar{x})$.*

Proof. Since $p_f^k \in \partial f(x^k; a^k)$ and $\partial f(\cdot;\cdot)$ is locally bounded and upper semicontinuous, there exist $\bar{p}_f \in \partial f(\bar{x})$ and an infinite $K' \subset K$ such that $p_f^k \xrightarrow{K'} \bar{p}_f$ [Kiw90, p. 159]. Hence $p_S^k = p^k - p_f^k \xrightarrow{K'} -\bar{p}_f \in \partial \delta_S(\bar{x})$ because $\delta_S(x) \ge \langle p_S^k, x - x^k \rangle - \alpha_S^k \ \forall x$ with $\alpha_S^k \xrightarrow{K'} 0$ and $x^k \xrightarrow{K} \bar{x}$. □

Then, since Lemmas 3.1–3.4 remain valid (see [Kiw90]), one may combine Lemmas 3.5–3.8 to show that every accumulation point of $\{x^k\}$ is stationary for f on S; moreover, Theorem 3.10 holds with f replaced by $f + \delta_S$.

7. The methods with subgradient locality measures. Instead of dropping nonlocal past subgradients, one may decrease their activity at search direction finding by using subgradient locality measures [Kiw85b, LSB81, Mif82]. Let us, therefore, redefine

$$\alpha(x, y) = \max\{\, |f(x) - \bar{f}(x; y)|, \gamma_1 |x - y|^2, \gamma_2 |g(y)||x - y| \,\}, \tag{7.1}$$

where γ_1 and γ_2 are positive parameters, and let (2.7) involve

$$\alpha_j^k = \max\{\, |f(x^k) - f_j^k|, \gamma_1 (s_j^k)^2, \gamma_2 |g^j| s_j^k \,\}, \tag{7.2}$$

so that α_j^k overestimates $\alpha(x^k, y^j)$ without storing y^j. A small value of α_j^k indicates that g^j is close to $\partial f(x^k)$. On the other hand, the term $\gamma_1 (s_j^k)^2$ decreases the contribution of g^j to $\check{f}^k$ near x^k if g^j is nonlocal (i.e., $|y^j - x^k|$ is relatively large). The role of the third term $\gamma_2 |g^j| s_j^k$ is rather technical (see Remark 7.2 and [Mif82] for additional motivations). If f is convex, one may use $\gamma_1 = \gamma_2 = 0$, since $g^j \in \partial_\epsilon f(x^k)$ for $\epsilon = \alpha_j^k$.

Omitting locality resets corresponds to setting $a^k = 0$ at Step 1 of Algorithm 2.1. More formally, at Step 0 choose $\gamma_1 > 0$, $\gamma_2 > 0$ ($\gamma_1 = \gamma_2 = 0$ if f is convex) and $M_g \ge N + 2$, at Step 1 set $\epsilon_p^k = m_p V^k$, delete Steps 4 and 5, and at Step 9 drop the requirement $k(l) \in J^{k+1}$.

To establish convergence, replace Lemma 3.1 by Lemma 7.1.

LEMMA 7.1. *If $\liminf_{k \to \infty} \max\{\, |x^k - \bar{x}|, |p^k|, \alpha_p^k \,\} = 0$ for some $\bar{x}$ then $0 \in \partial f(\bar{x})$.*

Proof. (i) By construction and (2.11), for each k there exist $M = N+1$ (not necessarily different) points $\bar{y}^{k,i}$ and multipliers $\bar{\lambda}_i^k \geq 0$, $i = 1{:}M$, such that $(p^k, 1) = \sum_{i=1}^M \bar{\lambda}_i^k (g(\bar{y}^{k,i}), 1)$ and $\sum_{i=1}^M \bar{\lambda}_i^k \alpha(x^k, \bar{y}^{k,i}) \leq \alpha_p^k$. Hence, since $\alpha(x,y) \geq f(x) - \bar{f}(x;y)$, $p^k \in \partial_{\alpha_p^k} f(x^k)$ if f is convex [Kiw85b, p. 109]. Let $K \subset \{1,2,\ldots\}$ be such that $x^k \xrightarrow{K} \bar{x}$, $p^k \xrightarrow{K} 0$, $\alpha_p^k \xrightarrow{K} 0$ and $\bar{\lambda}_i^k \xrightarrow{K} \bar{\lambda}_i \geq 0$, $i = 1{:}M$, with $\sum_{i=1}^M \bar{\lambda}_i = 1$ because $\sum_{i=1}^M \bar{\lambda}_i^k = 1$ for all k.

(ii) First, suppose $\gamma_1 > 0$ and $\gamma_2 > 0$. By (7.1), since $\bar{\lambda}_i^k \alpha(x^k, \bar{y}^{k,i}) \xrightarrow{K} 0$, we have $\bar{\lambda}_i^k |x^k - \bar{y}^{k,i}| \xrightarrow{K} 0$, $\bar{\lambda}_i^k |g(\bar{y}^{k,i})||x^k - \bar{y}^{k,i}| \xrightarrow{K} 0$ and $\bar{\lambda}_i^k \xrightarrow{K} \bar{\lambda}_i$, so either $\bar{\lambda}_i > 0$ and $\bar{y}^{k,i} \xrightarrow{K} \bar{x}$ or $\bar{\lambda}_i = 0$ and $\bar{\lambda}_i^k g(\bar{y}^{k,i}) \xrightarrow{K'} 0$ for $K' \subset K$. Therefore, by passing to the limit in $p^k = \sum_{i=1}^M \bar{\lambda}_i^k g(\bar{y}^{k,i})$ and using the local boundedness and upper semicontinuity of ∂f, we deduce the existence of $\bar{g}^i \in \partial f(\bar{x})$, $i = 1{:}M$, such that $0 = \sum_{i=1}^M \bar{\lambda}_i \bar{g}^i$, and so $0 \in \partial f(\bar{x})$ by the convexity of $\partial f(\bar{x})$.

(iii) If f is convex then, letting $k \in K$ approach infinity in the inequality $f(x) \geq f(x^k) + \langle p^k, x - x^k \rangle - \alpha_p^k$ for all x, we get $0 \in \partial f(\bar{x})$. □

From the above proof we deduce that $0 \in \partial f(x^k)$ if $\epsilon_p^k = 0$. Moreover, with the help of Lemma 7.1, it is easy to verify all the results of §§3 and 6 (to establish Lemma 6.1 without $a^k \xrightarrow{K} 0$, use the proof of Lemma 7.1). The relevant modifications of §4 can also be incorporated.

REMARK 7.2. If we set $\gamma_2 = 0$, part (ii) of the proof of Lemma 7.1 would require boundedness of $\{g(\bar{y}^{k,i})\}_{k \in K}$, $i = 1{:}M$ (e.g., boundedness of $\{y^k\}$, since g is locally bounded). To avoid such additional assumptions, we may use the distance resetting of [Kiw85b, Chap. 3]. Namely, at Step 0 choose a large number $\bar{a} \gg \max\{\delta^1, \rho^1\}$, at Step 10 let $\rho^{k+1} \leq \bar{a}$ if $t_L^k > 0$, and drop from J^{k+1} all j with $s_j^{k+1} > \bar{a}$ if $t_L^k > 0$. Again, a simple rearrangement of the proof of Lemma 3.6 suffices for verifying all the preceding results.

For subgradient aggregation, suppose in addition to (5.1) that

$$(\tilde{s}_p^k, \hat{\beta}_p^k, \xi^k) = \sum_{j=1}^{k-1} \tilde{\lambda}_j^{k-1} (s_j^k, |g^j| s_j^k, |g^j|) \tag{7.3}$$

and we know $\tilde{s}_p^k$, ξ^k and some $\tilde{\beta}_p^k \geq \hat{\beta}_p^k$ (with $(\tilde{s}_p^1, \tilde{\beta}_p^1, \xi^1) = (0, 0, |g^1|)$). Then we may use the locality measure $\tilde{\alpha}_p^k = \max\{|f(x^k) - \tilde{f}_p^k|, \gamma_1 (\tilde{s}_p^k)^2, \gamma_2 \tilde{\beta}_p^k\}$ (cf. (7.2)) in the aggregate constraint $-\tilde{\alpha}_p^k + \langle p^{k-1}, d \rangle \leq v$ appended to (2.8). Augmenting (5.3) with

$$(s_p^k, \beta_p^k, \xi^{k+1}) = \sum_{j \in J^k} \lambda_j^k (s_j^k, |g^j| s_j^k, |g^j|) + \lambda_p^k (\tilde{s}_p^k, \tilde{\beta}_p^k, \xi^k) \tag{7.4}$$

and setting $\tilde{s}_p^{k+1} = s_p^k + |x^{k+1} - x^k|$ and $\tilde{\beta}_p^{k+1} = \beta_p^k + \xi^{k+1} |x^{k+1} - x^k|$, we get (7.3) for k increased by 1 with new $\tilde{\beta}_p^k \geq \hat{\beta}_p^k$, since $\sum_{j=1}^k \tilde{\lambda}_j^k |g^j| s_j^{k+1} = \sum_{j=1}^k \tilde{\lambda}_j^k |g^j| s_j^k + \xi^{k+1} |x^{k+1} - x^k|$ and

$$(p^k, f_p^k, s_p^k) = \sum_{j=1}^k \tilde{\lambda}_j^k (g^j, f_j^k, s_j^k) \qquad \text{and} \qquad \beta_p^k \geq \sum_{j=1}^k \tilde{\lambda}_j^k |g^j| s_j^k. \tag{7.5}$$

Next, letting $\hat{\alpha}_p^k = \max\{|f(x^k) - f_p^k|, \gamma_1 (s_p^k)^2, \gamma_2 \beta_p^k\}$, we deduce from (5.3), (7.4) and the convexity of $\phi(z) = \max\{|z_1|, \gamma_1 (z_2)^2, \gamma_2 z_3\}$ that $\hat{\alpha}_p^k \leq \alpha_p^k$ (which suffices for

proving Lemma 3.4(ii) as in [Kiw90]), whereas by the Caratheodory theorem we may suppose that exactly $N+5$ multipliers $\tilde{\lambda}_j^k$ are nonzero in (7.5) for repeating the proof of Lemma 7.1 with $\hat{\alpha}_p^k \le \alpha_p^k$. Hence the preceding results remain valid.

REMARK 7.3. If we use the distance resetting of Remark 7.2, then Step 10 must also drop the aggregate subgradient if $\tilde{a}_p^{k+1} > \bar{a}$ and $t_L^k > 0$, with $\tilde{a}_p^{k+1} = a_p^k + |x^{k+1} - x^k|$ and a_p^k generated via (5.4).

As in [DeS83, Fle87, Mor83], we may say that approximate solutions of (2.6) (which are exact for a perturbed ρ^k) yield a *restricted step* (or *trust region*) method. The alternative *Levenberg–Marquardt* approach [Fle87, Lev44, Mar63, Mor83] consists in controlling d^k indirectly by changing u^k without any explicit bound on $|d^k|$. (Thus [Kiw90, Sch89, ScZ88, ScZ92] describe Levenberg–Marquardt methods.) Our corresponding *Levenberg–Marquardt method* is obtained simply by ignoring ρ^k everywhere, i.e., using (7.2) and $\epsilon_p^k = m_p V^k$ at Step 1, and deleting Steps 4, 5, and 6. Again, distance resetting, subgradient aggregation and zigzag searches can be used, and the convergence results are not affected.

8. Implementation. We shall now discuss some heuristic choices of ρ^k and u^k related to those used in other trust region methods [DeS83, DeS89, Fle87, Mor83].

If the unit stepsize is refused by Step 7 or Step 8, i.e., $f(x^k+d^k) > f(x^k)+m_L v^k$, we can use the function and derivative information obtained at x^k and x^k+d^k to estimate the required decrease in ρ^k and increase in u^k. If we set $q(t) = f(x^k+td^k)$ then we know $q(0)$, $q(1)$ and the unknown $q'(0)$ (typically equal to $\langle g(x^k), d^k\rangle$) can be estimated by v^k (usually $v^k = \langle g(x^k), d^k\rangle$). The quadratic $\phi(t) = \phi(0) + \phi'(0)t + \phi''(0)(t)^2/2$ which interpolates these data ($\phi(0) = q(0)$, $\phi(1) = q(1)$, $\phi'(0) = v^k$) has a minimizer $t_{\rm opt} = -\phi'(0)/\phi''(0)$ with $\phi''(0) = 2[\phi(1)-\phi(0)-\phi'(0)]$, and it is reasonable to set ρ^{k+1} and u^{k+1} to

$$\rho_{\rm int}^{k+1} = t_{\rm opt}|d^k| = |d^k|/\{2(1-[f(x^k+d^k)-f(x^k)]/v^k)\}, \tag{8.1}$$

$$u_{\rm int}^{k+1} = \phi''(0)/|d^k|^2 = -2v^k(1-[f(x^k+d^k)-f(x^k)]/v^k)/|d^k|^2 \tag{8.2}$$

in most cases. Note that $u_{\rm int}^{k+1}$ is the Hessian of $\psi(t) = f(x^k+td^k/|d^k|)$ if ψ is quadratic on $[0,1]$ and $q'(0) = v^k$. In general, $f(x^k+d^k) > f(x^k)+m_L v^k$ with $m_L \in (0,1/2)$ ensures that $\rho_{\rm int}^{k+1}/|d^k| < 1/2(1-m_L) < 1$ and $u_{\rm int}^{k+1}/u^k > 2(1-m_L) > 1$.

REMARK 8.1. If we choose u in our model $\check{f}^k + u|\cdot - x^k|^2/2$ so that it agrees with f at x^k+d^k, then $u = u_{\rm int}^{k+1}$ by (2.4).

REMARK 8.2. Kiwiel [Kiw90] introduced another candidate for u^{k+1} given by

$$u_{\rm opt}^{k+1} = 2u^k(1-[f(x^k+d^k)-f(x^k)]/v^k). \tag{8.3}$$

Note that $u_{\rm int}^{k+1} \ge u_{\rm opt}^{k+1}$, $u_{\rm int}^{k+1} = u_{\rm opt}^{k+1}$ if $\alpha_p^k = 0$, and $u_{\rm int}^{k+1} \approx u_{\rm opt}^{k+1}$ if $\alpha_p^k \ll u^k|d^k|^2$ by (2.10). Also $u_{\rm opt}^{k+1}/u^k > 2(1-m_L) > 1$ if $f(x^k+d^k) > f(x^k)+m_L v^k$ with $m_L \in (0,1/2)$. If we use the aggregate linearization $\check{f}^k(x) = f(x^k) - \alpha_p^k + \langle p^k, x-x^k\rangle$ (cf. §5) and choose u in our model $\check{f}^k + u|\cdot - x^k|^2/2$ so that its minimizer $x^k - p^k/u$ agrees with $x^k + t_{\rm opt}d^k$, where $t_{\rm opt}$ is found as for (8.1), then $u = u^k/t_{\rm opt} = u_{\rm opt}^{k+1}$.

Naturally, the new values of ρ^k and u^k are obtained by projecting $\rho_{\rm int}^{k+1}$ and $u_{\rm int}^{k+1}$ on their admissible intervals, e.g., Step 7 sets $u^k := \min(\max\{1.1u^k, u_{\rm int}^{k+1}\}, 10u^k)$.

Note that (8.1) and (8.2) (or (8.3)) may also be used for increasing ρ^k and decreasing u^k if f and $\check{f}^k$ are close at x^k+d^k as measured by $f(x^k+d^k) \le f(x^k)+m_c v^k$ with

$m_c \in (1/2, 1)$. Then $\rho_{\text{int}}^{k+1}/|d^k| > 1/2(1-m_c) > 1$, $u_{\text{int}}^{k+1}/u^k < 2(1-m_c)(1+\alpha_p^k/u^k|d^k|^2)$ and $u_{\text{opt}}^{k+1}/u^k < 2(1-m_c) < 1$ if $f(x^k+d^k) > f(x^k)+v^k$. Of course, (8.1)–(8.3) break down if $\phi''(0) \le 0$, but then, suspecting negative or null curvature of f between x^k and x^k+d^k, we may simply set $\rho^{k+1} = 4\rho^k$ and $u^{k+1} = u^k/4$ to enhance larger steps.

If the line search is interrupted at a stepsize t, we may use

$$\rho_{\text{int}}^{k+1} = t|d^k|/\{2(1 - [f(x^k + td^k) - f(x^k)]/[tv^k])\},$$

$$u_{\text{int}}^{k+1} = -2v^k(1 - [f(x^k + td^k) - f(x^k)]/[tv^k])/(t|d^k|^2),$$

$$u_{\text{opt}}^{k+1} = 2u^k(1 - [f(x^k + td^k) - f(x^k)]/[tv^k])/t.$$

If Step 6 sets $u = \max\{(1+\epsilon)|d^k|/\rho^k, \chi\}u^k$ with fixed $\epsilon \ge 0$ and $\chi > 1$, then eventually $\min\{1/(1+\epsilon), 1/\chi\}\rho^k \le |d^k| \le \rho^k$ if Step 5 is not executed (see [Kiw89b]). Thus the trust region constraint is satisfied with relative accuracy of $2/3$ if $\epsilon = 1/2$ and $\chi = 1.5$ (higher accuracy appears to be unnecessary).

9. Numerical results. We shall now report on computational testing of our algorithms with double precision Fortran codes on an IBM PC clone with relative accuracy $\epsilon_M \approx 2.2 \times 10^{-16}$ ($= 2.2E{-}16$). The parameters had values $\epsilon_{\text{opt}} = 1E{-}6$, $m_L = 0.1$, $m_R = 0.5$, $m_v = 0.1$, $\bar{t} = 0.1$, $\kappa_\delta = 0.9$, $\kappa_\rho = 0.7$, $m_p = 10$, $\rho^1 = 1$, $u^1 = |g(x^1)|/\rho^1$, $u_{\min} = u^1/10^{30}$, $\delta^1 = |g(x^1)|$ and $\gamma_1 = 1$.

To save space, we only list some problems in Table 9.1; details are available in the quoted references or from the author. For constrained problems of the form

$$\min f(x), \text{ s.t. } F_i(x) \le 0, \ i = 1{:}m, \ x \in S,$$

we minimized the exact penalty function $e(x) = f(x) + c\sum_{i=1}^m \max\{F_i(x), 0\}$ over $x \in S$ with a fixed penalty coefficient $c > 0$. Table 9.1 gives values of e at the initial point x^1 and solution $\bar{x}$, with $c = 10$ for problems 4, 6–8, and 15, and $c = 10000$ for problem 18. For problem Daferm the objective $f(x) = \max_{y \in S}\langle F(x), x-y\rangle$ is the gap function for the variational inequality $\langle F(\bar{x}), y-\bar{x}\rangle \ge 0$ for all $y \in S$. The Pentagon problem consists in maximizing the minimum distance between three points inside a regular pentagon in a plane, and its objective has the form $f(x) = \max\{-|A-B|, -|B-C|, -|A-C|\}$ with $A = (x_1, x_2)$, $B = (x_3, x_4)$, $C = (x_5, x_6)$ ([Pow89] uses a smooth approximation to f).

Our RS-LR method implements the restricted step method with locality resets of §2, and the LM-LM method is the Levenberg–Marquardt method with locality measures of §7. In Table 9.2, k and NF denote the final iteration number and the total number of objective evaluations respectively. The RS-LR and LM-LM methods used the stopping criteria

$$V^k \le \epsilon_{\text{opt}}^{1/2}(1 + |e(x^k)|) \quad \text{and} \quad a^k \le \epsilon_{\text{opt}}^{1/2}(1 + |x^k|) \tag{9.1}$$

and

$$-v^k \le \epsilon_{\text{opt}}^{1/2}(1 + |e(x^k)|) \quad \text{and} \quad s_p^k \le \epsilon_{\text{opt}}^{1/2}(1 + |x^k|)/10 \tag{9.2}$$

respectively. It may be seen that for $\epsilon_{\text{opt}} = 10^{-6}$ the relative accuracy in the objective

TABLE 9.1
List of test problems.

No.	Problem	n	$e(x^1)$	$e(\bar{x})$
1	Crescent [Kiw85b, p. 349]	2	4.25	0
2	MaxRos [Kiw86a]	2	4.25	0
3	FleRos [Fle87, p. 313, $\sigma = 10$]	2	6.6	0
4	Wriggle [Pow82]	2	1	0
5	Daferm [Fuk86]	5	220800	0
6	Hock78 [HoS81, Prob. 78]	5	72.75	-2.91970041
7	Hock80 [HoS81, Prob. 80]	5	60	0.0539498478
8	Hock81 [HoS81, Prob. 81]	5	59.5	0.0539498478
9	Equil5 [LeM78, p. 158]	5	20.22	0
10	Madsen [MSJ78]	6	0.22052	0.08541755
11	Pentagon [Pow89]	6	$-\sqrt{2}$	-1.859618696
12	Equil8 [LeM78, p. 158]	8	9.7878	0
13	Filter [Cha79]	9	0.6914	0.0061853
14	Equil10 [LeM78, p. 158]	10	26.978	0
15	ShellD [HoS81, Prob. 117]	15	2400	32.348679
16	LShell [LeM78, p. 154]	15	2400	32.348679
17	U2SK1 [Lem85]	15	618725	-7550.03025
18	Dembo7 [Dem76]	16	17381	174.788807
19	U3RAM1 [Lem82a]	35	59.4464	11.99191077
20	U3RAM2 [Lem82a]	35	214.854	16.86323008
21	U3RAC1 [Lem82a]	35	10036.9	1002.689846
22	U3RAC2 [Lem82a]	35	102626	2280.512762
23	U2SK1 [Lem85]	50	3.025E+8	-7560.025

value was about five or six figures in most cases. Each line search required on average two f-evaluations; Tables 9.3 and 9.4 show that this is related to the value of $\bar{t}$. (We note for Table 9.3 that U3RAM2 has a local minimum of 21.831555447.) Table 9.5 contains results for the method with restricted steps and locality measures (RS-LM). The results given in Tables 9.2–9.5 suggest that it should be possible to develop an efficient implementation of the RS-LR method that hardly ever needs a line search.

So far, our experience with zigzag searches has been quite negative; see Table 9.6. (Incidentally, Table 9.6 suggests that the stopping criterion (9.2) may be less reliable than (9.1).) In this context we also note that in practice the zigzag search of [Sch89, ScZ92] with the safeguards of [Kiw90] improves its initial u^k (chosen as in [Kiw90]) in negligibly few cases for $N \geq 5$, i.e., the search occurs too seldom to draw conclusions about its merits, if any.

Of course our results are preliminary, but a comparison with [Bih84a, Bih84b, Kiw85a, Kiw85b, Kiw89b, Lem80, Lem82b, Sch89] suggests some potential in the methods for developing efficient implementations.

Tables 9.7–9.10 give results for the objectives multiplied by a fixed coefficient $s = 100$ and $s = 0.01$. Making such algorithms less sensitive to the objective scaling would require adaptive choices of parameters and more sophisticated stopping criteria.

TABLE 9.2
Results for the RS-LR *and* LM-LM *methods*: $\bar{t} = 0.1$.

No.	Problem	RS-LR method			LM-LM method		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	32	52	6.9E-7	25	35	1.0E-6
2	MaxRos	17	32	7.3E-6	14	22	2.1E-7
3	FleRos	22	41	8.4E-6	14	21	8.7E-14
4	Wriggle	41	62	8.8E-5	61	81	4.8E-12
5	Daferm	21	36	6.6E-9	18	25	1.1E-10
6	Hock78	32	43	-2.91964	29	44	-2.919699
7	Hock80	22	30	0.05477	172	287	0.05398
8	Hock81	23	31	0.05475	172	287	0.05398
9	Equil5	31	56	5.6E-5	44	67	4.5E-8
10	Madsen	16	31	0.085451	15	25	0.0854176
11	Pentagon	15	33	-1.85961848	17	35	-1.85961870
12	Equil8	43	84	5.6E-5	44	67	4.5E-8
13	Filter	164	304	0.006194	126	266	0.0061858
14	Equil10	56	111	1.4E-4	42	66	4.5E-8
15	ShellD	65	82	32.34900	106	197	32.348685
16	LShell	187	389	32.34883	695	1490	32.348698
17	U2SK1	32	48	-7550.015	213	460	-7550.0256
18	Dembo7	47	60	174.6488	102	274	174.6500
19	U3RAM1	51	99	11.99193	54	65	11.991912
20	U3RAM2	60	103	16.86339	76	115	16.863230
21	U3RAC1	46	85	1002.68987	60	103	1002.68985
22	U3RAC2	32	51	2280.5132	63	100	2280.5128
23	U2SK1	57	89	-7547.56	413	925	-7560.024

TABLE 9.3
Results for the RS-LR *and* LM-LM *methods*: $\bar{t} = 1$.

No.	Problem	RS-LR method			LM-LM method		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	33	33	7.4E-7	33	33	8.5E-8
2	MaxRos	55	55	1.2E-14	27	28	4.0E-10
3	FleRos	79	79	2.4E-6	41	41	7.9E-10
4	Wriggle	37	39	1.0E-4	65	74	6.8E-10
5	Daferm	48	48	6.6E-4	17	17	5.5E-11
6	Hock78	29	29	-2.91968	41	41	-2.919688
7	Hock80	34	34	0.05398	172	275	0.05396
8	Hock81	34	34	0.05399	148	228	0.05376
9	Equil5	111	111	7.0E-6	54	54	1.3E-6
10	Madsen	81	81	0.085422	26	27	0.0854176
11	Pentagon	147	148	-1.8596187	23	26	-1.8596187
12	Equil8	111	111	1.8E-5	74	74	6.0E-9
13	Filter	263	263	0.0061861	184	194	0.0061857
14	Equil10	144	144	1.9E-5	90	90	6.4E-7
15	ShellD	78	78	32.34730	83	96	32.348686
16	LShell	471	477	32.3512	585	596	32.34874
17	U2SK1	60	60	-7322	526	823	-7550.0298
18	Dembo7	62	62	174.6484	102	274	174.6500
19	U3RAM1	115	122	11.9922	58	58	11.99191
20	U3RAM2	115	122	21.831556	89	91	21.905
21	U3RAC1	99	99	1002.691	148	148	1002.68986
22	U3RAC2	58	58	2280.5131	55	55	2280.5132
23	U2SK1	73	73	-7524.4	5739	5980	-7560.023

TABLE 9.4
Results for the RS-LR *method on* LShell.

$\bar{t}$	k	NF	null steps	resets	$e(x^k)$
1	471	477	358	121	32.35123
0.1	187	383	48	58	32.34883
0.01	278	689	13	77	32.34946
0.001	234	651	6	75	32.34870
0.0001	239	658	0	98	32.34868

TABLE 9.5
Results for the RS-LM *method.*

No.	Problem	$\bar{t} = 0.1$			$\bar{t} = 1$		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	25	35	1.0E-6	33	33	8.5E-8
2	MaxRos	14	22	2.1E-7	27	28	4.0E-10
3	FleRos	14	21	8.7E-14	44	44	3.5E-7
4	Wriggle	61	81	4.8E-12	87	111	2.0E-10
5	Daferm	18	26	8.9E-7	20	20	1.1E-10
6	Hock78	28	36	-2.91990	41	41	-2.919688
7	Hock80	108	155	0.05401	179	224	0.05396
8	Hock81	103	145	0.05400	103	145	0.05400
9	Equil5	44	67	2.5E-9	46	46	1.2E-6
10	Madsen	15	21	0.0854176	26	27	0.0854176
11	Pentagon	17	35	-1.8596187	23	26	-1.8596187
12	Equil8	42	66	4.5E-8	74	74	6.0E-9
13	Filter	123	266	0.0061859	170	175	0.0061856
14	Equil10	38	74	1.1E-7	90	90	6.4E-7
15	ShellD	63	103	32.34870	56	60	32.348671
16	LShell	421	942	32.34877	891	905	32.34871
17	U2SK1	32	48	-7550.015	60	60	-7321.9
18	Dembo7	95	127	174.6454	95	127	174.6543
19	U3RAM1	54	85	11.99191	58	58	11.99191
20	U3RAM2	76	115	16.863230	86	88	21.9520
21	U3RAC1	81	130	1002.6899	88	88	1002.68988
22	U3RAC2	53	80	2280.5128	55	55	2280.5132
23	U2SK1	57	89	-7547.96	73	73	-7524.2

TABLE 9.6
Zigzag searches on LShell.

Method	$\bar{t}$	k	NF	$e(x^k)$
RS-LR	0.1	538	1084	32.3514
RS-LR	1	847	977	32.3510
LM-LM	0.1	553	1344	32.3522
LM-LM	1	554	735	32.3916
RS-LM	0.1	463	1036	32.4496
RS-LM	1	500	580	32.4279

TABLE 9.7
Results for $\bar{t} = 0.1$ and the scaling coefficient $s = 100$.

No.	Problem	RS-LR method			LM-LM method		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	26	36	1.0E-6	96	107	1.2E-8
2	MaxRos	16	27	2.2E-11	30	41	4.2E-6
3	FleRos	22	41	3.4E-9	32	42	1.4E-10
4	Wriggle	60	82	5.8E-11	125	161	6.7E-12
5	Daferm	21	36	7.4E-9	18	25	5.5E-10
6	Hock78	33	45	-2.91970	58	76	-2.91970
7	Hock80	32	42	0.05484	116	159	0.05399
8	Hock81	59	69	0.05432	101	145	0.05415
9	Equil5	23	44	5.6E-2	65	86	1.0E-6
10	Madsen	14	25	0.085471	25	33	0.085418
11	Pentagon	11	21	-1.85961328	22	29	-1.85961869
12	Equil8	54	109	2.5E-9	60	91	1.1E-10
13	Filter	192	366	0.006186	190	308	0.006185
14	Equil10	65	128	3.2E-7	40	76	4.8E-8
15	ShellD	89	129	32.34868	47	61	32.34870
16	LShell	190	392	32.34991	205	405	32.39564
17	U2SK1	38	60	-7550.015	53	102	-7550.029
18	Dembo7	39	48	174.6486	120	300	174.6488
19	U3RAM1	50	97	11.99197	86	114	11.99191
20	U3RAM2	56	103	16.86505	64	91	16.92513
21	U3RAC1	46	85	1002.68987	53	82	1002.57886
22	U3RAC2	32	51	2280.5132	72	116	2280.5128
23	U2SK1	51	81	-7559.913	91	170	-7560.024

TABLE 9.8
Results for $\bar{t} = 1$ and the scaling coefficient $s = 100$.

No.	Problem	RS-LR method			LM-LM method		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	39	39	9.2E-5	97	97	2.8E-4
2	MaxRos	44	44	4.4E-11	38	38	6.4E-11
3	FleRos	71	71	9.5E-7	117	117	1.1E-8
4	Wriggle	38	39	5.6E-7	81	82	4.2E-7
5	Daferm	66	66	1.3E-9	17	17	-7.3E-12
6	Hock78	24	24	-2.91970	39	39	-2.91970
7	Hock80	42	42	0.05395	47	47	0.05395
8	Hock81	46	46	0.05395	54	54	0.05395
9	Equil5	138	138	6.6E-7	87	87	7.5E-11
10	Madsen	56	56	0.085419	28	28	0.0854182
11	Pentagon	57	57	-1.85961811	33	33	-1.85961869
12	Equil8	130	130	4.0E-10	74	74	6.0E-9
13	Filter	227	228	0.006187	301	301	0.006185
14	Equil10	158	158	1.6E-8	101	101	1.7E-9
15	ShellD	80	80	32.34869	63	63	32.34870
16	LShell	637	648	32.35292	852	879	32.34869
17	U2SK1	67	67	-75495.65	76	76	-7550.027
18	Dembo7	33	33	174.6879	95	105	174.7784
19	U3RAM1	141	141	11.99192	96	96	11.99191
20	U3RAM2	100	104	21.83156	133	135	22.01577
21	U3RAC1	110	110	1002.69006	186	186	1002.68991
22	U3RAC2	57	57	2280.5133	54	54	2280.5132
23	U2SK1	103	103	-7560.024	212	212	-7560.024

TABLE 9.9
Results for $\bar{t} = 0.1$ and the scaling coefficient $s = 0.01$.

No.	Problem	RS-LR method			LM-LM method		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	22	32	2.0E-3	17	34	7.4E-7
2	MaxRos	14	20	1.5E-3	15	30	4.9E-7
3	FleRos	23	40	2.9E-5	12	23	3.8E-5
4	Wriggle	11	13	7.5E-1	69	138	5.0E-5
5	Daferm	21	34	4.3E-6	15	20	3.2E-5
6	Hock78	17	21	-2.91802	16	24	-2.91970
7	Hock80	4	4	0.05661	5	5	0.05525
8	Hock81	4	4	0.05607	5	5	0.05525
9	Equil5	33	65	1.3E-4	28	54	4.3E-5
10	Madsen	15	30	0.085444	42	109	0.085427
11	Pentagon	10	17	-1.85961774	34	96	-1.85961865
12	Equil8	41	82	4.6E-4	32	62	7.0E-6
13	Filter	30	61	0.013534	504	1507	0.006192
14	Equil10	46	92	3.8E-3	33	63	2.7E-6
15	ShellD	36	44	32.34922	1208	3960	32.35021
16	LShell	275	508	32.38941	2364	6151	32.34883
17	U2SK1	33	51	-7550.004	5450	13587	-7550.029
18	Dembo7	43	60	174.6485	127	362	174.6512
19	U3RAM1	52	99	11.99222	50	105	11.99191
20	U3RAM2	61	112	16.86486	41	87	16.863231
21	U3RAC1	49	91	1002.68985	42	74	1002.68992
22	U3RAC2	32	51	2280.5132	43	69	2280.5128
23	U2SK1	31	40	-7239.613	8968	24141	-7560.024

TABLE 9.10
Results for $\bar{t} = 1$ and the scaling coefficient $s = 0.01$.

No.	Problem	RS-LR method			LM-LM method		
		k	NF	$e(x^k)$	k	NF	$e(x^k)$
1	Crescent	26	26	4.7E-4	19	37	0
2	MaxRos	48	48	1.2E-4	21	31	1.1E-5
3	FleRos	83	83	3.8E-5	21	25	8.6E-8
4	Wriggle	15	15	6.9E-1	69	138	5.0E-5
5	Daferm	23	23	2.8E-3	21	21	5.7E-5
6	Hock78	37	37	-2.91472	24	24	-2.91934
7	Hock80	4	4	0.05661	5	5	0.05525
8	Hock81	4	4	0.05607	5	5	0.05525
9	Equil5	108	108	4.0E-4	53	53	1.6E-5
10	Madsen	25	25	0.085426	56	137	0.085458
11	Pentagon	27	28	-1.85958080	39	94	-1.85961866
12	Equil8	150	150	4.1E-6	99	99	2.9E-5
13	Filter	66	66	0.0123334	965	2379	0.006192
14	Equil10	140	140	8.8E-5	87	87	5.3E-5
15	ShellD	78	78	32.34875	1981	6644	32.34898
16	LShell	1800	1081	32.38260	100000	11413	32.64178
17	U2SK1	22	22	-7246.999	9126	21132	-7550.026
18	Dembo7	53	53	174.6518	127	362	174.6512
19	U3RAM1	111	111	11.99192	98	107	11.99191
20	U3RAM2	114	129	22.33489	73	75	22.10413
21	U3RAC1	95	95	1002.69047	87	87	1002.68993
22	U3RAC2	57	57	2280.5133	54	54	2280.5132
23	U2SK1	37	37	-7119.986	6249	16823	-7560.023

10. Conclusions. The main conceptual difference between our algorithms and trust region and Levenberg–Marquardt methods for simpler problems [DeS83, DeS89, Fle87, Mor83] is that our first-order model $\check{f}^k$ must be improved after a null step. First, we have to be much more careful in diminishing ρ^k and increasing u^k, since $\check{f}^k$ need not be close to f at $x^k + d^k$ for all small $|d^k| \leq \rho^k$ and large u^k. Secondly, improving $\check{f}^k$ seems, in general, to require line searches, which are readily implementable only if f is upper semidifferentiable [Bih84b, Kiw85b] (stronger hypotheses were used in [LSB81, Mif77, Mif82, PMW83, Sch89, ScZ92]). Our techniques mitigate the need for expensive line searches; yet they cannot be avoided in theory.

Our new techniques of convergence analysis expand those in [Kiw85b, Kiw90] and, hence, will hopefully be applicable in other contexts. In particular, they can be used to show that there is no need for "artificial" increases of ϵ in the ϵ-controlled method of [PMW83] (cf. [PoT82]), and that the algorithm of [Mif77] with the line searches of [Kiw85a, PMW83] is in fact globally convergent ([Mif77] gives a weaker convergence result; cf. [Kiw85b, p. 181]).

Of course, *proximal bundle methods* that employ a first-order model $\check{f}^k$ and a simple proximal term $u^k|d|^2/2$ in (2.2) cannot, in general, converge superlinearly. Some improvement might result from replacing $u^k|d|^2$ by $\langle A_k d, d\rangle$, where A_k is an $N \times N$ variable metric matrix representing the curvature of f. We expect, however, that proximal bundle methods will remain competitive with variable metric ones for large problems, where storing an $N \times N$ matrix is expensive. In contrast, our codes have coped, e.g., with problems having $N = 2000$ variables by accumulating $M_g = 30$ or 50 subgradients, and using workspace of order M_g^2 only [Kiw94]; these experiments will be reported elsewhere.

Acknowledgments. I would like to thank the two anonymous referees for their valuable comments.

REFERENCES

[Bih84a] A. BIHAIN, *Numerical and Algorithmic Contributions to the Constrained Optimization of Some Classes of Non-Differentiable Functions*, dissertation doctorale, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium, 1984.

[Bih84b] ———, *Optimization of upper semidifferentiable functions*, J. Optim. Theory Appl., 44 (1984), pp. 545–568.

[Cha79] C. CHARALAMBOUS, *Acceleration of the least p-th algorithm for minimax optimization with applications*, Math. Programming, 17 (1979), pp. 276–297.

[Cla83] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[Dem76] R. S. DEMBO, *A set of geometric programming problems and their solutions*, Math. Programming, 10 (1976), pp. 192–213.

[DeS83] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[DeS89] ———, *A view of unconstrained optimization*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy-Kan, and M. J. Todd, eds., Vol. 1 of Handbooks of Operations Research and Management Science, North-Holland, Amsterdam, 1989, pp. 1–72.

[DiG80] L. C. W. DIXON AND M. GAVIANO, *Reflections on nondifferentiable optimization, part 2, convergence*, J. Optim. Theory Appl., 32 (1980), pp. 259–275.

[DNR87] I. S. DUFF, J. NOCEDAL, AND J. K. REID, *The use of linear programming for the solution of sparse sets of nonlinear equations*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 99–108.

[Fle87] R. FLETCHER, *Practical Methods of Optimization*, second ed., John Wiley, Chichester, 1987.

[Fuk86] M. FUKUSHIMA, *A relaxed projection method for variational inequalities*, Math. Programming, 35 (1986), pp. 58–70.

[Gol77] A. A. GOLDSTEIN, *Optimization of Lipschitz continuous functions*, Math. Programming, 13 (1977), pp. 14–22.

[HaN88] J. HASLINGER AND P. NEITTAANMÄKI, *Finite Element Approximations for Optimal Shape Design*, John Wiley, Chichester, 1988.

[HoS81] W. HOCK AND K. SCHITTKOWSKI, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems 187, Springer-Verlag, Berlin, 1981.

[HUL93] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[Kiw85a] K. C. KIWIEL, *A linearization algorithm for nonsmooth minimization*, Math. Oper. Res., 10 (1985), pp. 185–194.

[Kiw85b] ———, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Mathematics 1133, Springer-Verlag, Berlin, 1985.

[Kiw86a] ———, *An aggregate subgradient method for nonsmooth and nonconvex minimization*, J. Comput. Appl. Math., 14 (1986), pp. 391–400.

[Kiw86b] ———, *A method of linearizations for linearly constrained nonconvex nonsmooth optimization*, Math. Programming, 34 (1986), pp. 175–187.

[Kiw89a] ———, *A dual method for certain positive semidefinite quadratic programming problems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 175–186.

[Kiw89b] ———, *An ellipsoid trust region bundle method for nonsmooth convex minimization*, SIAM J. Control Optim., 27 (1989), pp. 737–757.

[Kiw89c] ———, *A survey of bundle methods for nondifferentiable optimization*, in Mathematical Programming: Recent Developments and Applications, M. Iri and K. Tanabe, eds., KTT/Kluwer, Tokyo, 1989, pp. 263–282.

[Kiw90] ———, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Programming, 46 (1990), pp. 105–122.

[Kiw94] ———, *A Cholesky dual method for proximal piecewise linear programming*, Numer. Math., 68 (1994), pp. 325–340.

[Kiw95a] ———, *Approximations in proximal bundle methods and decomposition of convex programs*, J. Optim. Theory Appl., 84 (1995), pp. 529–548.

[Kiw95b] ———, *Finding normal solutions in piecewise linear programming*, Appl. Math. Optim., 32 (1995), pp. 235–254.

[Lem75] C. LEMARÉCHAL, *An extension of Davidon methods to nondifferentiable problems*, Math. Programming Stud., 3 (1975), pp. 95–109.

[LeM78] C. LEMARÉCHAL AND R. MIFFLIN, EDS., *Nonsmooth Optimization*, Pergamon Press, Oxford, 1978.

[Lem80] C. LEMARÉCHAL, *Extensions Diverses des Méthodes de Gradient et Applications*, thèse de Doctorat d'Etat, Université de Paris IX-Dauphine, Paris, 1980.

[Lem82a] ———, *Le module* U3RA, Research Report, INRIA, Rocquencourt, 1982.

[Lem82b] ———, *Numerical experiments in nonsmooth optimization*, in Progress in Nondifferentiable Optimization, E. A. Nurminski, ed., CP–82–S8, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982, pp. 61–84.

[Lem85] ———, *Le module* U2SK1, Research Report, INRIA, Rocquencourt, 1985.

[Lem86] ———, *Constructing bundle methods for convex optimization*, in Fermat Days 85: Mathematics for Optimization, J. B. Hiriart-Urruty, ed., North-Holland, Amsterdam, 1986, pp. 201–240.

[Lem89] ———, *Nondifferentiable optimization*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy-Kan, and M. J. Todd, eds., Vol. 1 of Handbooks in Operations Research and Management Science, North-Holland, Amsterdam, 1989, pp. 529–572.

[Lev44] K. LEVENBERG, *A method for the solution of certain nonlinear problems in least squares*, Quart. Appl. Math., 2 (1944), pp. 164–168.

[LSB81] C. LEMARÉCHAL, J.-J. STRODIOT, AND A. BIHAIN, *On a bundle algorithm for nonsmooth optimization*, in Nonlinear Programming 4, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1981, pp. 245–282.

[MäN92] M. M. MÄKELÄ AND P. NEITTAANMÄKI, *Nonsmooth Optimization; Analysis and Algorithms with Applications to Optimal Control*, World Scientific, Singapore, 1992.

[Mar63] D. W. MARQUARDT, *An algorithm for least squares estimation of non-linear parameters*, J. Soc. Indust. Appl. Math., 11 (1963), pp. 431–441.

[MGN87] V. S. MIKHALEVICH, A. M. GUPAL, AND V. I. NORKIN, *Nonconvex Optimization Methods*, Nauka, Moscow, 1987. (In Russian.)

[Mif77] R. MIFFLIN, *An algorithm for constrained optimization with semismooth functions*,

Math. Oper. Res., 2 (1977), pp. 191–207.

[Mif82] R. MIFFLIN, *A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization*, Math. Programming Stud., 17 (1982), pp. 77–90.

[Mor83] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art, Bonn 1982, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[MSJ78] K. MADSEN AND K. J. SCHJAER-JACOBSEN, *Linearly constrained minimax optimization*, Math. Programming, 14 (1978), pp. 208–223.

[PMW83] E. POLAK, D. Q. MAYNE, AND Y. WARDI, *On the extension of constrained optimization methods from differentiable to nondifferentiable problems*, SIAM J. Control Optim., 25 (1983), pp. 179–203.

[PoM85] E. POLAK AND D. Q. MAYNE, *Algorithm models for nondifferentiable optimization*, SIAM J. Control Optim., 23 (1985), pp. 477–491.

[PoT82] E. POLAK AND S. TISHYADHIGAMA, *New convergence theorems for a class of feasible direction algorithms*, J. Optim. Theory Appl., 37 (1982), pp. 32–43.

[Pow82] M. J. D. POWELL, *Extensions to subroutine* VF02AD, in System Modelling and Optimization, R. F. Drenick and F. Kozin, eds., Lecture Notes in Control and Information Sciences 38, Springer-Verlag, Berlin, 1982, pp. 529–538.

[Pow89] ———, TOLMIN*: A Fortran package for linearly constrained optimization calculations*, Report DAMTP 1989/NA2, DAMTP, Univ. of Cambridge, Cambridge, England, 1989.

[Sch89] H. SCHRAMM, *Eine Kombination von Bundle- und Trust-Region Verfahren zur Lösung nichtdifferenzierbarer Optimierungsprobleme*, Bayreuther Mathematische Schriften, Heft 30, Universität Bayreuth, Bayreuth, 1989.

[ScZ88] H. SCHRAMM AND J. ZOWE, *A combination of the bundle approach and the trust region concept*, in Advances in Mathematical Optimization, J. Guddat et al., eds., Vol. 45 of Mathematical Research, Akademie-Verlag, Berlin, 1988, pp. 196–209.

[ScZ92] ———, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim., 2 (1992), pp. 121–152.

[Sho79] N. Z. SHOR, *Minimization Methods for Non-Differentiable Functions*, Naukova Dumka, Kiev, 1979 (Russian). English transl., Springer-Verlag, Berlin, 1985.

[StN88] J.-J. STRODIOT AND V. H. NGUYEN, *On the numerical treatment of the inclusion* $0 \in \partial f(x)$, in Topics in Nonsmooth Mechanics, J. J. Moreau, P. D. Panagiotopoulos, and G. Strang, eds., Birkhäuser-Verlag, Basel, 1988, pp. 267–294.

[Wol75] P. WOLFE, *A method of conjugate subgradients for minimizing nondifferentiable functions*, Math. Programming Stud., 3 (1975), pp. 145–173.

SIAM J. OPTIMIZATION
Vol. 6, No. 1, pp. 250–264, February 1996
© 1996 Society for Industrial and Applied Mathematics
014

ON THE COMPLEXITY OF THE PRODUCTION-TRANSPORTATION PROBLEM*

DORIT S. HOCHBAUM† AND SUNG-PIL HONG‡

Abstract. The production-transportation problem (PTP) is a generalization of the transportation problem. In PTP, we decide not only the level of shipment from each source to each sink but also the level of supply at each source. A concave production cost function is associated with the assignment of supplies to sources. Thus the objective function of PTP is the sum of the linear transportation costs and the production costs. We show that this problem is generally NP-hard and present some polynomial classes. In particular, we propose a polynomial algorithm for the case in which the transportation cost matrix has the Monge property and the number of sources is fixed. The algorithm generalizes a polynomial algorithm of Tuy, Dan, and Ghannadan [*Oper. Res. Lett.*, 14 (1993), pp. 99–109] for the problem with two sources.

Key words. production-transportation problem, concave minimization, parametric linear programming, Monge sequence

AMS subject classifications. 68Q25, 90B10, 90C26

0. Introduction. It is known that the minimization problem over a polyhedron is polynomial when the objective function is *convex* [GLS88]. In contrast, many concave minimization problems are NP-hard. We consider here a concave minimization problem over transportation constraints called the production-transportation problem (PTP). PTP is a generalization of the transportation problem. In PTP we need to decide not only the level of shipment from each source to each sink but also the level of supply at each source. A concave production cost function is associated with the assignment of supplies to sources. The objective function is the sum of the linear transportation costs and the concave production costs.

A special class of this problem has been previously studied by Tuy, Dan, and Ghannadan [TDG93]. It was shown that when there are only two sources, the problem can be reduced to a problem of finding all breakpoints of a parametric two-source linear transportation problem with a parametrized supply level. The number of breakpoints of the parametric problem is bounded by the number of sinks and can be found in strongly polynomial time. For a two-source problem, this results in a strongly polynomial time algorithm.

We give here the proof of the NP-hardness of PTP by reducing the set cover problem to PTP. We also describe some subclasses that are solvable in polynomial time. One polynomial subclass is PTP with fixed number of sinks. Another polynomial subclass is the problem with fixed number of sources and a transportation cost matrix satisfying the *Monge* property [Hof63]. The Monge property of a matrix is recognizable in polynomial time [ACHS89]. Trivial examples of matrices with the Monge property are those with identical costs in the rows or columns of the transportation cost matrix. With such cost matrices the problem is solvable in linear time even with an arbitrary number of sources or sinks.

It was pointed out recently by Tuy et al. [TGMV93a] that a strongly polynomial algorithm was independently developed for the problem with a fixed number of sources. It was also shown that the algorithm can be used to solve the minimum concave cost network flow problem (MCCNFP) with a fixed number of sources and nonlinear arc costs in strongly polynomial time. Also, when the number of sources and nonlinear arc costs are fixed, this problem can be reduced to a PTP with a fixed number of sources in strongly polynomial time [TGMV93b].

* Received by the editors August 17, 1993; accepted for publication (in revised form) June 23, 1994. This research has been supported in part by Office of Naval Research grant N00014-91-J-1241.

† School of Business Administration and Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720 (dorit@hochbaum.berkeley.edu).

‡ Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720.

Thus the results in [TGMV93a] establish the strong polynomiality of the PTP with a fixed number of sources, which has a considerable potential of practical applications.

The algorithm of [TGMV93a], however, has no explicitly specified complexity. The algorithm presented in this paper shows that the problem is solved more efficiently when the problem additionally has the transportation cost matrix with the Monge property.

The paper is organized as follows. Section 1 presents the formulation of PTP. In §2, we prove that PTP is NP-hard and discuss some polynomial subclasses of PTP. Section 3 presents a polynomial algorithm for the problem with a transportation cost matrix satisfying the Monge property and a fixed number of sources. Finally, some open problems are presented in §4.

1. The PTP. Consider a transportation problem with a set of sources, $\{1, \ldots, m\}$ and a set of sinks, $\{1, \ldots, n\}$. Let c_{ij} be the cost of transporting a unit from i to j. The supply allocation to the sources is not prescribed but depends on a concave cost function $g(x_1, x_2, \ldots, x_m)$ for $x_1, x_2, \ldots, x_m$, the variables representing the supply levels at the m sources. The problem is to allocate the supplies to the sources and to send them to the sinks at the minimum total cost. Thus the PTP, which is formulated as follows, is a generalization of the transportation problem:

$$\text{(PTP)} \qquad \min\ g(x_1, x_2, \ldots, x_m) + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij},$$

$$\sum_{j=1}^{n} x_{ij} = x_i \quad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, \ldots, n,$$

$$x_i, x_{ij} \geq 0 \quad i = 1, \ldots, m,\ j = 1, \ldots, n.$$

It is reasonable to assume that the marginal cost of production decreases as the production level increases. The production cost function g which is concave reflects these cost economies of scale. Note that if the production system is homogeneous for all supply centers, the cost function g is symmetric with respect to the m variables, $x_1, x_2, \ldots, x_m$.

Let $B = b_1 + b_2 + \cdots + b_n$ be the total demand. PTP is feasible if and only if there exists $\mathbf{x}$ such that $x_1 + x_2 + \cdots + x_m = B$ and each x_i is nonnegative. A nonnegative vector, $\mathbf{x} \in \mathbf{R}^m$ is called a *feasible production plan* if $x_1 + x_2 + \cdots + x_m = B$.

Let $\mathbf{x}^1, \mathbf{x}^2 \in \mathbf{R}^m$ be feasible production plans. Consider the production plan, $\mathbf{x}^3 = \frac{1}{2}(\mathbf{x}^1 + \mathbf{x}^2)$. Then $\mathbf{x}^3$ is also a feasible production plan. Note that the concavity of g implies that the cost of the production plan $\mathbf{x}^3$ is at least as much as the sum of the halves of the costs of the production plan $\mathbf{x}^1$ and $\mathbf{x}^2$.

2. The complexity status of PTP. In this section, we show that PTP is NP-hard and present some polynomial subclasses. As mentioned in [TGMV93a], the NP-hardness can be observed from the fact that the minimum concave cost flow problem with a single source and m nonlinear arc costs, which is NP-hard [GP90], can be reduced to a PTP with $m + 1$ factories. Also, as pointed out in [TGMV93a], the plant location problem is a special case of PTP.

In this paper, we prove the NP-hardness of PTP more directly: we show that the set cover problem is reducible to PTP in strongly polynomial time. As a by-product of this particular reduction, we conclude that even if the problem has g as a separable function with identical component functions and the transportation cost matrix with each element zero or the same constant, PTP is still NP-hard.

2.1. PTP is NP-hard. We are unable to argue the membership of (the decision problem of) PTP in NP since g is assumed to be an arbitrary concave function and the input size of g is not well defined. We assume a computation model with a function evaluation oracle providing a single evaluation as unit operation.

We demonstrate the NP-hardness of PTP by showing that PTP is at least as hard as a known NP-hard problem—the optimization version of a known NP-complete decision problem. As g may be arbitrary, the polynomial reducibility of an NP-hard problem to PTP needs to be elaborated on. In this paper, we say that an NP-hard problem, (P), is polynomially reducible to PTP if,

(i) (P) is formulated as an instance of PTP, and

(ii) the sum of the input sizes of the numbers in the instance (excluding the concave function g) is polynomially bounded in the input size of (P).

With this definition, we prove that PTP is NP-hard by showing that the optimization problem SET COVER (see MINIMUM COVER of [GJ79]) is polynomially reducible to PTP.

SET COVER

Given a collection $K = \{S_1, S_2, \ldots, S_m\}$ of subsets of a finite set $S = \{1, 2, \ldots, n\}$. Find a subcollection K' of K of a minimum number of subsets such that the union of the subsets in K' is equal to S.

For a minimum cover K' of S, define a map from S to K' which assigns a set $S(j) \in K'$ with each element j of S so that $j \in S(j)$. Such a map exists since K' is a cover. Also the map is onto since otherwise the cover is not minimum.

SET COVER can be viewed as the problem of assigning with each element $j \in S$ a subset $S(j)$ containing j so that the total number of the subsets utilized in the assignment is minimum.

THEOREM 2.1. PTP *is* NP-*hard.*

Proof. We show that SET COVER is polynomially reducible to PTP.

For $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, let

$$x_{ij} = \begin{cases} 1 & \text{if } S_i = S(j), \\ 0 & \text{otherwise.} \end{cases}$$

Since each element is assigned with exactly one set, $\sum_{i=1}^{m} x_{ij} = 1$ for each $j = 1, \ldots, n$.

Consider the PTP with transportation costs

$$c_{ij} = \begin{cases} 0 & \text{if } j \in S_i, \\ M & \text{otherwise,} \end{cases} \tag{2.1.1}$$

for M a constant greater than $1/n$ and a concave production cost function,

$$g(x_1, x_2, \ldots, x_m) = (x_1 + 1)^{1/n} + (x_2 + 1)^{1/n} + \cdots + (x_m + 1)^{1/n}.$$

Consider the following PTP formulation with these costs. In this proof, we show that an optimal solution of SET COVER can be recovered from an optimal solution of this PTP formulation:

$$\text{(P2.1)} \qquad \min (x_1+1)^{1/n} + (x_2+1)^{1/n} + \cdots + (x_m+1)^{1/n} + \sum_{\substack{i,j \text{ s.t.} \\ j \notin S_i}} M x_{ij},$$

$$\sum_{j=1}^{n} x_{ij} = x_i \quad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n,$$

$$x_{ij} \geq 0 \quad i = 1, \ldots, m,\ j = 1, \ldots, n.$$

For this purpose, we need the following two claims.

Let $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m; \bar{x}_{11}, \bar{x}_{12}, \ldots, \bar{x}_{mn})$ be an optimal solution of (P2.1).

CLAIM 1. $\bar{x}_{ij} > 0$ *only if* $c_{ij} = 0$.

Proof. Assume the contrary: suppose there exists a pair $1 \leq p \leq m$ and $1 \leq q \leq n$ such that $\bar{x}_{pq} > 0$ and $c_{pq} = M$. Since SET COVER is feasible, there is $1 \leq p' \leq m$ such that $c_{p'q} = 0$. Consider the solution $\hat{\mathbf{x}}$ obtained from $\bar{\mathbf{x}}$ by modifying the values of x_{pq} and $x_{p'q}$ as follows:

$$x_{pq} \leftarrow 0 \quad \text{and} \quad x_{p'q} \leftarrow \bar{x}_{p'q} + \bar{x}_{pq}.$$

Since $x_{p'q}$ is increased by $\bar{x}_{pq}$, $x_{p'}$ is also increased by $\bar{x}_{pq}$. Since $(x_{p'}+1)^{1/n}$ is strictly concave, the increase in the objective function due to the increase in $x_{p'q}$ is less than

$$\frac{1}{n}(\bar{x}_{p'q}+1)^{\frac{1-n}{n}} \bar{x}_{pq} \leq \frac{1}{n} \bar{x}_{pq},$$

where

$$\frac{1}{n}(\bar{x}_{p'}+1)^{\frac{1-n}{n}}$$

is the derivative of $(x_{p'}+1)^{1/n}$ at $x_{p'} = \bar{x}_{p'}$.

On the other hand, the decrease in the objective function due to the decreasing of $\bar{x}_{pq}$ is greater than $M \bar{x}_{pq}$. Thus the net change of the objective value is less than $(\frac{1}{n} - M)\bar{x}_{pq}$.

Since $M > 1/n$ the new solution $\hat{\mathbf{x}}$ has an objective function value less than that of $\bar{\mathbf{x}}$. This contradicts the optimality of $\bar{\mathbf{x}}$ for (P2.1). Thus we conclude that $\bar{x}_{ij} > 0$ only if $c_{ij} = 0$. □

CLAIM 2. $\bar{x}_i$ *is integer for every* $i = 1, \ldots, m$.

Proof. Suppose there is $1 \leq r \leq m$ such that $\bar{x}_r$ is not an integer. Then there exists an $1 \leq s \leq n$ such that $\bar{x}_{rs}$ is a fraction. This implies that there is an $1 \leq r' \leq m$, $r' \neq r$ such that $\bar{x}_{r's}$ is a fraction.

Assume that $\bar{x}_{r'} \leq \bar{x}_r$. Consider the feasible solution $\mathbf{x}'$ obtained by setting $x'_{r's} = 0$ and $x'_{rs} = \bar{x}_{rs} + \bar{x}_{r's}$.

Since the separable terms for $x_{r'}$ and x_r in the concave function g are identical and strictly concave, the assumption, $\bar{x}_{r'} \leq \bar{x}_r$, implies that the objective function value of the new solution $\mathbf{x}'$ is less than that of the optimal solution $\bar{\mathbf{x}}$; a contradiction.

Similarly, when $\bar{x}_r \leq \bar{x}_{r'}$ we can derive a contradiction by showing that decreasing x_{rs} to zero and increasing $x_{r's}$ from $\bar{x}_{r's}$ to $\bar{x}_{r's} + \bar{x}_{rs}$ decreases the objective value. Hence $\bar{x}_i$ is an integer for every $i = 1, \ldots, m$. □

Now we are ready to show how to derive an integral optimal solution of (P2.1) given optimal solution $\bar{\mathbf{x}}$. Consider the following transportation problem:

$$\text{(P2.2)} \qquad \min \sum_{\substack{i,j \text{ s.t.} \\ j \notin S_i}} M x_{ij},$$

$$\sum_{j=1}^{n} x_{ij} = \bar{x}_i \quad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n,$$

$$x_{ij} \geq 0 \quad i = 1, \ldots, m,\ j = 1, \ldots, n.$$

Notice from Claim 1 that $(\bar{x}_{11}, \bar{x}_{12}, \ldots, \bar{x}_{mn})$ is also an optimal solution of (P2.2). $\bar{x}_i$ are integers by Claim 2 and hence we can find an integral optimal solution, $\mathbf{x}^* = (x^*_{11}, x^*_{12}, \ldots, x^*_{mn})$ in polynomial time by using, for example, the minimum cost flow algorithm of [Orlin93]. Furthermore, $\mathbf{x}^*$ and $\bar{\mathbf{x}}$ have the same objective value for (P2.1) as $x^*_i = \bar{x}_i$ for every $i = 1, \ldots, m$ and the value of the linear cost term is zero for both solutions. So $(x^*_1, x^*_2, \ldots, x^*_m, x^*_{11}, x^*_{12}, \ldots, x^*_{mn})$ is an integral optimal solution of (P2.1). Also by applying Claim 1 to $\mathbf{x}^*$ and from (2.1.1), $\{x^*_{ij}\}$ is a feasible solution of SET COVER.

So far we have shown that a feasible solution of SET COVER can be derived from any optimal solution of (P2.1) in (strongly) polynomial time. To complete the proof, it remains to show that an integral optimal solution of (P2.1) is also an optimal solution of SET COVER. To show this, it suffices to prove that g is a strictly monotonic increasing function in the size of a cover. Notice that the size of a cover is equal to the number of *positive* (integer) elements in $(x_1, x_2, \ldots, x_m)$.

LEMMA 2.2. *Let p be an integer satisfying $0 < p < \min\{m, n\}$. Let $(x_1, x_2, \ldots, x_m)$ be a feasible solution of* (P2.1) *with p positive elements and let $(x'_1, x'_2, \ldots, x'_m)$ be a feasible solution of* (P2.1) *with $p+1$ positive elements. Then,*

$$(x_1+1)^{1/n}+(x_2+1)^{1/n}+\cdots+(x_m+1)^{1/n} < (x'_1+1)^{1/n}+(x'_2+1)^{1/n}+\cdots+(x'_m+1)^{1/n}.$$

Proof. Consider the following two problems:

$$\text{(P2.3)} \qquad P = \max\,(y_1+1)^{1/n} + (y_2+1)^{1/n} + \cdots + (y_p+1)^{1/n},$$

$$y_1 + y_2 + \cdots + y_p = n,$$

$$y_i \geq 1 \text{ integer}, \quad i = 1, \ldots, p.$$

$$\text{(P2.4)} \qquad Q = \min\,(y_1+1)^{1/n} + (y_2+1)^{1/n} + \cdots + (y_{p+1}+1)^{1/n},$$

$$y_1 + y_2 + \cdots + y_{p+1} = n,$$

$$y_i \geq 1 \text{ integer}, \quad i = 1, \ldots, p+1.$$

The vector consisting of the positive elements of $(x_1, x_2, \ldots, x_m)$ is feasible for (P2.3). Hence,

$$(x_1+1)^{1/n} + (x_2+1)^{1/n} + \cdots + (x_m+1)^{1/n} \leq P.$$

Similarly,

$$(x_1' + 1)^{1/n} + (x_2' + 1)^{1/n} + \cdots + (x_m' + 1)^{1/n} \geq Q.$$

In order to complete the proof of the lemma, we need to show that $P < Q$.

Since (P2.3) is symmetric with respect to its variables and the objective is concave, the optimal solution to the relaxed problem has all variables equal to n/p. Therefore,

$$P \leq p\left(\frac{n}{p} + 1\right)^{1/n}.$$

Equation (P2.4) is also symmetric but it involves minimization. Hence the optimal solution of the continuous version of (P2.4) is achieved at any vertex, i.e., any solution vector with p 1's and a single $n-p$ as its entry. Thus,

$$Q \geq p\,2^{1/n} + (n - p + 1)^{1/n}.$$

To show $P < Q$, it suffices to prove

$$p\left(\frac{n}{p} + 1\right)^{1/n} < p\,2^{1/n} + (n - p + 1)^{1/n}, \tag{2.1.2}$$

or, equivalently,

$$\left(p\left(\frac{n}{p} + 1\right)^{1/n}\right)^n < (p\,2^{1/n} + (n - p + 1)^{1/n})^n.$$

Since

$$\left(p\left(\frac{n}{p} + 1\right)^{1/n}\right)^n = p^n + n\,p^{n-1},$$

(2.1.2) is equivalent to

$$(p\,2^{1/n} + (n - p + 1)^{1/n})^n > p^n + n\,p^{n-1}. \tag{2.1.3}$$

But

$$\begin{aligned}
&(p\,2^{1/n} + (n - p + 1)^{1/n})^n \\
&\quad = 2p^n + \binom{n}{n-1} p^{n-1} 2^{(n-1)/n} (n - p + 1)^{1/n} + \cdots \\
&\qquad + \binom{n}{1} p\,2^{1/n} (n - p + 1)^{(n-1)/n} + (n - p + 1) \\
&\quad > p^n + \binom{n}{n-1} p^{n-1} 2^{(n-1)/n} (n - p + 1)^{1/n} \\
&\quad = p^n + n\,p^{n-1} 2^{(n-1)/n} (n - p + 1)^{1/n}.
\end{aligned}$$

Since $n - p + 1 > 1$,

$$p^n + n\,p^{n-1} 2^{(n-1)/n} (n - p + 1)^{1/n} \geq p^n + n\,p^{n-1}.$$

Hence (2.1.3) follows and the proof of Lemma 2.2, and hence of Theorem 2.1, is complete. □

From the above reduction it follows that PTP is NP-hard even when g is separable and symmetric (i.e., every component function is identical) and each element of the transportation cost matrix is zero or a same constant.

2.2. Two polynomial classes of PTP.

2.2.1. PTP with a fixed number of sinks. Consider the problem PTP as in §1. Since $g(x_1, x_2, \ldots, x_m)$ is a concave function and $x_i = \sum_{j=1}^{n} x_{ij}$ is a linear function, g is a concave function of $\{x_{ij}\}$. Hence the objective is a concave function of $\{x_{ij}\}$ and an optimal solution is attained at a vertex of the polyhedron defined by the last n constraints in $\{x_{ij}\}$ and the nonnegative restrictions on $\{x_{ij}\}$. Thus to solve the problem, it suffices to enumerate all vertices of the polyhedron and choose a vertex with the maximum objective value.

Each vertex of the polyhedron corresponds to a set of n independent columns of the matrix defined by the n constraints. So in a vertex, there is only one nonzero variable, say x_{i_j}, from m variables of jth constraint where $j = 1, 2, \ldots, n$. (In other words, the polyhedron is the product, $\prod_{j=1}^{n} S_j$, where S_j is the $m-1$ dimensional simplex determined by the intersection of the hyperplane, $\sum_{i=1}^{m} x_{ij} = b_j$ and the nonnegative orthant, $x_{ij} \geq 0$.) Hence there are $O(m^n)$ vertices and each vertex is of the form

$$x_{ij} = \begin{cases} b_j & \text{if } i = i_j, \\ 0 & \text{if } i \neq i_j \end{cases}$$

for some i_j for each $j = 1, \ldots, n$.

Thus, when the number of sinks, n is fixed, PTP is solvable in polynomial time.

2.2.2. PTP with identical costs in rows or columns. Consider the problem PTP with a transportation cost matrix in which the costs are identical in each row. That is, for each $i = 1, 2, \ldots, m$, there is a constant c_i such that $c_{ij} = c_i$ for all $j = 1, 2, \ldots, n$. The objective function is written as $g(x_1, x_2, \ldots, x_m) + \sum_{i=1}^{m} c_i x_i$, which is concave with respect to $(x_1, x_2, \ldots, x_n)$. So an optimal solution is attained at a vertex of the polyhedron of PTP. In this case, the polyhedron is the simplex which is the intersection of the hyperplane $\sum_{i=1}^{m} x_i = B\,(= b_1 + b_2 + \cdots + b_n)$ and the nonnegative orthant $x_i \geq 0$. Hence each vertex is of the form,

$$(*) \qquad x_i^* = \begin{cases} B & \text{if } i = i', \\ 0 & \text{if } i \neq i' \end{cases}$$

for some $1 \leq i' \leq m$. Note that there are $O(m)$ vertices and hence $O(m)$ function evaluations are enough to solve the problem.

A similar observation applies when the costs are identical along each column. Namely, for each $j = 1, 2, \ldots, n$, there is a constant c_j such that $c_{ij} = c_j$ for all $i = 1, 2, \ldots, m$. In this case the linear cost term is the constant

$$\sum_{j=1}^{n} c_j \left(\sum_{i=1}^{m} x_{ij} \right) = \sum_{j=1}^{n} c_j b_j$$

and hence the objective function is again concave in $(x_1, x_2, \ldots, x_n)$. The optimal solution is again of the form (*) and can be determined in linear time. A special case of the above polynomial class is when all c_{ij} are identical, $c_{ij} = c$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Therefore the problem is easy when the linear term matrix has a special structure so that the objective function is concave with respect with the variables, $x_1, x_2, \ldots, x_m$. Thus the general structure of the linear term in the objective function appears to be an important factor in the "hardness" of the problem. For example, as we have seen in the previous subsection, even when all elements of the cost matrix are zero or one, if in arbitrary positions, it is enough to make the problem NP-hard.

3. A strongly polynomial algorithm for PTP with the Monge cost matrix and a fixed number of sources. In this section, we show that if the transportation cost matrix (c_{ij}) has the *Monge* property and m is fixed, then PTP is solvable in strongly polynomial time.

We show first that PTP is solvable by enumerating the breakpoints of a multiparametric transportation problem (§3.1). Section 3.2 has a discussion and definition of the *Monge* property. In §3.3, we show that the breakpoints of the multiparametric transportation defined in §3.1, can be enumerated in strongly polynomial time if the transportation cost matrix has the Monge property and m is fixed.

The approach here could be viewed as a generalization of the approach in [TDG92] for $m = 2$, as matrices with only two rows (or two columns) always satisfy the Monge property (see, e.g., [ACHS89]).

3.1. A parametric problem. Consider the parametric transportation problem

$$\text{(TP}(\boldsymbol{\alpha})\text{)} \qquad z(\boldsymbol{\alpha}) = \min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij},$$

$$\sum_{j=1}^{n} x_{ij} = \alpha_i \quad i = 1,\dots,m,$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1,\dots,n,$$

$$x_{ij} \geq 0 \quad i = 1,\dots,m,\ j = 1,\dots,n,$$

where each α_i is nonnegative and $\alpha_1 + \cdots + \alpha_m = B$ (where, $B = b_1 + \cdots + b_m$).

It is known that on the domain $D = \{\boldsymbol{\alpha} : \alpha_i \geq 0 \text{ for } i = 1, 2, \dots, m,\ \alpha_1 + \cdots + \alpha_m = B\}$, the optimal value of (TP($\boldsymbol{\alpha}$)), $z(\boldsymbol{\alpha})$, is a piecewise affine function (see, e.g., [Murty83]). D is the union of polyhedral subdomains, $D_1, D_2, \dots, D_M$, so that $z(\boldsymbol{\alpha})$ is linear on each of the subdomains. Note that the subdomains have disjoint relative interiors.

Let $w(\boldsymbol{\alpha}) = g(\boldsymbol{\alpha}) + z(\boldsymbol{\alpha})$ (where g is the concave cost production function of PTP). Then the optimal value of PTP is equal to

$$\min_{\boldsymbol{\alpha}\in D} w(\boldsymbol{\alpha}) = \min_{k=1,\dots,M} \min_{\boldsymbol{\alpha}\in D_k} w(\boldsymbol{\alpha}). \tag{3.1}$$

Note that D and its subdomains D_k are (bounded) polytopes. Hence there is a finite set of points of D, $L = \{\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \dots, \boldsymbol{\alpha}^{\mathbf{N}}\}$ such that each subdomain D_k is the convex hull of a subset L_k of L. Each element of L is called a breakpoint of the parametric transportation problem, (TP($\boldsymbol{\alpha}$)).

Notice that $w(\boldsymbol{\alpha})$ is a concave function on each D_k since $z(\boldsymbol{\alpha})$ is affine on each D_k. So $\min_{\boldsymbol{\alpha}\in D_k} w(\boldsymbol{\alpha})$ can be determined by evaluating function values of $w(\boldsymbol{\alpha})$ at the breakpoints $\boldsymbol{\alpha} \in L_k$ and choosing a breakpoint giving the smallest value. From (3.1), the optimal objective function value of PTP is obtained by evaluating function values of $w(\boldsymbol{\alpha})$ at all the breakpoints of L and choosing a breakpoint giving the smallest value.

Thus, to solve PTP, it suffices to find all breakpoints and the corresponding solution of (TP($\boldsymbol{\alpha}$)).

The parametric approach does not necessarily lead to an algorithm for the problem, unless it is known how to find all the breakpoints. Tuy, Dan, and Ghannadan [TDG92] showed that for $m = 2$, (TP($\boldsymbol{\alpha}$)) has at most $n + 1$ breakpoints that can be found in $O(n \log n)$ elementary

operations. As discussed in §2.1, PTP has 2^n vertices when $m = 2$. Thus the parametric approach represents a substantial improvement.

Although it is not explicitly observed in [TDG92], the reason for this efficiency is that every $2 \times n$ cost matrix has the *Monge* property [ACHS89] which is discussed in the following section.

Remark. In [TDG92], for $m = 2$ the parametric transportation problem is derived in the context of *rank* 2 *condition*. It is also possible to derive (TP(α)) using *rank m condition*, which is a straightforward generalization of rank 2 condition. Using rank m condition, we can show that a more general class of concave minimization problem can be reduced into the problem enumerating the breakpoints of a parametric linear problem.

3.2. The Monge property. Consider the transportation problem (TP) defined with the cost matrix $C = (c_{ij})$:

$$\text{(TP)} \qquad \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij},$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, \ldots, n,$$

$$x_{ij} \geq 0 \quad i = 1, \ldots, m,\ j = 1, \ldots, n.$$

C is said to have the *Monge property* [Hof63] if there exists a permutation $((i_1, j_1), (i_2, j_2), \ldots, (i_{mn}, j_{mn}))$ of indices of the cost matrix such that

(M) for every $1 \leq i, k \leq m$, $1 \leq j, l \leq n$, whenever (i, j) precedes both $(i, l)$and (k, j) the corresponding entries in matrix C are such that $c_{ij} + c_{kl} \leq c_{il} + c_{kj}$.

Any permutation satisfying (M) is called a *Monge sequence*. If C has the Monge property (and hence admits a Monge sequence), for any nonnegative integers, $a_1, \ldots, a_m$, $b_1, \ldots, b_n$ (with $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$), (TP) is solved efficiently by a greedy algorithm. The following theorem is due to Hoffman [Hof63].

THEOREM 3.1. *A permutation* $((i_1, j_1), (i_2, j_2), \ldots, (i_{mn}, j_{mn}))$ *is a Monge sequence of* C *if and only if for any nonnegative integers,* $a_1, \ldots, a_m$, $b_1, \ldots, b_n$ *with* $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, *the solution obtained by Algorithm Greedy* (in Figure 1) *is an optimal solution of* (TP).

The Monge property can be characterized in polynomial time. The algorithm of [ACHS89] tests whether an $m \times n$ matrix has a Monge sequence in $O(m^2 n \log n)$.

Every $2 \times n$ matrix, C has a Monge sequence: renumber the columns of the matrix so that $c_{11} - c_{21} \leq c_{12} - c_{22} \leq \cdots \leq c_{1n} - c_{2n}$. Then $((1, 1), (1, 2), \ldots, (1, n), (2, 1), (2, 2), \ldots, (2, n))$ is a Monge sequence as can be easily checked. The parametric transportation problem for $m = 2$ follows Algorithm Greedy.

ALGORITHM GREEDY

```
begin
    u_i ← a_i for i = 1, ..., m;  v_j ← b_j for j = 1, ..., n;
    for k = 1 to mn do
    begin
            x_{i_k j_k} ← min{u_{i_k}, v_{j_k}};
            u_{i_k} ← u_{i_k} − x_{i_k j_k};
            v_{i_k} ← v_{i_k} − x_{i_k j_k};
    end; {for}
end; {Algorithm}
```

FIG. 1. *Algorithm Greedy.*

$$(\mathrm{TP}_2(\beta)) \qquad \min \sum_{i=1}^{2} \sum_{j=1}^{n} c_{ij} x_{ij},$$

$$\sum_{j=1}^{n} x_{1j} = \beta,$$

$$\sum_{i=1}^{2} x_{ij} = b_j \quad j = 1, \ldots, n,$$

$$x_{ij} \geq 0 \quad i = 1, 2,\ j = 1, \ldots, n,$$

for $0 \leq \beta \leq B$ (only a single parameter is required by deleting a redundant constraint).

THEOREM 3.2 (see [TDG92]). (i) *The breakpoints of the optimal value function in* $(\mathrm{TP}_2(\beta))$ *are in the set,* $\{\beta^0, \beta^2, \ldots, \beta^n\}$, *where*

$$\beta^0 = 0,\ \beta^{k+1} = \beta^k + b_{k+1},\ k = 0, 1, \ldots, n-1.$$

(ii) *If* $\sum_{j=0}^{l-1} b_j \leq \beta \leq \sum_{j=0}^{l} b_j$ (*with* $b_0 = 0$) *for some* $0 \leq l \leq n-1$, *then the following solution is optimal for* $(\mathrm{TP}_2(\beta))$:

$$x_{11} = b_1,\ x_{21} = 0; \ldots; x_{1l-1} = b_{l-1}, x_{2l-1} = 0;$$

$$x_{1l} = \beta - \sum_{j=0}^{l-1} b_j, x_{2l} = \sum_{j=0}^{l} b_j - \beta;$$

$$x_{1l+1} = 0, x_{2l+1} = b_{l+1}; \ldots; x_{1n} = 0, x_{2n} = b_n.$$

The algorithm presented in the following section reduces a parametric problem into a two-row problem. We then use the algorithm implied by Theorem 3.2 to solve the two-row problem.

ALGORITHM PTP

begin

1 set $I = \{1, 2, \ldots, m\}$ and $J = \{1, 2, \ldots, n\}$;

2 $Q \leftarrow \{\alpha_1 \geq 0,\ \alpha_2 \geq 0, \ldots, \alpha_{m-1} \geq 0,\ \alpha_1 + \alpha_2 + \cdots + \alpha_{m-1} \leq B\}$;

3 set $u_i = \alpha_i$ for $i = 1, \ldots, m-1$, $u_m = B - (\alpha_1 + \alpha_2 + \cdots + \alpha_{m-1})$ and $v_j = b_j$ for $j = 1, \ldots, n$;

4 call Procedure Breakpoint-Finder (I,J);

5 evaluate the objective function value for the solution corresponding to each breakpoint;

6 choose a solution of minimum value as an optimal solution of PTP;

end;

FIG. 2. *Algorithm* PTP.

3.3. An algorithm. Consider Algorithm PTP presented in Figure 2. It consists of three main parts: initialization (lines 1–3), breakpoint enumeration for (TP($\boldsymbol{\alpha}$)) (line 4) which is done by the procedure Breakpoint-Finder described in Figure 3, and optimization (lines 5–6), which selects a breakpoint and a corresponding solution yielding the smallest objective value.

From the arguments of §3.1 the validity of Algorithm PTP immediately follows from the validity of the procedure Breakpoint-Finder.

The procedure Breakpoint-Finder applies Algorithm Greedy to the parametric problem (TP($\boldsymbol{\alpha}$)) using the Monge sequence. At each iteration, Algorithm Greedy needs to be combined with a branching step to enumerate the possible outcomes of optimal solution as the right-hand sides are parametrized in terms of $\boldsymbol{\alpha}$.

At each branching step, the current problem branches into two subproblems of smaller dimensions. This branching is recursively repeated until the current problem has only two rows. Then the two-row parametric problem is solved using Theorem 3.1.

THEOREM 3.3. *Procedure Breakpoint-Finder correctly returns all the breakpoints of* (TP($\boldsymbol{\alpha}$)).

Proof. At each iteration the procedure considers the cell (i, j) of the current problem, which is at the head of the Monge sequence (line 1). By Theorem 3.1, the greedy optimal solution is such that

$$x_{ij} = \begin{cases} u_i & \text{if } u_i \leq v_j, \\ v_j & \text{if } u_i \geq v_j. \end{cases} \tag{3.3.1}$$

Thus we need to consider two cases, $u_i \leq v_j$ and $u_i \geq v_j$. When $u_i \leq v_j$, we assign u_i to the cell (i, j) and the equation $u_i \leq v_j$ is added to Q to specify the subdomains on which the assignment is optimal (line 3).

Also if $u_i \leq v_j$, all cells other than (i, j) of the row i will not be assigned any positive shipment in the greedy solution. Hence the row i is deleted from further consideration and the demand level of the column j needs to be decreased to $v_j - u_i$ (line 4).

Lines 6–8 describe the analog procedure for the case, $u_i \geq v_j$.

After the branching step, the current problem branches into two subproblems; the one with reduced rows, and the other with reduced columns if the number of columns is larger than 1. For each subproblem, the branching is recursively repeated if the number of rows is > 2 (lines 5 and 8).

PROCEDURE BREAKPOINT-FINDER(I,J)

```
begin
    if |I| > 2 then do
    begin
1       select the cell (i,j) such that i ∈ I and j ∈ J, which is
        at the top of the Monge sequence;
2       delete (i,j) from the Monge sequence;
        {row deletion}
3       let x_ij = u_i; Q ← Q ∪ {u_i ≤ v_j};
4       I ← I − {i}; v_j ← v_j − u_i;
5       call Procedure Breakpoint-Finder (I,J);
        {column deletion}
        if |J| > 1, then do
        begin
6           let x_ij = v_j; Q ← Q ∪ {v_j ≤ u_i};
7           J ← J − {j} ; u_i ← u_i − v_j;
        end; {if}
8       else, call Procedure Breakpoint-Finder (I,J);
    end; {if}
    else, do
    begin {solving two-row problem}
9       say I = {1,2} and J = {1,2,...,k};
10      sort {c11 − c21, c12 − c22, ..., c1k − c2k} in an increasing order
        {we assume, for the simplicity, that the sequence is already sorted} ;
        for p = 1 to k do
        begin
11          set x11 = v11, x21 = 0, ..., x1p−1 = v1p−1, x2p−1 = 0, x1p = u1 − Σ_{q≤p−1} v_q,
            x2p = Σ_{q≤p} v_q − u1, x1p+1 = 0, x2p+1 = vp+1, ..., x1k = 0, x2k = vk;
12          Q ← Q ∪ {Σ_{q≤p−1} v_q ≤ u1, u1 ≤ Σ_{q≤p} v_q}, where v0 = 0;
            {calculation of breakpoints}
13          solve every subset of m − 1 equations of Q;
14          accept a solution (α1, α2, ..., α_{m−1}) as a breakpoint if
            α_i ≥ 0 for all i and α1 + α2 + ··· + α_{m−1} ≤ B;
15          return the accepted breakpoints and the
            corresponding solution {x_ij};
        end; {for}
    end; {else}
end; {Procedure}
```

FIG. 3. *Procedure Breakpoint-Finder.*

Note that u_i and v_j are affine functions of $\boldsymbol{\alpha}$. Initially $u_i = \alpha_i$ and $v_j = b_j$. By induction it is easy to see that at each iteration, $u_i = (\rho_{i1}\alpha_1 + \rho_{i2}\alpha_2 + \cdots + \rho_{im}\alpha_m) - \rho_{i0}$ and $v_j = \sigma_{j0} - (\sigma_{j1}\alpha_1 + \sigma_{j2}\sigma_2 + \cdots + \sigma_{jm}\alpha_m)$, where ρ_{ik} and σ_{jl} are nonnegative integers for $k, l = 0, \ldots, m$.

So the linear equation, $u_i = v_j$ defines a hyperplane that divides the domain $D = \{\boldsymbol{\alpha} : \alpha_i \geq 0 \text{ for } i = 1, 2, \ldots, m-1,\ \alpha_1 + \cdots + \alpha_{m-1} \leq B\}$ into two polyhedral subdomains.

Clearly on each subdomain the greedy solution of x_{ij} defined in (3.3.1) is an affine function of $\boldsymbol{\alpha}$. Hence the cost contribution of the greedy solution of x_{ij} to the objective value $z(\boldsymbol{\alpha})$ is also an affine function of $\boldsymbol{\alpha}$ on each subdomain. Thus by induction on the number of iterations, it follows that the cost of the currently assigned shipment levels (x_{ij}'s with $(i,j) \notin I \times J$) is an affine function of $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ in the current subdomain defined by the equations in Q.

So far we have shown that at each branching iteration, the procedure assigns the shipment level to an additional cell in an optimal manner and generates the equations in $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$ that describe the region on which the additional assignment is optimal and the cost incurred is an affine function of $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$. The equations are then added to Q. Thus the equations in Q describe a polyhedral subset of $D = \{\boldsymbol{\alpha} : \alpha_i \geq 0 \text{ for } i = 1, 2, \ldots, m-1\}$, on which the current greedy solution is optimal and the cost of the solution is an affine function.

This branching is repeated until the number of remaining rows becomes 2 (line 9). The optimal parametric solution can then be found using Theorem 3.2. The solution (line 11) and the equations describing corresponding ranges (line 12) are obtained from Theorem 3.2 (ii) by simply substituting $n = k$, $\beta = u_1$, and $b_l = v_l$ for $l = 1, \ldots, k$. If there are k columns in the problem, there are k corresponding ranges. Clearly, in each of the ranges the cost incurred by the shipment levels assigned to the cells in the two-row problem is an affine function of the $(\alpha_1, \alpha_2, \ldots, \alpha_{m-1})$.

After line 11, we obtain a complete set of parametric solution $\{x_{ij} \mid i = 1, \ldots, m, j = 1, \ldots, n\}$ of (TP($\boldsymbol{\alpha}$)). The equations in Q after the execution of line 12 describe a complete subdomain on which the objective function value $z(\boldsymbol{\alpha})$ of the above solution is an affine function.

Note that the total number of equations in Q is at most $m+n$. Thus each subdomain has at most $\binom{m+n}{m-1}$ vertices (breakpoints of (TP($\boldsymbol{\alpha}$)) which can be found by solving every subset of $m-1$ equations of Q as in line 13. In line 14, we choose only the solutions that satisfy the valid conditions, namely, each α_i is nonnegative and $\alpha_1 + \cdots + \alpha_{m-1} \leq B$.

Thus the procedure Breakpoint-Finder correctly generates the complete set of breakpoints of (TP($\boldsymbol{\alpha}$)) and the corresponding solutions. The theorem follows. □

From §3.1 and Theorem 3.3, we have Corollary 3.4.

COROLLARY 3.4. *Algorithm* PTP *is valid.*

Now we show that Algorithm PTP is strongly polynomial.

THEOREM 3.5. *Assume that there is a function evaluation oracle that provides a single evaluation as unit operation. Then the complexity of* PTP *is* $O(n^m)$.

Proof. The initialization (lines 1–3) of the algorithm can be done in $O(n)$ steps as can be easily verified. The number of unit operations required for the optimization (lines 5–6) is linear in the number of breakpoints generated by the procedure Breakpoint-Finder: the number of function evaluations (line 5) is the same as the number of breakpoints and the minimum value of line 6 can be found in linear time in the number of function values by using the algorithm in [BFPRT72].

The number of breakpoints generated by the procedure Breakpoint-Finder is bounded by the number of elementary arithmetic operations executed by Breakpoint-Finder.

So to prove the theorem, it suffices to show that the running time of Breakpoint-Finder is $O(n^m)$.

Let $T(m,n)$ be the number of elementary operations taken by the procedure Breakpoint-Finder when a parametric problem has m rows and n columns. Then by the recursive structure of the procedure, we have

$$T(m,n) \leq T(m-1,n) + T(m,n-1). \tag{3.3.2}$$

Applying (3.3.2) repeatedly, we get

$$\begin{aligned}
&T(m-1,n)+T(m,n-1)\\
&\quad\le T(m-1,n)+T(m-1,n-1)+T(m,n-2)\\
&\quad\le T(m-1,n)+T(m-1,n-1)+T(m-1,n-2)+T(m,n-3)\\
&\quad\le T(m-1,n)+T(m-1,n-1)+\cdots+T(m-1,2)+T(m,1)\\
&\quad\le (n-1)T(m-1,n)+T(m,1).
\end{aligned}$$

To evaluate $T(m,1)$, note that for a single column problem, the algorithm performs $O(m)$ unit operations. The initialization takes $O(m)$ operations. There are only $m-1$ row deletions from lines 1–8 of Procedure Breakpoint-Finder. The remaining steps of lines 9–10 require $O(m)$ operations. Thus there exists a constant $A>0$ such that,

$$T(m,n)\le (n-1)T(m-1,n)+Am. \tag{3.3.3}$$

We now show that the number of elementary operations required to solve the two-row problem is $T(2,n)=O(n^2)$. When the number of rows is 2 ($|I|=2$), after the initialization which takes $O(n)$ operations, the algorithm performs the steps of lines 9–15 in Procedure Breakpoint-Finder. The sorting in line 10 can be done in $O(n\log n)$ steps. For each $p=1,\dots,k$, lines 11 and 12 can be done in $O(n)$ steps. The operations of lines 13 and 14 can be done in $O(n)$ steps since, as shown in the proof of Theorem 3.3, when m is fixed, the total number of the sets of equations to be solved is $\binom{m+n}{m-1}=O(n^{m-1})$. Each set of equations is solvable in $O(m^3)$ which is a constant as m is fixed.

Thus for each $p=1,\dots,k$, the complexity is $O(n)$ and hence $T(2,n)=O(n^2)$ as $k\le n$.

Assume, by induction, that for every $k=2,\dots,m-1$, $T(k,n)=Bk^2n^k$, where B is a constant with $B>A$. Then by (3.3.3),

$$\begin{aligned}
T(m,n)&\le (n-1)T(m-1,n)+Am\\
&\le (n-1)B(m-1)^2n^{m-1}+Am\\
&\le nB(m-1)^2n^{m-1}+Am\\
&= Bm^2n^m-2Bmn^{m+}Bn^m+Am\\
&< Bm^2n^m-2Bmn^{m+}Bn^m+Bm\ (\text{since } B>A)\\
&< Bm^2n^m.
\end{aligned}$$

Hence the theorem follows. □

Remark. A careful look at the algorithm and its complexity analysis indicates that we may assume a weaker form of oracle. The function evaluations are required solely to find a solution yielding the smallest objective value. However a smallest valued solution can be found without calculating function values explicitly if only we can determine the ordinal sizes of the function values of given solutions used in comparisons (see, e.g., [BFPRT72]). Thus it suffices to assume an oracle to determine the relative size of the function values (rather than calculating the explicit function values) of given solutions.

4. Open problems. It is an interesting problem to improve the polynomial algorithm presented in this paper. It may be possible, for instance, to find $O(n)$ algorithms for the cases discussed in §§2.2 and 3.

The PTP with a fixed number of sources is an important subclass. It has applications in many practical problems that have a small number of sources (e.g., factories or warehouses) and a large number of sinks (e.g., retailers or consumers). As mentioned earlier, the algorithm of [TGMV93] establishes the strong polynomiality of this problem. The value of the algorithm seems, however, to be theoretical rather than practical. So it is desirable to find algorithms of improved complexities that are less dependent on the constant m.

Further research in a somewhat different direction is to develop efficient heuristics for the general case. No such approximations are known even if we make some additional assumptions on the concave production cost functions such as symmetry and separability.

REFERENCES

[ACHS89] N. Alon, S. Cosares, D. S. Hochbaum, and R. Shamir, *An algorithm for the detection and construction of Monge sequences*, Linear Algebra Appl., 114-115 (1989), pp. 669–680.

[BFPRT72] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan, *Time bounds for selection*, J. Computer System Sci., 7 (1972), pp. 448–461.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of* NP-*completeness*, W. H. Freeman, San Francisco, CA, 1979.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Algorithms and Combinatorics 2, Springer-Verlag, New York, Berlin, 1988.

[GP90] G. M. Guisewite and P. M. Pardalos, *Minimum concave cost network flow problems: Applications, complexity and algorithms*, Ann. Oper. Res., 25 (1990), pp. 75–100.

[Hof63] A. J. Hoffman, *On simple linear programming problems*, in Proc. 7th Symposium in Pure Mathematics of the American Mathematical Society, V. Klee, ed., 1963.

[TDG93] H. Tuy, N. D. Dan, and S. Ghannadan, *Strongly polynomial time algorithms for certain concave minimization problems on networks*, Oper. Res. Lett., 14 (1993), pp. 99–109.

[TGMV93a] H. Tuy, S. Ghannadan, A. Migdalas, and P. Värbrand, *Strongly polynomial algorithm for a production-transportation problem with a fixed number of nonlinear variables*, Department of Mathematics, Linköping University, preprint, 1993.

[TGMV93b] ———, *The minimum concave cost network flow problem with a fixed number of sources and nonlinear arc costs*, Department of Mathematics, Linköping University, preprint, 1993.

[Murty83] K. G. Murty, *Linear Programming*, Wiley, New York, 1983.

[Orlin93] J. B. Orlin, *A faster strongly polynomial minimum cost flow algorithm*, Oper. Res., 41 (1993), pp. 338–350.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 265–282, May 1996
© 1996 Society for Industrial and Applied Mathematics

A UNIFIED ANALYSIS OF HOFFMAN'S BOUND VIA FENCHEL DUALITY*

JAMES V. BURKE† AND PAUL TSENG†

Abstract. In 1952, Hoffman showed that the distance from any point to the solution set of a linear system is bounded above by a constant times the norm of the residual. Subsequently, this bound has been studied extensively and has found many practical uses. In this paper, we consider an extension of Hoffman's bound to a partially infinite-dimensional setting in which the norm defining the distance is replaced by a positively homogeneous convex function and the linear system is replaced by a convex inclusion of the form $Ax - a \in K$, where A is a continuous linear operator from some normed linear space into R^m, a is an element of the range space of A, and K is a nonempty closed convex cone in R^m. When specialized to the finite-dimensional case, we unify and extend many existing results on Hoffman's bound. Our analysis is based on the use of Fenchel duality to express the distance as the supremum of a certain concave function over a bounded subset of the polar of K. Much of our analysis also extends to the case where K is a nonempty closed convex set, not only a convex cone.

Key words. Fenchel duality, error bound, linear system, positively homogeneous convex function, convex cone, semidefinite programming

AMS subject classifications. 49M39, 90C25, 90C31, 90C34, 90C48, 49K40

1. Introduction.

In 1952, A. Hoffman [14] published a very interesting result showing that the distance, measured in some l_p-norm ($p \geq 1$), from any point x to the solution set of a linear system

$$A\xi \leq a,$$

where A is a real matrix and a is a real vector, is bounded above by some constant (depending on the matrix A only) times the norm of the residual

$$[Ax - a]_+,$$

where $[x]_+$ denotes the positive part of x. A local version of this bound was discovered slightly earlier by Rosenbloom [36]. Why is Hoffman's bound interesting? First, many iterative methods for solving linear systems have the property that they decrease the residual to zero. Hoffman's bound guarantees that the iterates generated by these methods are approaching the solution set. Second, Hoffman's bound is a key to the sensitivity analysis of linear/integer programs (see [9], [31], [33]), the computation of local error bounds (see [25], [31]), and the convergence analysis of descent methods for linearly constrained minimization (see [10], [11], [16], [25], [37]–[39]).

In the original work of Hoffman [14], an explicit formula for the constant was also given for the cases $p = 1, 2, \infty$. In [36], this was done for $p = 2$. Subsequently, much research has focused on sharpening this constant and on extending the bound to l_p spaces for other values of p. First Robinson [31], using LP duality, derived a new estimate of the constant for the case $p = 2$. Mangasarian [26] and Mangasarian and Shiau [29], also using LP duality, derived simpler estimates for the case $p = \infty$ (although their argument readily extends to the case $p = 1$). An estimate similar to that of [29] was also obtained by Cook et al. [9] in the context of integer programming

* Received by the editors March 26, 1993; accepted for publication (in revised form) November 1, 1994. This research was supported by National Science Foundation grants DMS-9303772 and CCR-9103804.

† Department of Mathematics, University of Washington, Seattle, WA 98195.

although, as was pointed out in [29, Remark 2.5], the latter estimate is weaker than that given in [29] (also see [18]). More recently, Hoffman's bound has been refined and extended to the case of arbitrary p by Li [22] and by Güler, Hoffman, and Rothblum [13], to the case of an infinite system of linear inequalities by Hu and Wang [15], and to the infinite-dimensional case under the l_∞-norm by Bergthaller and Singer [3]. In particular, a sharp estimate of the constant was given in [22] which, as was remarked in [22], can also be inferred from the analysis in [3]. A simplified proof of Hoffman's original bound was given by Güler [12] and, very recently, Klatte and Thiere [20] showed that the sharp estimate of the constant given in [22] is equal to the estimate given in [19] and in [31]. Finally, we note that Hoffman's bound has also been extended to convex programs and systems of convex inequalities [2], [7], [27], [32], to systems of analytic equalities/inequalities [24], and to the extremal solutions of linear programs [21]. However, such extensions are beyond the scope of the present paper and will not be treated any further.

In this paper, we consider the general setting in which A is a continuous linear operator from some real normed linear space X to another real normed linear space Y, a is an element of Y, and "$\leq$" denotes a partial ordering on Y induced by some nonempty closed convex cone K in Y. (Extensions of our results to the case where K is a convex set are discussed in §6.) Special attention will be given to the partially infinite case in which Y is finite-dimensional (see §§3–5). Here, we identify the normed space Y with the pair $(\mathbf{R}^m, \|\cdot\|)$, where $\|\cdot\|$ is a given norm on $\mathbf{R}^m$. The partially infinite case arises in a number of emerging applications, including constrained interpolation, spectral estimation, and semi-infinite linear programming (see [5] and references therein). Applications to the case in which the cone K is not polyhedral arise in the study of *positive semidefinite programming* [1], [6], [30].

Our two main contributions are (1) to unify and extend previous results on Hoffman's bound (for a linear system) to the partially infinite case and (2) to extend the theory of Hoffman bounds to closed convex sets and, in particular, the case of nonpolyhedral convex cones such as the cone of real positive semidefinite matrices. In particular, we derive general conditions under which the distance, measured in the norm on X, from any point x in X to the solution set of the convex cone inclusion

$$A\xi - a \in K$$

is bounded above by some constant (depending on A and K only) times the distance, measured in the norm on Y, from $Ax - a$ to K. We give an estimate of the constant in this bound which, in the case where X is an l_p (L_p) space for some $1 < p < \infty$, is as sharp as existing estimates. We also show that, in many cases, the norm used to measure the distance on X can be replaced by a gauge function [34, §15], i.e., a nonnegative positively homogeneous convex function. These results unify and extend existing results on Hoffman's bound (see [3], [9], [13], [22], [29]) which consider only the case in which K is the Cartesian product of closed intervals (see §5).

There are two key elements to our analysis. First, we use a version of Fenchel's duality theorem for convex programs with generalized constraints to express the distance from x to the solution set of $A\xi - a \in K$ as the supremum of a certain concave function over K°, the polar cone of K (see (1)). Second, we construct a cone $W \subset K^\circ$ that is large enough so that the preceding supremum is unchanged when K° is replaced by W (see (2)) and yet is small enough so that $W \cap (A^*)^{-1}\mathrm{B}^\circ$ is bounded, where B° denotes the dual ball in Y^*. Then, by observing that this supremum is unchanged when W is further intersected with $(A^*)^{-1}\mathrm{B}^\circ$, the desired bound readily

follows (see §2 for a detailed argument). We note that there are typically several possible choices for the cone W and that different choices of W yield bounds that may be very different in nature. Moreover, for some choices of W, the set $W \cap (A^*)^{-1}\mathrm{B}^\circ$ is intimately related to the extreme points of $K^\circ \cap (A^*)^{-1}\mathrm{B}^\circ$ (see Lemma 5).

Following [34] and [35], we adopt the following notation throughout. We denote by $\mathbf{R}^m$ the m-dimensional real space. All vectors in $\mathbf{R}^m$ are column vectors and the superscript ^{T} denotes transpose. For any subspace S of $\mathbf{R}^m$, we denote by $S^\perp$ the orthogonal complement of S. For any real normed linear space X, we denote by $\|\cdot\|$ the norm on X and by X^* the vector space of continuous linear functionals on X. For each $x \in X$ and $x^* \in X^*$, we denote by $\langle x^*, x\rangle$ the value of the function x^* at x (so the bilinear form $\langle\cdot,\cdot\rangle$ is a pairing of X^* and X). It is well known that X^* is a Banach space when endowed with the dual norm $\|x^*\|_* = \sup_{\|x\|\le 1} \langle x^*, x\rangle$ defined for all $x^* \in X^*$. We will use B to denote the unit ball on X (i.e., $\mathrm{B} = \{x \in X : \|x\| \le 1\}$) and use B° to denote the dual unit ball on X^* (i.e., $\mathrm{B}^\circ = \{x^* \in X^* : \|x^*\|_* \le 1\}$). For each nonempty closed convex set C in X, we let ψ_C denote the indicator function for C, i.e.,

$$\psi_C(x) := \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{else,} \end{cases}$$

and let

$$\operatorname{dist}(x \mid C) := \inf_{\xi \in C} \|x - \xi\|$$

for all $x \in X$. The recession cone of C, denoted $\operatorname{rec}(C)$, is the set of directions y such that $C + y \subset C$. When X is finite-dimensional, we denote by $\operatorname{ri} C$ the relative interior of C.

In this paper we simultaneously deal with two normed linear spaces X and Y and, for simplicity, we use $\|\cdot\|$ to denote the norm on either X or Y and use $\|\cdot\|_*$ to denote the dual norm on either X^* or Y^*. Analogously, we use $\langle\cdot,\cdot\rangle$ to denote the bilinear form on either $X \times X^*$ or $Y \times Y^*$, use B to denote the unit ball on either X or Y, and use B° to denote the dual unit ball on either X^* or Y^*, etc. For any convex function $f\colon X \mapsto \mathbf{R} \cup \{+\infty\}$, we denote by f^* the conjugate of f, i.e.,

$$f^*(x^*) = \sup_x \{\langle x^*, x\rangle - f(x)\}$$

for all $x^* \in X^*$. The effective domain of the function f, denoted $\operatorname{dom}(f)$, is the set of points $x \in X$ for which $f(x) < +\infty$. For any nonempty closed convex cone K in Y, we denote by $\operatorname{lin}(K) = K \cap (-K)$ the lineality of K, by $\operatorname{span}(K)$ the linear span of the elements of K, and by K° the polar of K, i.e.,

$$K^\circ = \{\, y^* \in Y^* : \langle y^*, y\rangle \le 0 \text{ for all } y \in K \,\}.$$

For any linear operator A from X to Y, we denote by $A^*\colon Y^* \mapsto X^*$ the adjoint of A. We also denote by $\operatorname{ran}(A)$ the range of A and by $\ker A^*$ the kernal of A^*.

2. Extensions of Hoffman's bound. Let X and Y be two real normed linear spaces. Let f be a real-valued convex function on X whose conjugate f^* is nonnegative everywhere, i.e., $f^*(x^*) \ge 0$ for all $x^* \in X^*$. (We can think of f as a generalization of the norm function on X.) Let K be a nonempty closed convex cone in Y and let A be a linear operator from X to Y. Our goal is to determine conditions under which

the "generalized distance" from a point $x \in X$ to the solution set of the convex cone inclusion $A\xi - a \in K$, given by

$$\inf_{A\xi - a \in K} f(x - \xi),$$

is bounded above by some constant times

$$\operatorname{dist}(Ax - a \mid K)$$

uniformly in x and in a. In the case where $X = \mathrm{R}^n$, $Y = \mathrm{R}^m$ for some m and n, f is the norm function on X, and K is the nonpositive orthant in Y, this bound reduces to Hoffman's original bound.

In our analysis, we use a version of Fenchel's duality theorem and certain properties of convex sets to express the above generalized distance as the supremum of the dual norm over a bounded subset of K°. More precisely, we first use a version of Fenchel's duality theorem to express the "generalized distance" from x to the solution set of $A\xi - a \in K$ as the supremum of a concave function over K°:

$$\inf_{A\xi - a \in K} f(x - \xi) = \sup_{y^* \in K^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^* y^*)\}\ . \tag{1}$$

Then we find a cone $W \subset K^\circ$ (independent of x) such that the above supremum is unchanged when K° is replaced by W, i.e.,

$$\sup_{y^* \in K^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^* y^*)\} = \sup_{y^* \in W} \{\langle y^*, Ax - a\rangle - f^*(A^* y^*)\}. \tag{2}$$

Finally, we use the nonnegativity of f^* and a basic property of the dual norm to bound the right-hand side of (2) by $\operatorname{dist}(Ax - a \mid K)$ times the constant

$$\sup_{y^* \in W \cap (A^*)^{-1} D} \|y^*\|_*\ ,$$

where D is the "generalized dual unit ball" given by $D = \{\ x^* \in X^* : f^*(x^*) \le 1\ \}$. Thus the two relations (1) and (2) are key to extending Hoffman's bound to our problem setting. To ensure that the bound is not trivial (i.e., the above constant is not equal to ∞), we will further choose W so that

$$W \cap (A^*)^{-1} D \text{ is bounded.}$$

(Thus, if $K^\circ \cap (A^*)^{-1} D$ is bounded, we can simply choose K° to be W.) We defer the discussion of when the strong duality relation (1) holds to §3 and the discussion of when the desired cone W exists to §§4 and 5.

We show below that if (1) and (2) hold, we immediately obtain a nontrivial extension of Hoffman's bound to our problem setting (also see [12, p. 3] for a related derivation for linear systems). We will refine this extended bound shortly (see Theorem 2).

THEOREM 1. *Let X and Y be real normed linear spaces. Let f be a real-valued convex function on X whose conjugate f^* is nonnegative everywhere, let K be a nonempty closed convex cone in Y, let A be a continuous linear operator from X to Y, let $a \in \operatorname{ran}(A) - K$, and let $x \in X$. Suppose that* (1) *holds and there is a cone $W \subset K^\circ$ satisfying* (2). *Then*

$$\inf_{A\xi - a \in K} f(x - \xi) \le \mu_1 \operatorname{dist}(Ax - a \mid K)\ , \tag{3}$$

where $\mu_1 \in \{-\infty\} \cup [0, \infty]$ *is given by*

$$\mu_1 := \sup_{\substack{y^* \in W \\ A^* y^* \in \operatorname{dom}(f^*)}} \|y^*\|_* \,. \tag{4}$$

(We use the convention that $\infty \cdot 0 = (-\infty) \cdot 0 = 0$ *and* $\mu_1 = -\infty$ *if the supremum in* (4) *is taken over an empty set.)*

Proof. Using (1) and (2), we obtain

$$\begin{aligned} \inf_{A\xi - a \in K} f(x - \xi) &= \sup_{y^* \in W} \{\langle y^*, Ax - a\rangle - f^*(A^* y^*)\} \\ &\leq \sup_{\substack{y^* \in W \\ A^* y^* \in \operatorname{dom}(f^*)}} \langle y^*, Ax - a\rangle \\ &\leq \sup_{\substack{y^* \in W \\ A^* y^* \in \operatorname{dom}(f^*)}} \|y^*\|_* \operatorname{dist}(Ax - a \mid K) \\ &= \mu_1 \operatorname{dist}(Ax - a \mid K), \end{aligned}$$

where the first inequality follows from the nonnegativity of f^* and the second inequality follows from the inequality

$$\langle y^*, y\rangle \leq \|y^*\|_* \operatorname{dist}(y \mid K) \qquad \forall y \in Y, y^* \in K^\circ, \tag{5}$$

which in turns follows from the observation (cf. the definition of the dual norm and the polar cone) that $\langle y^*, y\rangle \leq \langle y^*, y - y'\rangle \leq \|y^*\|_* \|y - y'\|$ for all $y \in Y$, $y^* \in K^\circ$, $y' \in K$. (In fact, it follows from the minimum norm duality theorem of functional analysis (see [23, §5.13]) that equality in (5) is attained for some $y^* \in K^\circ$ or, equivalently, that $\operatorname{dist}(y \mid K) = (\psi_{B^\circ \cap K^\circ})^*(y)$ (see [8, Thm. 3.1]).) □

In some literature, the bound (3) is stated with the right-hand quantity $\operatorname{dist}(Ax - a \mid K)$ replaced by $\operatorname{dist}(\bar{a} - a \mid K)$ or by $\|\bar{a} - a\|$, where $\bar{a}$ is any element of $Ax - K$. However, these modified bounds are equivalent to (3) in the sense that one holds if and only if the others hold. To see this, note that for any $\bar{a} \in Ax - K$ we have

$$\begin{aligned} \operatorname{dist}(Ax - a \mid K) &\leq \operatorname{dist}(Ax - \bar{a} \mid K) + \operatorname{dist}(\bar{a} - a \mid K) \\ &= \operatorname{dist}(\bar{a} - a \mid K) \leq \|\bar{a} - a\|, \end{aligned}$$

with equality holding throughout when $\bar{a}$ is an element of $Ax - K$ nearest to a in the norm on Y.

In many cases, we can choose the cone W so that in addition

$$A^* W \neq \{0\} \quad \text{whenever} \quad W \neq \{0\} \tag{6}$$

(see Lemma 5). This has the advantage that when f is positively homogeneous, the bound given in Theorem 1 can be further refined, as we show in the next two theorems.

First, we consider the case where f is real-valued positively homogeneous convex of degree 1 and positive except at the origin. We refine the bound given in Theorem 1 by applying the theorem to this case and then showing that the supremum in (4) can be taken over a smaller set. In what follows, we denote by f° the polar of f on X^* [34, p. 128], i.e.,

$$f^\circ(x^*) := \sup_{x \neq 0} \frac{\langle x^*, x\rangle}{f(x)}.$$

THEOREM 2. *Let X, Y, K, A, a, and x be as in Theorem* 1 *and let f be a real-valued positively homogeneous convex function on X of degree* 1 *that is positive except at the origin. Suppose that* (1) *holds and there is a cone $W \subset K^\circ$ satisfying* (2). *Then*

$$\inf_{A\xi - a \in K} f(x-\xi) \leq \mu_2 \operatorname{dist}(Ax - a \mid K), \tag{7}$$

where $\mu_2 \in \{-\infty\} \cup [0, \infty]$ is given by

$$\mu_2 := \sup_{\substack{y^* \in W \\ f^\circ(A^* y^*) \leq 1}} \|y^*\|_* \,. \tag{8}$$

(We use the convention that $\infty \cdot 0 = (-\infty) \cdot 0 = 0$ and $\mu_2 = -\infty$ if the supremum in (8) *is taken over an empty set.) If in addition W satisfies* (6), *then the inequality sign in* (8) *can be replaced by an equality sign.*

Proof. Since $f(0) = 0$ so that $f^*(x^*) \geq 0$ for all $x^* \in X^*$, we have, upon applying Theorem 1, that (3) holds with μ_1 given by (4). Since f is positively homogeneous of degree 1 so that $f^*(x^*) < \infty$ if and only if $f^\circ(x^*) \leq 1$, we see that $\mu_2 = \mu_1$ and hence (7) holds (with μ_2 given by (8)). Now, suppose that W also satisfies (6), and we will show that the inequality sign in (8) can be replaced by an equality sign.

First, consider the case where $W \neq \{0\}$ and $A^* y^* \neq 0$ for all nonzero $y^* \in W$. Then the supremum in (8) is unchanged when y^* is further restricted to satisfy $A^* y^* \neq 0$. Since W is a cone and, as is easily checked, f° is positively homogeneous of degree 1 and positive except at the origin, multiplying any such y^* by $1/f^\circ(A^* y^*)$ would yield a new y^* in W that satisfies $f^\circ(A^* y^*) = 1$ and whose dual norm is no less than before. This shows that the supremum in (8) is unchanged when the inequality sign is replaced by an equality sign.

Second, consider the case where $W \neq \{0\}$ and $A^* y^* = 0$ for some nonzero $y^* \in W$. Then the supremum in (8) is taken over an unbounded set, so the supremum equals ∞. Suppose that we replace the inequality sign in (8) by an equality sign. Then, since $A^* W \neq \{0\}$ by (6), the supremum in (8) would be taken over a nonempty set which, by the existence of y^*, is unbounded. Thus, the supremum in (8) would still equal ∞.

Third, consider the case where $W = \{0\}$. Then, since f is positive except at the origin so that $f^*(0) = 0$, we have that the right-hand side of (2) equals zero, and so (1) and (2) imply that the left-hand side of (7) equals zero. Since f is zero only at the origin, we have $Ax - a \in K$, implying $\operatorname{dist}(Ax - a \mid K) = 0$, so both sides of (7) equal zero (under the convention $(-\infty) \cdot 0 = 0$) regardless of whether the inequality sign in (8) is replaced by an equality sign. □

Next, we consider the case where f is real-valued positively homogeneous convex of degree $1 < p < \infty$ and positive except at the origin. We will refine the bound given in Theorem 1 by suitably modifying the proof of this theorem and of Theorem 2. This refined bound is closely related to the bound given by (7) when X is an ℓ_p space and will be used in §4 to unify existing results on Hoffman's bound.

THEOREM 3. *Let X, Y, K, A, a, and x be as in Theorem* 1 *and let f be a real-valued positively homogeneous convex function on X of degree $p \in (1, \infty)$ that is positive except at the origin. Suppose that* (1) *holds and there is a cone $W \subset K^\circ$ satisfying* (2) *and* (6). *Then*

$$\inf_{A\xi - a \in K} f(x-\xi)^{\frac{1}{p}} \leq \mu_3 \operatorname{dist}(Ax - a \mid K), \tag{9}$$

where $\mu_3 \in \{-\infty\} \cup [0, \infty]$ *is given by*

$$\mu_3 := \left(\frac{1}{p}\right)^{\frac{1}{p}} \left(\frac{1}{q}\right)^{\frac{1}{q}} \sup_{\substack{y^* \in W \\ f^*(A^*y^*)=1}} \|y^*\|_* \tag{10}$$

with $q \in (1, \infty)$ *satisfying* $1/p + 1/q = 1$. *(We use the convention that* $\infty \cdot 0 = (-\infty) \cdot 0 = 0$ *and* $\mu_3 = -\infty$ *if the supremum in* (10) *is taken over an empty set.)*

Proof. We divide the proof into three cases as in the proof of Theorem 2. First, suppose that $W \neq \{0\}$ and $A^*y^* \neq 0$ for all nonzero $y^* \in W$. Since f is real-valued and positively homogeneous convex of degree $p > 1$, we also have that $f^*(x^*) > 0$ for all $x^* \neq 0$. (To see this, note that $x^* \neq 0$ implies the existence of an $x \in X$ with $\langle x^*, x \rangle > 0$. Then $\langle x^*, \theta x \rangle - f(\theta x) = \theta \langle x^*, x \rangle - \theta^p f(x) > 0$ for all $\theta > 0$ sufficiently small.) Then, it follows that $f^*(A^*y^*) > 0$ for all nonzero $y^* \in W$. By combining (1) with (2) and then using the preceding observation, we obtain the following chain of equalities and inequalities:

$$\begin{aligned}
\inf_{A\xi - a \in K} f(x - \xi) &= \sup_{y^* \in W} \{\langle y^*, Ax - a \rangle - f^*(A^*y^*)\} \\
&\leq \sup_{\substack{y^* \in W \\ f^*(A^*y^*) < \infty}} \{\|y^*\|_* \operatorname{dist}(Ax - a \mid K) - f^*(A^*y^*)\} \\
&= \sup_{\substack{y' \in W \\ f^*(A^*y')=1}} \left\{ \sup_{\substack{\theta \geq 0 \\ y^* = \theta y'}} \{\|y^*\|_* \operatorname{dist}(Ax - a \mid K) - f^*(A^*y^*)\} \right\} \\
&= \sup_{\substack{y' \in W \\ f^*(A^*y')=1}} \left\{ \sup_{\theta \geq 0} \{ \theta \|y'\|_* \operatorname{dist}(Ax - a \mid K) - \theta^q \} \right\},
\end{aligned}$$

where the inequality follows from (5). The second equality follows from the observations that W is a cone which is not simply the origin and that $0 < f^*(A^*y^*)$ for all nonzero $y^* \in W$. The final equality follows from the fact that f^* is positively homogeneous of degree q. Now, for any positive number t, the supremum of the function $\theta \mapsto \theta t - \theta^q$ (also using $q > 1$) is attained at $\theta = (t/q)^{1/(q-1)}$ with a value of

$$\frac{q-1}{q} \left(\frac{1}{q}\right)^{1/(q-1)} t^{q/(q-1)} = \frac{1}{p} \left(\frac{1}{q}\right)^{p/q} t^p ,$$

where the equality follows from the fact that $p = q/(q-1)$. Using this in the above relation and then taking the pth root of both sides yields (9) with μ_3 given by (10).

Second, suppose that $W \neq \{0\}$ and $A^*y^* = 0$ for some nonzero $y^* \in W$. Then, since $A^*W \neq \{0\}$ by (6), the supremum in (10) is taken over a nonempty set which, by the existence of y^*, is unbounded. Thus, $\mu_3 = \infty$ and so (9) holds trivially.

Third, suppose that $W = \{0\}$. Then since f^* is positively homogeneous so that $f^*(0) = 0$, we have that the right-hand side of (2) equals zero and so (1) and (2) imply that the left-hand side of (9) equals zero. Since f is zero only at the origin, we have $Ax - a \in K$, implying that $\operatorname{dist}(Ax - a \mid K) = 0$ and hence both sides of (9) equal zero (under the convention $(-\infty) \cdot 0 = 0$). ∎

In the case where X is an ℓ_p (L_p) space with $1 < p < \infty$, we can apply (7) with μ_2 given by (8) and f taken to be the norm function on X, for which we have $f^\circ(x^*) = \|x^*\|_*$. We can also apply (9) with μ_3 given by (10) and f taken to be the norm function on X raised to the pth power, for which we have

$$f^*(x^*) = \left(\frac{1}{p}\right)^{\frac{q}{p}} \left(\frac{1}{q}\right) \|x^*\|_*^q .$$

Then note that $\mu_2 = \mu_3$, so the two bounds (7) and (9) are in fact equivalent in this case.

Although the bounds given in Theorems 1, 2, and 3 are attractive for their generality and simplicity, their utility is still in question since we have yet to identify conditions under which the strong duality relation (1) holds and to find a cone $W \subset K^\circ$ such that (2) and (6) hold with the corresponding constants μ_1, μ_2, μ_3 finite. We will address these and related issues in the remaining sections.

3. Checking for strong duality. In this section we assume that $Y = \mathbf{R}^m$ for some m and we identify a constraint qualification under which the strong duality relation (1) holds.

LEMMA 4. *Let f be a real-valued convex function on a real normed linear space X, let A be a continuous linear operator from X to $\mathbf{R}^m$ for some m, and let K be a nonempty closed convex cone in $\mathbf{R}^m$ of the form $K = K_1 \cap K_2$ for some closed convex polyhedral set K_1 and some closed convex set K_2. Then, for every $a \in \operatorname{ran}(A) - (K_1 \cap \operatorname{ri} K_2)$ and every $x \in X$, the strong duality relation (1) holds.*

Proof. Fix any $a \in \operatorname{ran}(A) - (K_1 \cap \operatorname{ri} K_2)$ and any $x \in X$. Upon applying a recent result of Borwein and Lewis [4, Cor. 4.6] to the primal problem

$$\inf_{A\xi \in K+a} \{f(x-\xi) + \psi_{K+a}(A\xi)\}$$

and using the observation that $(\psi_{K+a})^*$ is the pointwise sum of $(\psi_K)^* = \psi_{K^\circ}$ and the linear function $y^* \mapsto \langle y^*, a\rangle$, it readily follows that the strong duality relation (1) holds. □

The above duality result and its relatives are extremely powerful tools in convex analysis. Interested readers are referred to [4, 17, 34, 35] for further discussions of their applications.

4. Choosing the cone W: General case. In this section we assume that $Y = \mathbf{R}^m$ for some m, and we propose a choice for the cone $W \subset K^\circ$ and identify conditions under which (2) and (6) hold with the corresponding constants μ_2, μ_3 finite. Then, by applying the results of previous sections with this choice of W, we obtain the desired extension of Hoffman's bound to the cone inclusion case.

Let K be a nonempty closed convex cone in $\mathbf{R}^m$ and let A be a continuous linear operator from the normed linear space X to $\mathbf{R}^m$. Let $S \subset \mathbf{R}^m$ be the subspace

$$S := \ker A^* \cap \operatorname{lin}(K^\circ)\,, \tag{11}$$

define the cone

$$\hat{K} := K + S, \tag{12}$$

and consider the following subset W_1 of $\hat{K}^\circ$ given by

$$\begin{aligned} W_1 &:= \left\{ y^* \in \hat{K}^\circ : \begin{array}{c}\text{There does not exist nonzero } z^* \in \hat{K}^\circ \\ \text{with } A^*z^* = 0 \text{ and } y^* - z^* \in \hat{K}^\circ.\end{array} \right\} \\ &= \hat{K}^\circ \backslash [\hat{K}^\circ + [(\ker A^* \cap \hat{K}^\circ) \backslash \{0\}]], \end{aligned} \tag{13}$$

where we adopt the convention that the Minkowski sum of any set with the empty set is itself the empty set. Roughly speaking, W_1 comprises the minimal elements of $\hat{K}^\circ$ under the partial ordering: $y^* \prec y'$ if $y' \neq y^*$, $A^*y' = A^*y^*$, and $y' - y^* \in \hat{K}^\circ$. It is readily seen that W_1 is a cone (although in general nonconvex) and, in the case

where $\ker A^* \cap \hat{K}^\circ \neq \{0\}$, is contained in the boundary of $\hat{K}^\circ$. Also, notice that (see [34, Cor. 16.4.2])

$$\hat{K}^\circ = K^\circ \cap S^\perp,$$

and so $W_1 \subset \hat{K}^\circ \subset S^\perp$. The purpose of restricting W_1 to the subspace $S^\perp$ is to ensure that its intersection with $\ker A^*$ contains no line (since it can be seen that $S^\perp = \operatorname{ran}(A) + \operatorname{span}(K)$), which in turn is needed to ensure that the set over which the supremum in (8) or (10) (with W set to W_1) is taken is bounded.

We state and prove the first main result of this section, showing that the choice $W = W_1$ indeed satisfies (2) and (6), among other things.

LEMMA 5. *Let A be a continuous linear operator from the normed linear space X to $\mathbf{R}^m$. Let K be a nonempty closed convex cone in $\mathbf{R}^m$. Let S, $\hat{K}$, and W_1 be given by, respectively,* (11), (12), *and* (13). *Then the following hold.*

(a) *Let f be a real-valued convex function on X. For every $a \in \operatorname{ran}(A) - K$ and every $x \in X$,*

$$\sup_{y^* \in K^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\} = \sup_{y^* \in W_1} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\}. \tag{14}$$

(b) *Let D be any convex set in X^* for which $(A^*)^{-1}D$ is closed and has $\ker A^*$ as its recession cone. Then $W_1 \cap (A^*)^{-1}D$ is contained in the convex hull of the extreme points of the closed convex set*

$$G := \hat{K}^\circ \cap (A^*)^{-1}D.$$

(Thus, $W_1 \cap (A^*)^{-1}D$ is bounded whenever the extreme points of G are bounded, which in particular holds when G is polyhedral or when G is bounded.)

(c) *$W_1 = \{0\}$ if and only if $A^*W_1 = \{0\}$ if and only if $\operatorname{ran}(A) \subset K$.*

Proof. For every $x^* \in A^*\hat{K}^\circ$, let us define the nonempty closed convex set

$$C(x^*) := \{\, y^* \in \hat{K}^\circ : A^*y^* = x^* \,\}.$$

Since

$$\operatorname{lin}(\operatorname{rec}(C(x^*))) = \ker A^* \cap \operatorname{lin}(K^\circ) \cap S^\perp = S \cap S^\perp = \{0\},$$

where $\operatorname{rec}(C(x^*))$ denotes the recession cone of $C(x^*)$, we conclude that $C(x^*)$ contains at least one extreme point [34, Cor. 18.5.3].

We claim that for every $x^* \in A^*\hat{K}^\circ$, the extreme points of $C(x^*)$ are in W_1. To see this, fix any extreme point y of $C(x^*)$. If y were not in W_1, there would exist a nonzero $z \in \hat{K}^\circ$ such that $y - z \in \hat{K}^\circ$ and $A^*z = 0$. Then both $y + z$ and $y - z$ would be in $\hat{K}^\circ$ with $x^* = A^*(y - z) = A^*(y + z)$, implying that both $y + z$ and $y - z$ are in $C(x^*)$. But since $y = (1/2)(y + z) + (1/2)(y - z)$, this would show that y is not an extreme point of $C(x^*)$, which is a contradiction.

(a) Fix any $a \in \operatorname{ran}(A) - K$ and any $x \in X$. Let $\bar{x}$ be any element of X satisfying $A\bar{x} - a \in K$. (Such an $\bar{x}$ exists since $a \in \operatorname{ran}(A) - K$.)

First, we claim that

$$\sup_{y^* \in K^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\} = \sup_{y^* \in \hat{K}^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\}. \tag{15}$$

To see this, consider any $y^* \in K^\circ$. Since $S \subset \operatorname{lin}(K^\circ)$ so that $K^\circ = S + [K^\circ \cap S^\perp]$, we can write y^* as $y^* = y_1^* + y_2^*$ for some $y_1^* \in S$ and some $y_2^* \in K^\circ \cap S^\perp$. Consequently,

$$\begin{aligned}\langle y^*, Ax - a\rangle - f^*(A^*y^*) &= \langle y_2^*, Ax - a\rangle - f^*(A^*y_2^*) + \langle y_1^*, -a\rangle \\ &= \langle y_2^*, Ax - a\rangle - f^*(A^*y_2^*)\ ,\end{aligned}$$

where the second equality follows from $-a \in \operatorname{ran}(A) + K \subset \operatorname{ran}(A) + \operatorname{span}(K) = S^\perp$.

Second, we claim that for every $x^* \in A^*\hat{K}^\circ$,

$$\sup_{y^* \in C(x^*)} \langle y^*, Ax - a\rangle = \sup_{y^* \in C(x^*) \cap W_1} \langle y^*, Ax - a\rangle. \tag{16}$$

To see this, observe that for every $y^* \in C(x^*)$ we have

$$\begin{aligned}\langle y^*, Ax - a\rangle &= \langle y^*, A(x - \bar{x})\rangle + \langle y^*, A\bar{x} - a\rangle \\ &\le \langle y^*, A(x - \bar{x})\rangle \\ &= \langle x^*, x - \bar{x}\rangle,\end{aligned}$$

where the inequality follows from the fact that $A\bar{x} - a \in K$ and $y^* \in K^\circ$ (cf. $C(x^*) \subset \hat{K}^\circ \subset K^\circ$). Thus, the supremum on the left-hand side of (16) is finite. Since $y^* \mapsto \langle y^*, Ax - a\rangle$ is linear, this supremum is unchanged when we further restrict y^* to the extreme points of $C(x^*)$. By the claim shown earlier, all extreme points of $C(x^*)$ are in W_1.

Upon using (15) and (16) and the observation that

$$\hat{K}^\circ = \bigcup_{x^* \in A^*\hat{K}^\circ} C(x^*),$$

we obtain

$$\begin{aligned}\sup_{y^* \in K^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\} &= \sup_{y^* \in \hat{K}^\circ} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\} \\ &= \sup_{x^* \in A^*\hat{K}^\circ} \left\{ \sup_{y^* \in C(x^*)} \langle y^*, Ax - a\rangle - f^*(x^*) \right\} \\ &= \sup_{x^* \in A^*\hat{K}^\circ} \left\{ \sup_{y^* \in C(x^*) \cap W_1} \langle y^*, Ax - a\rangle - f^*(x^*) \right\} \\ &= \sup_{y^* \in \hat{K}^\circ \cap W_1} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\} \\ &= \sup_{y^* \in W_1} \{\langle y^*, Ax - a\rangle - f^*(A^*y^*)\},\end{aligned}$$

where the last equality follows trivially from the fact that $W_1 \subset \hat{K}^\circ$.

(b) If G is the empty set, then so is $W_1 \cap (A^*)^{-1}D$ (since $W_1 \subset \hat{K}^\circ$) and the claim follows trivially. Suppose that $G \neq \emptyset$. Since the lineality of G is the same as that of $C(0)$, which we have already shown to be $\{0\}$, it follows that G is a closed convex set containing no lines. Then, by a fundamental representation theorem for a closed convex set in $\mathbf{R}^m$ (see [34, Thm. 18.5]), any point y^* in G may be represented as

$$y^* = y' + z^*$$

for some y' in the convex hull of the extreme points of G and some z^* in the recession cone of G. Since the recession cone of G is precisely $\ker A^* \cap \hat{K}^\circ$, then, whenever y^* is not in the convex hull of the extreme points of G, the z^* in the previous representation of y^* must be nonzero and satisfy $z^* \in \ker A^* \cap \hat{K}^\circ$ and $y^* - z^* \in G \subset \hat{K}^\circ$. This in turn shows that $y^* \notin W_1$.

(c) Clearly if $A^*W_1 \neq \{0\}$, then $W_1 \neq \{0\}$. Conversely, if $A^*W_1 = \{0\}$, it is easily checked that $W_1 = \{0\}$. (Since for any nonzero $y^* \in \hat{K}^\circ$ with $A^*y^* = 0$, we have that $z^* := y^*$ is in $\hat{K}^\circ$ and satisfies $A^*z^* = 0$ and $y^* - z^* \in \hat{K}^\circ$, so $y^* \notin W_1$.)

Since $\hat{K} = K + [\ker A^* \cap \operatorname{lin}(K^\circ)]$ (see (11), (12)), we have

$$\begin{aligned} \operatorname{ran}(A) \not\subset K &\iff \operatorname{ran}(A) \not\subset \hat{K} \\ &\iff A^*\hat{K}^\circ \neq \{0\}. \end{aligned}$$

Hence, if $\operatorname{ran}(A) \not\subset K$, there is a nonzero x^* in $A^*\hat{K}^\circ$ so, by an earlier claim, $C(x^*)$ contains at least one extreme point, say y^*, which is in W_1. Since $A^*y^* = x^* \neq 0$, this shows that $A^*W_1 \neq \{0\}$. Conversely, suppose that $\operatorname{ran}(A) \subset K$ or, equivalently, $A^*\hat{K}^\circ = \{0\}$. Since $W_1 \subset \hat{K}^\circ$, it follows that $A^*W_1 = \{0\}$. □

Remarks.

1. We remark that the converse of Lemma 5(b) also holds in the sense that every extreme point of G is an element of $W_1 \cap (A^*)^{-1}D$. This follows from the observation that every extreme point y^* of G is an extreme point of $C(A^*y^*)$, which in turn, as was argued in the preceding proof, is an element of W_1. Thus, maximizing a concave function over $W_1 \cap (A^*)^{-1}D$ is equivalent to maximizing the same function over the extreme points of G.

2. Part (c) of Lemma 5 provides a simple check of when (6) with $W = W_1$ holds, namely, $\operatorname{ran}(A) \not\subset K$ or, equivalently (since $\operatorname{ran}(A)$ is a subspace), $\operatorname{ran}(A) \not\subset \operatorname{lin}(K)$. Thus whether (6) with $W = W_1$ holds depends on K through $\operatorname{lin}(K)$ only. And if (6) with $W = W_1$ does not hold or, equivalently, $\operatorname{ran}(A) \subset K$, then the inclusion $Ax - a \in K$ has a solution if and only if $-a \in K$, in which case the distance $\operatorname{dist}(Ax - a \mid K)$ equals zero for all $x \in X$.

Using Lemmas 4 and 5, we are now in a position to state conditions on A, K, a, and x under which the hypotheses of Theorems 2 and 3 with $W = W_1$ hold. We also give conditions on A and K under which the associated constants μ_2 and μ_3 are finite. We summarize the results in the next theorem.

THEOREM 6. *Let X, A, K, $\mathbf{R}^m$ be as in Lemma 5 and let W be the cone W_1 given by* (11), (12), and (13). *Assume that $K = K_1 \cap K_2$ for some polyhedral cone K_1 and some closed convex cone K_2 in $\mathbf{R}^m$. Then the following hold.*

(a) *Let f be a real-valued positively homogeneous convex function on X of degree 1 that is positive except at the origin. For every $a \in \operatorname{ran}(A) - (K_1 \cap \operatorname{ri} K_2)$ and every $x \in X$, the bound* (7) *with μ_2 given by* (8) *holds. Moreover, if $\operatorname{ran}(A) \not\subset K$, then the inequality sign in* (8) *can be replaced by an equality sign, and if either K is polyhedral or $\operatorname{ran}(A) + K = \mathbf{R}^m$, then μ_2 is finite.*

(b) *Let f be a real-valued positively homogeneous convex function on X of degree $p \in (1, \infty)$ that is positive except at the origin. For every $a \in \operatorname{ran}(A) - (K_1 \cap \operatorname{ri} K_2)$ and every $x \in X$, the bound* (9) *with μ_3 given by* (10) *holds. Moreover, if $\operatorname{ran}(A) \not\subset K$ and either K is polyhedral or $\operatorname{ran}(A) + K = \mathbf{R}^m$, then μ_3 is finite.*

Proof. We prove part (b) only. Part (a) can be similarly proved by replacing f^*, μ_3 and Theorem 3 in the argument below with, respectively, f°, μ_2 and Theorem 2. Fix any $a \in \operatorname{ran}(A) - (K_1 \cap \operatorname{ri} K_2)$ and any $x \in X$. If $\operatorname{ran}(A) \subset K$, then $-a \in K$ and it follows that $Ax - a \in K$ for all $x \in X$, and so the bound (9) holds trivially

(since both sides of (9) equal zero). If $\operatorname{ran}(A) \not\subset K$, then parts (a) and (c) of Lemma 5 yield that (2) and (6) with $W = W_1$ hold. Since (1) holds by Lemma 4, it follows from Theorem 3 that the bound (9) also holds.

Suppose that $\operatorname{ran}(A) \not\subset K$ and either K is polyhedral or $\operatorname{ran}(A) + K = \mathbf{R}^m$. We will show that μ_3 is finite. Since $\operatorname{ran}(A) \not\subset K$, Lemma 5(c) yields that (6) with $W = W_1$ holds and so either μ_3 is finite or $\mu_3 = \infty$. By the choice of f, there exists a scalar $\alpha > 0$ such that

$$\{x^* \in X^* : f^*(x^*) \leq 1\} \subset \alpha \mathrm{B}^\circ. \tag{17}$$

Choose any $x^1, \ldots, x^k$ in X such that $Ax^1, \ldots, Ax^k$ span $\operatorname{ran}(A)$ and normalized so that $\|Ax^1\| = \cdots = \|Ax^k\| = 1$. Let

$$D := \{x^* \in X^* : -\alpha \leq \langle x^*, x^i \rangle \leq \alpha,\ i = 1, \ldots, k\}.$$

Then D contains $\alpha \mathrm{B}^\circ$ and the set

$$(A^*)^{-1}D = \{y^* \in \mathbf{R}^m : -\alpha \leq \langle y^*, Ax^i \rangle \leq \alpha,\ i = 1, \ldots, k\}$$

is polyhedral with recession cone $\ker A^*$. Let G be as defined in Lemma 5(b). If K is polyhedral, then G is also polyhedral. If $\operatorname{ran}(A) + K = \mathbf{R}^m$ so that $S^\perp = \mathbf{R}^m$, then

$$G = K^\circ \cap (A^*)^{-1}D,$$

which must be bounded since its recession cone is $\ker A^* \cap K^\circ = (\operatorname{ran}(A) + K)^\circ = \{0\}$. Thus, in either case the extreme points of G form a bounded set so, by Lemma 5(b), $W_1 \cap (A^*)^{-1}D$ is bounded. Since D contains $\alpha \mathrm{B}^\circ$, it follows from (17) that $\mu_3 \neq \infty$. □

A few remarks about the condition $\operatorname{ran}(A) + K = \mathbf{R}^m$ are in order. First, this condition implies $S^\perp = \mathbf{R}^m$ and hence $\hat{K} = K + S = K$. Second, this condition is equivalent to the solvability of the inclusion $Ax - a \in K$ for all choices of the vector a. Third, although this condition (and assuming $\operatorname{ran}(A) \not\subset K$) suffices to guarantee the finiteness of μ_2 and of μ_3, it is far from necessary since K being polyhedral also suffices.

5. Choosing the cone W: Cartesian product case. As we had remarked in §1, most previous extensions of Hoffman's bound consider only the case in which $Y = \mathbf{R}^m$ for some m and K is the Cartesian product of closed intervals. Moreover, the constants are typically expressed as the supremum of the dual norm taken over all y^* in $K^\circ \cap (A^*)^{-1}\mathrm{B}^\circ$ such that the the "rows" of A corresponding to the nonzero components of y^* are linearly independent. In this section we consider this Cartesian product case and show that these extensions can alternatively be deduced from Theorem 6, so that Theorem 6 unifies and further extends previous extensions of Hoffman's bound. To facilitate the comparison, we need the following key lemma showing that the aforementioned supremum is unchanged if it is instead taken over all y^* in $W_1 \cap (A^*)^{-1}\mathrm{B}^\circ$.

LEMMA 7. *Let X, A, $\mathbf{R}^m$ be as in Lemma 5. Let $A_1, \ldots, A_m$ be the unique elements of X^* satisfying*

$$Ax = (\langle A_1, x \rangle, \ldots, \langle A_m, x \rangle)^T \quad \forall x \in X. \tag{18}$$

Let K be a nonempty closed convex cone in $\mathbf{R}^m$ of the form

$$K = I_1 \times \cdots \times I_m, \tag{19}$$

where $I_1, \ldots, I_m$ are closed intervals, each of which is either $(-\infty, 0]$ or $\{0\}$ or $[0, \infty)$. Let $\mathcal{I} = \{ i \in \{1, \ldots, m\} : I_i = \{0\} \}$ and assume that A_i, $i \in \mathcal{I}$, are linearly independent. Let S, $\hat{K}$, and W_1 be given by, respectively, (11), (12), *and* (13), *and let W_2 be the cone given by*

$$W_2 := \left\{ y^* \in K^\circ : \begin{array}{c} \textit{The elements } A_i, \textit{ with either } i \in \mathcal{I} \textit{ or the } i\textit{th} \\ \textit{component of } y^* \textit{ nonzero, are linearly independent} \end{array} \right\}.$$

Then, for any convex set D in X^* such that the right-hand supremum below is finite, we have

$$\sup_{\substack{y^* \in W_2 \\ A^* y^* \in D}} \|y^*\|_* = \sup_{\substack{y^* \in W_1 \\ A^* y^* \in D}} \|y^*\|_* . \tag{20}$$

Proof. Since A_i, $i \in \mathcal{I}$, are linearly independent, it is readily seen that $S^\perp$, which equals $\operatorname{ran}(A) + \operatorname{span}(K)$, is all of $\mathbf{R}^m$. Thus, $\hat{K}^\circ = K^\circ$. Consider any $y^* \in W_1$ with $A^* y^* \in D$. Suppose that y^* is not in W_2, so there exists a nonzero vector $z^* \in \mathbf{R}^m$ with $A^* z^* = 0$ and whose support (i.e., $\{i \in \{1, \ldots, m\} : z_i^* \neq 0\}$) is contained in the union of $\mathcal{I}$ with the support of y^*. Let L be the line segment

$$L := \{y^* + \theta z^* \ : \ \theta \in \mathbf{R}\} \cap K^\circ.$$

Then, for any $y' \in L$, the support of y' is contained in the union of $\mathcal{I}$ with the support of y^*, so y' is in W_1 (since y^* is in W_1 and $\hat{K}^\circ = K^\circ$ is the Cartesian product of closed intervals); moreover, y' satisfies $A^* y' = A^* y^* \in D$. Hence we conclude that L is a subset of $W_1 \cap (A^*)^{-1} D$ which, by the assumption that the right-hand supremum in (20) is finite, is bounded. Then the convex function $y' \mapsto \|y'\|_*$ attains its maximum over L at an endpoint of L. Let y' be such an endpoint. Then $y' \in W_1$, $A^* y' \in D$, $\|y'\|_* \geq \|y^*\|_*$, and, since y' is an endpoint of L, the union of $\mathcal{I}$ with the support of y' is strictly contained in the union of $\mathcal{I}$ with the support of y^*. If y' is not in W_2, then we repeat the above reduction procedure with y' in place of y^*, etc. After at most m repetitions, we would obtain a y^* satisfying in addition $y^* \in W_2$. Since the choice of y^* was arbitrary, the equality (20) must hold. □

Remarks.

1. The cone W_2 has been employed in the literature to replace the cone K° in the reduction (2). However, such a reduction is applicable only in the case where K is the Cartesian product of closed intervals (so that W_2 is defined).

2. When A is expressed in the form (18), its adjoint can be expressed analogously in the form

$$A^* y^* = y_1^* A_1 + \cdots + y_m^* A_m \quad \forall y^* = (y_1^*, \ldots, y_m^*)^T \in \mathbf{R}^m.$$

In the case where $X = \mathbf{R}^n$ for some n and A is represented as a matrix, the A_i's are simply the columns of A^T.

3. Under the preceding assumption that A_i, with $i \in \mathcal{I}$, are linearly independent, we in fact have

$$W_2 \subset W_1.$$

To see this, recall that $S^\perp = \mathbf{R}^m$ in this case, so if there were some $y^* \in W_2$ not in W_1, there would exist a nonzero $z^* \in K^\circ$ with $y^* - z^* \in K^\circ$ and $A^* z^* = 0$. Then the

first two relations would imply that the union of $\mathcal{I}$ with the support of z^* is contained in the union of $\mathcal{I}$ with the support of y^*, which together with $A^*z^* = 0$ would show that A_i, with either $i \in \mathcal{I}$ or the ith component of y^* nonzero, are linearly dependent, thus contradicting the hypothesis that y^* is in W_2.

By applying Theorem 6(a) with $f(\cdot) = \|\cdot\|$ and Lemma 7, we have the following extension of Hoffman's bound, which can be readily compared with previous extensions of this bound.

THEOREM 8. *Let X, A, K, R^m be as in Lemma 7. Assume that A and K have the form (18) and (19), respectively, and $\{A_i : I_i = \{0\}\}$ are linearly independent. Let W_2 be as in Lemma 7. Then, for every $a \in \mathrm{ran}\,(A) - (K_1 \cap \mathrm{ri}\,K_2)$ and every $x \in X$, we have*

$$\inf_{A\xi - a \in K} \|x - \xi\| \leq \mu_4 \mathrm{dist}\,(Ax - a \mid K), \tag{21}$$

where $\mu_4 \in \{-\infty\} \cup [0, \infty]$ is given by

$$\mu_4 := \sup_{\substack{y^* \in W_2 \\ \|A^*y^*\|_* = 1}} \|y^*\|_* . \tag{22}$$

Moreover, if $\mathrm{ran}\,(A) \not\subset K$, then μ_4 is finite.

We note that in the case where A and K have the form (18) and (19), respectively, the assumption that $\{A_i : I_i = \{0\}\}$ be linearly independent can be made without loss of generality. This follows from the observation that an $a = (a_1, \ldots, a_m)^T$ and an $x \in X$ satisfy $Ax - a \in K$ if and only if they satisfy $\langle A_i, x\rangle - a_i \in I_i$ for all $i \notin \mathcal{I} \backslash \mathcal{I}'$, where $\mathcal{I} = \{i \in \{1, \ldots, m\} : I_i = \{0\}\}$ and $\mathcal{I}'$ is any subset of $\mathcal{I}$ such that $\{A_i : i \in \mathcal{I}'\}$ are linearly independent and span the space spanned by $\{A_i : i \in \mathcal{I}\}$.

Below we relate Theorem 8 to existing extensions of Hoffman's bound. First, consider the special case in which

$$X = \mathrm{R}^n, \quad K = (-\infty, 0]^m ,$$

and the norms on X and R^m are, respectively, l_p- and l_r-norms for some $p \in [1, \infty]$ and some $r \in [1, \infty]$. Then the set G defined in Lemma 5(b), with D taken to be the l_∞-ball in R^n, is precisely the set $\sigma(A)$ defined by Güler, Hoffman, and Rothblum [13], and it follows from Lemma 5(b) and the remark following Lemma 5 that μ_4 given by (22) is equal to the constant $K(A)$ given in [13]. Hence the bound (21) is equivalent to that given in [13] (see Theorem 2.1 and Corollary 2.3 therein) for $p = r = \infty$ and is at least as sharp as the latter for all other values of p and r. Second, consider the special case in which

$$X = \mathrm{R}^n, \quad A = \begin{pmatrix} B \\ C \end{pmatrix}, \quad K = (-\infty, 0]^k \times \{0\}^l$$

for some $k \times n$ matrix B and some $l \times n$ matrix C of full row rank (with $k + l = m$). Since K has the form (19) and C has full row rank, we can replace W_1 in the definition of μ_4 (cf. (22)) with W_2 and (21) would still hold. Then it is readily seen that μ_4 is equal to the constant $\alpha_{\mu,\nu}(A, C)$ given by Li [22] for every choice of the norm on X. Hence (21) is equivalent to the bound given in [22] (see Theorem 2.4 therein). (The bound in [22] is stated with $\mathrm{dist}\,(Ax - a \mid K)$ replaced by $\|\bar{a} - a\|$, where $\bar{a}$ is any element of $Ax - K$. However, as we noted in §2, this change does not affect the sharpness of the bound.) Also, when the norm on X is the l_∞-norm and the norm

on $\mathbf{R}^m$ is the l_r-norm for some $r \in [1, \infty]$, it may be seen that (21) is equivalent to the bound given by Mangasarian and Shiau [29, Thm. 2.2]. If the norm on $\mathbf{R}^m$ is monotone, further simplification of the right-hand side in the bounds is possible (see [26] and [29]). In general, the constant in Hoffman's bound (and its extensions) is quite difficult to compute. Studies of the computational issues are given in [19], [28].

6. Extension to convex set inclusion. For simplicity, we have restricted the preceding discussion to the case of a convex cone inclusion. However, our approach readily extends to the more general case of a convex set inclusion of the form

$$B\xi - b \in C, \tag{23}$$

where B is a continuous linear operator from some real normed linear space X to another real normed linear space Y, b is an element of Y, and C is some nonempty closed convex set in Y. The idea of the extension is to apply a standard lifting trick to recast the set inclusion (23) as a cone inclusion of the kind studied in previous sections, and then to adapt the results of previous sections to the cone inclusion. We illustrate this idea below.

The following two convex cones associated with the set C will play significant roles in our analysis: the cone generated by $\{-1\} \times C$ given by

$$\operatorname{cone}(\{-1\} \times C) := \cup_{\lambda \geq 0}(\{-1\} \times C) \cdot \lambda$$

and the recession cone of C denoted by $\operatorname{rec}(C)$. A key property of $\operatorname{rec}(C)$ is that $\operatorname{rec}(C)^\circ$ is the closure of $\operatorname{dom}(\psi_C^*)$ (called the *barrier cone* of C) [34, Cor. 14.2.1].

Observe that the set inclusion (23) has the same solution set as the following cone inclusion:

$$A\xi - a \in K,$$

where we let

$$A = \begin{bmatrix} 0 \\ B \end{bmatrix}, \qquad a = \begin{bmatrix} 1 \\ b \end{bmatrix}, \qquad K = \operatorname{cl} \operatorname{cone}(\{-1\} \times C), \tag{24}$$

where cl denotes the closure. Let the norm on $\mathbf{R} \times Y$ be defined by $\|(\mu, y)\| = |\mu| + \|y\|$. Now, we can apply Theorem 6 directly to the cone inclusion, but the resulting bound turns out to be far from sharp. Instead, we will suitably modify the proof of Theorem 6 to take into account the structures of A, a, and K and thus obtain a much sharper bound.

THEOREM 9. *Let X, B, C be as given above with $Y = \mathbf{R}^m$ for some m. Assume that $C = C_1 \cap C_2$ for some closed convex polyhedral set C_1 and some closed convex set C_2 in $\mathbf{R}^m$. Then for every $b \in \operatorname{ran}(B) - (C_1 \cap \operatorname{ri} C_2)$ and every $x \in X$, we have*

$$\inf_{B\xi - b \in C} \|x - \xi\| \leq \mu_5 \operatorname{dist}(Bx - b \mid C), \tag{25}$$

where $\mu_5 \in \{-\infty\} \cup [0, \infty]$ is given by

$$\mu_5 := \sup_{\substack{(\alpha, y^*) \in W_1 \\ \|B^* y^*\|_* \leq 1}} \|y^*\|_* \tag{26}$$

and W_1 is given by (11), (12), *and* (13) *with A and K defined as in* (24). *(We use the convention that $\infty \cdot 0 = (-\infty) \cdot 0 = 0$ and $\mu_5 = -\infty$ if the supremum in* (26) *is taken*

over an empty set.) Moreover, if $\operatorname{ran}(B) \not\subset \operatorname{rec}(C)$, *then the inequality sign in* (26) *can be replaced by an equality sign, and if either* C *is polyhedral or* $\operatorname{ran}(B)+\operatorname{rec}(C) = \mathbf{R}^m$, *then* μ_5 *is finite.*

Proof. Fix any $b \in \operatorname{ran}(B) - (C_1 \cap \operatorname{ri} C_2)$ and any $x \in X$. Let a be given by (24) and let $K_1 = \operatorname{cl}\operatorname{cone}(\{-1\} \times C_1)$, $K_2 = \operatorname{cl}\operatorname{cone}(\{-1\} \times C_2)$. It is straightforward to verify that $a \in \operatorname{ran}(A) - (K_1 \cap \operatorname{ri} K_2)$ and hence, by Lemmas 4 and 5(a) (with $f(\cdot) = \|\cdot\|$), we have

$$\begin{aligned}
\inf_{B\xi - b \in C} \|x - \xi\| &= \inf_{A\xi - a \in K} \|x - \xi\| \\
&= \sup_{z^* \in W_1} \{\langle z^*, Ax - a\rangle - \psi_{\mathrm{B}^\circ}(A^* z^*)\} \\
&= \sup_{(\alpha, y^*) \in W_1} \left\{ \left\langle \begin{pmatrix} \alpha \\ y^* \end{pmatrix}, \begin{pmatrix} -1 \\ Bx - b \end{pmatrix} \right\rangle - \psi_{\mathrm{B}^\circ}(B^* y^*) \right\} \\
&= \sup_{\substack{(\alpha, y^*) \in W_1 \\ \|B^* y^*\|_* \le 1}} \{\langle y^*, Bx - b\rangle - \alpha\} \\
&\le \sup_{\substack{(\alpha, y^*) \in W_1 \\ \|B^* y^*\|_* \le 1}} \{\langle y^*, Bx - b\rangle - \psi_C^*(y^*)\} \\
&\le \sup_{\substack{(\alpha, y^*) \in W_1 \\ \|B^* y^*\|_* \le 1}} \|y^*\|_* \operatorname{dist}(Bx - b \mid C) ,
\end{aligned}$$

where the first inequality follows from the fact that $W_1 \subset K^\circ = \operatorname{epi}(\psi_C^*)$ so that $(\alpha, y^*) \in W_1$ implies $\alpha \ge \psi_C^*(y^*)$. The last inequality follows from the observation that

$$\begin{aligned}
\psi_C^*(y^*) &= \sup_{y \in C} \langle y^*, y\rangle \\
&= \langle y^*, Bx - b\rangle - \inf_{y \in C} \langle y^*, Bx - b - y\rangle \\
&\ge \langle y^*, Bx - b\rangle - \|y^*\|_* \operatorname{dist}(Bx - b \mid C)
\end{aligned}$$

for all $y^* \in \mathbf{R}^m$.

If $\operatorname{ran}(B) \not\subset \operatorname{rec}(C)$, then $\ker B^* \not\supset \operatorname{rec}(C)^\circ$ and there would exist a $y^* \in \operatorname{rec}(C)^\circ$ such that $B^* y^* \ne 0$. Then, by an argument analogous to the proof of Theorem 2, we obtain that the inequality sign in (26) can be replaced by an equality sign.

If C is polyhedral, then so is K and the finiteness of μ_5 follows from an argument analogous to the proof of Theorem 6. If $\operatorname{ran}(B) + \operatorname{rec}(C) = \mathbf{R}^m$, then $(B^*)^{-1}\mathrm{B}^\circ \cap \operatorname{rec}(C)^\circ$ would be bounded since its recession cone is $\ker B^* \cap \operatorname{rec}(C)^\circ = (\operatorname{ran}(B) + \operatorname{rec}(C))^\circ = \{0\}$. Since $W_1 \subset K^\circ = \operatorname{epi}(\psi_C^*)$ so that $(\alpha, y^*) \in W_1$ implies $y^* \in \operatorname{dom}(\psi_C^*)$, we have that

$$\mu_5 = \sup_{\substack{(\alpha, y^*) \in W_1 \\ \|B^* y^*\|_* \le 1}} \|y^*\|_* \le \sup_{\substack{y^* \in \operatorname{dom}(\psi_C^*) \\ \|B^* y^*\|_* \le 1}} \|y^*\|_* = \sup_{\substack{y^* \in \operatorname{rec}(C)^\circ \\ y^* \in (B^*)^{-1}\mathrm{B}^\circ}} \|y^*\|_* < \infty, \tag{27}$$

where the last equality follows from the fact that $\operatorname{cl}\operatorname{dom}(\psi_C^*) = \operatorname{rec}(C)^\circ$ [34, Cor. 14.2.1]. □

Remarks.

1. Note that if $\operatorname{ran}(B) \subset \operatorname{rec}(C)$, then the inclusion $B\xi - b \in C$ has a solution if and only if $-b \in C$, in which case every $\xi \in X$ is a solution.

2. Under the condition $\operatorname{ran}(B) + \operatorname{rec}(C) = \mathbf{R}^m$, we have

$$\operatorname{ran}(A) + \operatorname{span}(K) = \mathbf{R}^{m+1},$$

in which case S given by (11) reduces to the origin and μ_5 can be more simply written as

$$\mu_5 = \sup_{\substack{y^* \in W_3 \\ \|B^* y^*\|_* \leq 1}} \|y^*\|_* ,$$

where

$$W_3 := \left\{ y^* \in \operatorname{rec}(C)^\circ : \begin{array}{c} \text{There does not exist a nonzero } z^* \in \operatorname{rec}(C)^\circ \\ \text{with } B^* z^* = 0 \text{ and } y^* - z^* \in \operatorname{rec}(C)^\circ \end{array} \right\} .$$

Also, under this condition, the right-hand side of (27) is finite and hence can be used to estimate μ_5. In general, the bound (25) still holds if we replace W_1 in (26) by any subset of $\mathrm{R} \times \operatorname{rec}(C)^\circ$ that contains W_1.

3. In the special case where C, C_1, and C_2 are all cones, Theorem 9 reduces to Theorem 6(a) with $f(\cdot) = \|\cdot\|$.

REFERENCES

[1] F. Alizadeh, *Combinatorial Optimization with Interior Point Methods and Semidefinite Matrices*, Ph.D. thesis, Department of Computer Science, University of Minnesota, Minneapolis, MN, 1991.

[2] A. A. Auslender and J.-P. Crouzeix, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 1–11.

[3] C. Bergthaller and I. Singer, *The distance to a polyhedron*, Linear Algebra Appl., 169 (1992), pp. 111–129.

[4] J. M. Borwein and A. S. Lewis, *Partially finite convex programming, Part* I: *Quasi relative interiors and duality theory*, Math. Programming, 57 (1992), pp. 15–48.

[5] ———, *Partially finite convex programming, Part* II: *Explicit lattice models*, Math. Programming, 57 (1992), pp. 15–48.

[6] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[7] L. H. Brady, *Condition Constants for Solutions of Convex Inequalities*, Ph.D. thesis, Department of Computer Sciences, University of Wisconsin, Madison, WI, 1988.

[8] J. V. Burke and S.-P. Han, *A Gauss–Newton approach to solving generalized inequalities*, Math. Oper. Res., 11 (1986), pp. 632–643.

[9] W. Cook, A. M. H. Gerards, A. Schrijver, and É. Tardos, *Sensitivity results in integer linear programming*, Math. Programming, 34 (1986), pp. 251–264.

[10] J.-L. Goffin, *The relaxation method for solving systems of linear inequalities*, Math. Oper. Res., 5 (1980), pp. 388–414.

[11] O. Güler, *Augmented Lagrangian algorithms for linear programming*, J. Optim. Theory Appl., 75 (1992), pp. 445–470.

[12] ———, *Distance to a convex polyhedron and Hoffman's lemma*, Department of Management Sciences, University of Iowa, Iowa City, IA, July 1991, manuscript.

[13] O. Güler, A. J. Hoffman, and U. R. Rothblum, *Approximations to solutions to systems of linear inequalities*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 688–696.

[14] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Natl. Bur. Standards, 49 (1952), pp. 263–265.

[15] H. Hu and Q. Wang, *On approximate solutions of infinite systems of linear inequalities*, Linear Algebra Appl., 114/115 (1989), pp. 429–438.

[16] A. N. Iusem and A. R. De Pierro, *On the convergence properties of Hildreth's quadratic programming algorithm*, Math. Programming, 47 (1990), pp. 37–51.

[17] V. Jeyakumar, *Duality and infinite-dimensional optimization*, Nonlinear Anal., 15 (1990), pp. 1111–1122.

[18] D. Klatte, *Eine Bemerkung zur parametrischen quadratischen Optimierung*, in Seminarbericht Nr. 50, Sektion Mathematik, Humboldt-Universität zu Berlin, Berlin, 1983, pp. 174–185.

[19] D. KLATTE AND G. THIERE, *Error bounds for solutions of linear equations and inequalities*, Z. Oper. Res., 41 (1995), pp. 191–214.
[20] ———, *A note on Lipschitz constants for solutions of linear inequalities and equations*, Institut für Operations Research, Universität Zürich, to appear in Linear Algebra Appl., 1996.
[21] W. LI, *Sharp Lipschitz constants for basic optimal solutions and basic feasible solutions of linear programs*, SIAM J. Control Optim., 32 (1994), pp. 140–153.
[22] ———, *The sharp Lipschitz constants for feasible and optimal solutions of a perturbed linear program*, Linear Algebra Appl., 187 (1993), pp. 15–40.
[23] D. G. LUENBERGER, *Optimization by Vector Space Methods*, John Wiley and Sons, New York, 1969.
[24] Z.-Q. LUO AND J.-S. PANG, *Error bounds for analytic systems and their applications*, Math. Programming, 67 (1994), pp. 1–28.
[25] Z.-Q. LUO AND P. TSENG, *On the linear convergence of descent methods for convex essentially smooth minimization*, SIAM J. Control Optim., 30 (1992), pp. 408–425.
[26] O. L. MANGASARIAN, *A condition number for linear inequalities and linear systems*, in Methods of Operations Research 43, Proc. 6th Symposium über Operations Research, Universität Augsburg, September 7–9, 1981, G. Bamberg and O. Opitz, eds., Verlagsgruppe Athenäum/Hain/Scriptor/Hanstein, Konigstein, 1981, pp. 3–15.
[27] ———, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.
[28] O. L. MANGASARIAN AND T.-H. SHIAU, *A variable-complexity norm maximization problem*, SIAM J. Alg. Discrete Methods, 7 (1986), pp. 455–461.
[29] ———, *Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems*, SIAM J. Control Optim., 25 (1987), pp. 583–595.
[30] Y. NESTEROV AND A. NEMIROVSKII, *Interior-Point Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.
[31] S. M. ROBINSON, *Bounds for error in the solution set of a perturbed linear program*, Linear Algebra Appl., 6 (1973), pp. 69–81.
[32] ———, *An application of error bounds for convex programming in a linear space*, SIAM J. Control, 13 (1975), pp. 271–273.
[33] ———, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.
[34] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[35] ———, *Conjugate Duality and Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1974.
[36] P. C. ROSENBLOOM, *Quelques classes de problèmes extrémaux*, Bull. Soc. Math. France, 79 (1951), pp. 1–58.
[37] R. A. TAPIA, Y. ZHANG, AND Y. YE, *On the convergence of the iteration sequence in primal-dual interior point methods*, Math. Programming, 68 (1995), pp. 141–154.
[38] P. TSENG AND D. P. BERTSEKAS, *On the convergence of the exponential multiplier method for convex programming*, Math. Programming, 60 (1993), pp. 1–19.
[39] P. TSENG AND Z.-Q. LUO, *On the convergence of the affine scaling algorithm*, Math. Programming, 56 (1992), pp. 301–319.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 283–298, May 1996
© 1996 Society for Industrial and Applied Mathematics
002

A LINESEARCH PROCEDURE IN BARRIER METHODS FOR SOME CONVEX PROGRAMMING PROBLEMS*

A. MELMAN†

Abstract. A method is proposed for the linesearch procedure in barrier function methods for convex quadratically constrained quadratic programming problems, which includes linear and quadratic programming as special cases. The same linesearch problem appears in barrier methods for minimization over the cone of positive semidefinite matrices and a similar problem is also obtained when computing the eigenvalues of a matrix perturbed by the addition of a matrix of rank one. The method brackets the minimum of the function obtained in this linesearch by using the roots of successive rational bounds on its derivative and then uses these bounds to generate an approximation to the root. It is globally convergent with a robustness similar to that of the bisection method and it exhibits a superlinear order of convergence. A convergence analysis is presented together with numerical results illustrating the methods.

Key words. interior-point method, linesearch, perturbed eigenvalues, global convergence, superlinear convergence

AMS subject classifications. 65B99, 65F15, 90C05, 90C20, 90C25

1. Introduction.

Much interest has been generated by interior-point methods in recent years (see, e.g., [10], [11], [4], [5], [7], [17], [18] to name just a few). Many of those methods must solve at each step a subproblem consisting of a one-dimensional search for the optimal stepsize. Generally speaking, the methods proposed for determining the stepsize are variants of Newton's method or other higher-order methods. Important work has been done in this respect by, among others, W. Murray and M. H. Wright [13] and more recently by F. Jarre [8].

In this paper we determine the minimum of the function obtained in the linesearch for certain interior-point methods by computing the unique root of its derivative in the feasible region. This derivative exhibits a singularity on the boundary of the feasible region, which prohibits the straightforward application of Newton's method.

We develop an algorithm with a robustness similar to that of the bisecton method but with a superlinear order of convergence. It will be based on what will be called the "basic algorithm." This basic algorithm generates upper and lower bounds on the optimal stepsize. Its only purpose is to serve as a basis for the method we propose, which combines these bounds to generate an improved approximation to the stepsize. The way in which this is carried out will be determined by results from the asymptotic analysis of the basic algorithm.

Our method was mainly designed for problems with a convex quadratic objective function and convex quadratic constraints, which therefore includes linear and quadratic programming as special cases. The mathematical formulation of the problem is as follows:

$$\text{(P)} \qquad \min\left\{\frac{1}{2}x^TQx + a^Tx \mid \frac{1}{2}x^TQ_ix + a_i^Tx \le b_i, \quad i = 1,\dots,m\right\},$$

where Q , $Q_i \in I\!R^{(n,n)}$ are symmetric positive semidefinite, $x, a, a_i \in I\!R^n$ and $b_i \in I\!R$. We assume that this problem has a bounded solution. Such problems appear in a

* Received by the editors October 1, 1992; accepted for publication (in revised form) October 21, 1994.

† Faculty of Industrial Engineering and Management, Technion (Israel Institute of Technology), Haifa 32000, Israel. Current address: Department of Industrial Engineering and Management, Ben Gurion University, Beer-Sheva 84105, Israel.

number of fields; for instance, it was shown recently [1] that an important class of problems in truss topology design can be formulated in this way.

Exact linesearches are perhaps less important in interior-point methods for linear programming, where usually no linesearch is performed and a fixed proportion of the step to the boundary is taken. This situation may change drastically as soon as the problems become nonlinear (as in (P)). For such problems, significant savings in the number of iterations can be obtained by performing an accurate linesearch, as opposed to an approximate one or no linesearch at all. This is true in particular for truss topology design problems, as was observed during numerical experiments by Gil Roth at the Optimization Laboratory of the Technion. We have summarized some of those results in Table 1, where the number of iterations (i.e., the number of Newton steps) is given with exact (eight correct digits) and inexact linesearches (Armijo–Goldstein rule) for problems of various sizes. As can be seen from these results, the larger the size of the problems, the larger the saving in iterations. This makes performing an accurate linesearch worthwhile since each of these iterations requires $\mathcal{O}(n^3)$ arithmetic operations, whereas the linesearch typically executes only $\mathcal{O}(n)$ operations.

TABLE 1
Effect of an exact linesearch on the number of iterations.

# Variables	# Constraints	Exact Linesearch	Inexact Linesearch
99	150	30	40
127	1234	54	71
163	2040	55	79
193	2852	67	107
243	4492	73	107
343	8958	91	159
451	15556	112	210

Additional applications of our methods are in barrier methods for minimization over the cone of positive semidefinite matrices and in the computation of the eigenvalues of a matrix perturbed by the addition of a matrix of rank one.

In §2 the groundwork is laid for the basic algorithm, which is formally stated in §3 and for which the convergence analysis is given in §4. In §5 acceleration techniques for this algorithm are considered. Practical implementations of an accelerated algorithm are given in §6. Numerical results are presented in §9. In §7 we briefly describe how the ideas from §2 can be applied to the linesearch for other barrier methods, whereas in §8, we discuss the application of our methods to minimization over the cone of positive semidefinite matrices and to a problem in matrix theory.

2. Preliminaries. Let

$$\phi(x,\mu) = \frac{c^T x}{\mu} - \sum_{i=1}^{m} \ln\left(b_i - a_i^T x - \frac{1}{2}x^T Q_i x\right) \tag{1}$$

be the logarithmic barrier function used for problem (P), where, without loss of generality, we have made the assumption that the objective function is linear. (If the objective function f_0 is not linear, one can add a variable x_{n+1} and a constraint $f_0(x) \leq x_{n+1}$ and then minimize x_{n+1}.)

It is further assumed that d is a descent direction for ϕ, i.e., the function defined

by

$$g(t) = \phi(x + td, \mu) \tag{2}$$

satisfies $g'(0) < 0$. The problem we consider then consists of finding the unique feasible point t^* at which $g(t)$ obtains a minimum or at which $g'(t) = 0$. The convexity of g, together with the the fact that $g'(0) < 0$, guarantees that this point yields a lower function value at $t = t^*$ than at $t = 0$.

The function $g'(t)$ is given by

$$g'(t) = \frac{c^T d}{\mu} + \sum_{i=1}^{m} \frac{a_i^T d + x^T Q_i d + (d^T Q_i d)t}{(b_i - a_i^T x - \frac{1}{2} x^T Q_i x) - (a_i^T d + x^T Q_i d)t - \frac{1}{2}(d^T Q_i d)t^2} .$$

With $\zeta_i = \frac{a_i^T d + x^T Q_i d}{b_i - a_i^T x - \frac{1}{2} x^T Q_i x}$, $\eta_i = \frac{d^T Q_i d}{b_i - a_i^T x - \frac{1}{2} x^T Q_i x}$, and $K = \frac{c^T d}{\mu}$, this can be rewritten as

$$g'(t) = K + \sum_{i=1}^{m} \frac{\zeta_i + \eta_i t}{1 - \zeta_i t - \frac{1}{2}\eta_i t^2} . \tag{3}$$

Using the roots of the denominator in this last expression, it is not hard to see that the function $g'(t)$ is of the form

$$g'(t) = K + \sum_{i=1}^{m} \frac{\alpha_i}{1 - \alpha_i t} . \tag{4}$$

In what follows we write f for g'.

Let us now return to (4). With $I = \{i \mid \alpha_i \geq 0\}$ and $J = \{i \mid \alpha_i < 0\}$, we define the following quantities:

$$P(z) = \sum_{i \in I} \frac{\alpha_i}{1 - \alpha_i z}, \qquad N(z) = \sum_{i \in J} \frac{\alpha_i}{1 - \alpha_i z},$$

$$u = \max_I \{\alpha_i\}, \quad l = \min_J \{\alpha_i\}, \quad h = \max_J \{\alpha_i\}.$$

The next lemma states two inequalities which lie at the basis of the methods.

LEMMA 2.1. *For $t \geq 0$, the following inequalities hold:*

$$f(t) \geq K + P(0) - u + \frac{N(0)}{1 - ht} + \frac{u}{1 - ut}, \tag{5}$$

$$f(t) \leq K + \frac{P(0)}{1 - ut} + \frac{N(0)}{1 - lt} . \tag{6}$$

Proof. The proof follows immediately from the definitions of the quantities u, l, h, $N(z)$, and $P(z)$. □

We will denote the upper and lower bounds on $f(t)$, respectively, by

$$R(t) = K + \frac{P(0)}{1 - ut} + \frac{N(0)}{1 - lt} \quad \text{and} \quad S(t) = K + P(0) - u + \frac{N(0)}{1 - ht} + \frac{u}{1 - ut} .$$

The previous lemma is illustrated in Figure 1 with the function

$$f(t) = -5 - \frac{1}{1 + t} - \frac{2}{1 + 2t} + \frac{2}{1 - 2t} + \frac{3}{1 - 3t} ,$$

for which

$$S(t) = -3 - \frac{3}{1 + t} + \frac{3}{1 - 3t} \quad \text{and} \quad R(t) = -5 - \frac{3}{1 + 2t} + \frac{5}{1 - 3t} .$$

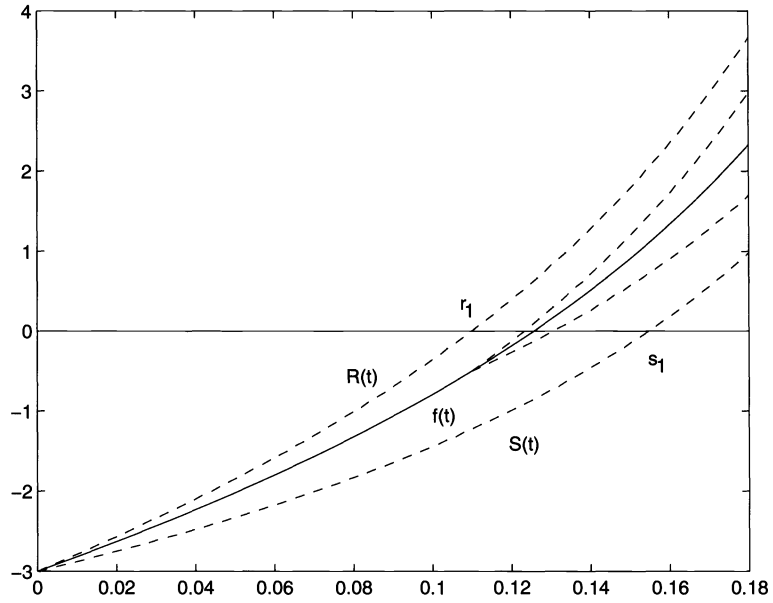


FIG. 1. *The function* $-5 - \frac{1}{1+t} - \frac{2}{1+2t} + \frac{2}{1-2t} + \frac{3}{1-3t}$ *with its corresponding upper and lower bounds.*

3. The basic algorithm. We now construct the basic algorithm. Inequalities (5) and (6) give rise to the following procedure for finding the root of $f(t)$: Compute the roots r and s of $R(t)$ and $S(t)$, respectively, (i.e., solve a quadratic equation) and perform the change of variables $v = t - r$. Since

$$\frac{\alpha_i}{1-\alpha_i t} = \frac{\alpha_i}{1-\alpha_i r - \alpha_i v},$$

then, with $\tilde{\alpha}_i = \frac{\alpha_i}{1-\alpha_i r}$, each term in $f(t)$ can be written as $\frac{\tilde{\alpha}_i}{1-\tilde{\alpha}_i v}$. With $\tilde{u}$, $\tilde{l}$, and $\tilde{h}$ defined analogously to $\tilde{\alpha}_i$, we obtain a new function for $v \geq 0$:

$$\tilde{f}(v) = K + \sum_{i=1}^{m} \frac{\tilde{\alpha}_i}{1-\tilde{\alpha}_i v},$$

and new bounds:

$$\tilde{S}(v) = K + P(r) - \tilde{u} + \frac{N(r)}{1-\tilde{h}v} + \frac{\tilde{u}}{1-\tilde{u}v} \quad \text{and} \quad \tilde{R}(v) = K + \frac{P(r)}{1-\tilde{u}v} + \frac{N(r)}{1-\tilde{l}v},$$

such that $\forall v \geq 0$, $\tilde{S}(v) \leq \tilde{f}(v) \leq \tilde{R}(v)$.

The new bounds on the root t^* are then given by

$$r + \tilde{r} \leq t^* \leq r + \tilde{s}\,,$$

where $\tilde{r}$ and $\tilde{s}$ are the roots of $\tilde{R}(v)$ and $\tilde{S}(v)$, respectively. This process is repeated until a suitable accuracy is reached. Note that for $t \geq 0$, $\alpha_i \leq \alpha_j$ iff $\frac{\alpha_i}{1-\alpha_i t} \leq \frac{\alpha_j}{1-\alpha_j t}$. This means that the indices

$$i_u = \{i | \alpha_i = u\}, \quad i_l = \{i | \alpha_i = l\}, \quad i_h = \{i | \alpha_i = h\}$$

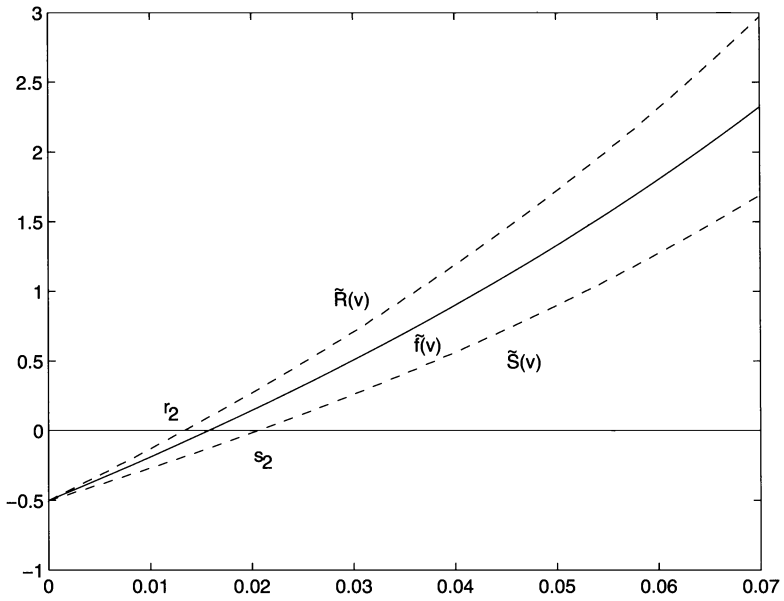


FIG. 2. *The second iteration of the algorithm.*

never change throughout the algorithm. The first two iterations are illustrated by Figures 1 and 2 respectively.

The basic algorithm can now formally be stated.

BASIC ALGORITHM
Input: $\{\alpha_i\}_{i=1}^m$, K, accuracy parameter ϵ
begin
Compute i_u, i_l, i_h; Set $t = 0$
while $|f(t)| > \epsilon$ **do**
begin
Compute $P = \sum_I \alpha_i$, $N = \sum_J \alpha_i$;
Set $u = \alpha_{i_u}$, $l = \alpha_{i_l}$, $h = \alpha_{i_h}$;
Compute the roots r of $R(t)$ and s of $S(t)$ on $[0, \frac{1}{u}]$;
Update $t \leftarrow t + r$ $\forall i$: $\alpha_i \leftarrow \frac{\alpha_i}{1-\alpha_i t}$
end
end

4. Convergence. Here we investigate the convergence of the basic algorithm. The results of this section are needed later in the construction of an accelerated algorithm.

Let r_k and s_k be the roots of $R(v)$ and $S(v)$, respectively, at the kth iteration and let

$$\Sigma_n = \sum_{i=1}^{n} r_i, \quad \alpha_i^* = \frac{\alpha_i}{1-\alpha_i t^*}, \quad u^* = \frac{u}{1-ut^*}, \quad l^* = \frac{l}{1-lt^*}, \quad h^* = \frac{h}{1-ht^*},$$

$$\gamma(z) = \frac{\sum_{i=1}^{m} \left(\frac{\alpha_i}{1-\alpha_i z}\right)^2}{\frac{u}{1-uz}P(z) + \frac{l}{1-lz}N(z)}, \qquad \nu(z) = \frac{\sum_{i=1}^{m} \left(\frac{\alpha_i}{1-\alpha_i z}\right)^2}{\frac{h}{1-hz}N(z) + \left(\frac{u}{1-uz}\right)^2},$$

$$\gamma^* = \gamma(t^*), \qquad \nu^* = \nu(t^*) .$$

The main results are then stated in the following theorem.

THEOREM 4.1. (1) *The sequences* $\{r_n\}$ *and* $\{s_n\}$ *converge to zero and satisfy*

$$r_n = \gamma^*(t^* - \Sigma_{n-1}) + \mathcal{O}((t^* - \Sigma_{n-1})^2), \tag{7}$$

$$s_n = \nu^*(t^* - \Sigma_{n-1}) + \mathcal{O}((t^* - \Sigma_{n-1})^2). \tag{8}$$

(2) *The sequence* $\{\Sigma_n\}$ *converges to* t^* *linearly, with the convergence rate given by*

$$\lim_{n\to\infty} \frac{t^* - \Sigma_n}{t^* - \Sigma_{n-1}} = 1 - \gamma^*. \tag{9}$$

Proof. We start by observing that $r_n \to 0$ as $n \to \infty$. This follows immediately from the construction of the algorithm, where r_n is a lower bound for $(t^* - \Sigma_{n-1})$, such that for $\forall n$, $\sum_{i=0}^{n} r_i \leq t^*$. Since $t^* < \infty$, $r_n \to 0$ as $n \to \infty$.

The equation for r_n is given by

$$K + \frac{P(\Sigma_{n-1})}{1 - \left(\frac{u}{1-u\Sigma_{n-1}}\right) r_n} + \frac{N(\Sigma_{n-1})}{1 - \left(\frac{l}{1-l\Sigma_{n-1}}\right) r_n} = 0. \tag{10}$$

Because we are interested in $n \to \infty$, which implies $r_n \to 0$, we write the previous equation as

$$K + P(\Sigma_{n-1})\left(1 + \frac{u}{1-u\Sigma_{n-1}} r_n\right) + N(\Sigma_{n-1})\left(1 + \frac{l}{1-l\Sigma_{n-1}} r_n\right) + \mathcal{O}(r_n^2) = 0,$$

and therefore, since the denominator in the right-hand side of the following equation is always strictly positive,

$$r_n = \frac{-(K + P(\Sigma_{n-1}) + N(\Sigma_{n-1}))}{\frac{u}{1-u\Sigma_{n-1}}P(\Sigma_{n-1}) + \frac{l}{1-l\Sigma_{n-1}}N(\Sigma_{n-1})} + \mathcal{O}(r_n^2). \tag{11}$$

Because $r_n \to 0$, $K + P(\Sigma_{n-1}) + N(\Sigma_{n-1}) = f(\Sigma_{n-1}) \to 0$ and, since t^* is the unique root of $f(t)$ on the interval considered, this means that $\Sigma_n \to t^*$.

Because $f(\Sigma_n) \to 0$ when $n \to \infty$, it follows from the construction of the algorithm that s_n is the positive root of a monotonic increasing function $S_n(v)$ $(v \geq 0)$ with $S_n(0) \to 0$ from below as $n \to \infty$. Because of this and because $S_n'(v)$ is bounded away from zero, its root s_n must also go to zero as $n \to \infty$.

Let us now proceed as we did for r_n. This gives

$$s_n = \frac{-(K + P(\Sigma_{n-1}) + N(\Sigma_{n-1}))}{\frac{h}{1-h\Sigma_{n-1}}N(\Sigma_{n-1}) + \left(\frac{u}{1-u\Sigma_{n-1}}\right)^2} + \mathcal{O}(s_n^2). \tag{12}$$

Furthermore,

$$\frac{\alpha_i}{1 - \alpha_i\Sigma_{n-1}} = \frac{\alpha_i}{1 - \alpha_i t^*} - \frac{\alpha_i^2}{(1 - \alpha_i t^*)^2}(t^* - \Sigma_{n-1}) + \mathcal{O}\left((t^* - \Sigma_{n-1})^2\right) . \tag{13}$$

This, together with equations (11) and (12), yields

$$r_n+\mathcal{O}((r_n)^2) = \frac{-(K+P(t^*)+N(t^*))+\left(\sum_{i=1}^m(\alpha_i^*)^2\right)(t^*-\Sigma_{n-1})}{u^*P(t^*)+l^*N(t^*)}+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right),$$

$$s_n+\mathcal{O}((s_n)^2) = \frac{-(K+P(t^*)+N(t^*))+\left(\sum_{i=1}^m(\alpha_i^*)^2\right)(t^*-\Sigma_{n-1})}{h^*N(t^*)+(u^*)^2}+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right).$$

Since $K+P(t^*)+N(t^*)=0$, we obtain for $n\to\infty$

$$r_n+\mathcal{O}((r_n)^2) = \gamma^*(t^*-\Sigma_{n-1})+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right), \tag{14}$$

$$s_n+\mathcal{O}((s_n)^2) = \nu^*(t^*-\Sigma_{n-1})+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right). \tag{15}$$

It remains to be shown that $\mathcal{O}(r_n)=\mathcal{O}(t^*-\Sigma_{n-1})$ and, analogously, that $\mathcal{O}(s_n)=\mathcal{O}(t^*-\Sigma_{n-1})$. We start by dividing both sides of (14) by r_n:

$$1+\mathcal{O}(r_n) = \frac{\gamma^*(t^*-\Sigma_{n-1})}{r_n}+\frac{\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right)}{r_n}.$$

As $n\to\infty$, the left-hand side in the previous equation goes to one. This means that the second term in the right-hand side must go to zero since otherwise the first term would go to infinity, implying that the equality cannot hold. The first term in the right-hand side therefore goes to one and, since $\gamma^*\neq 0$, this means that $\mathcal{O}(r_n)=\mathcal{O}(t^*-\Sigma_{n-1})$. Proceeding analogously for s_n, equations (14) and (15) can therefore finally be written as

$$r_n = \gamma^*(t^*-\Sigma_{n-1})+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right),$$

$$s_n = \nu^*(t^*-\Sigma_{n-1})+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right).$$

Note that $\gamma^*\leq 1$ and $\nu^*\geq 1$ as they should be.

In the remainder of the proof the rate of convergence for $\{\Sigma_n\}$ is derived. One has

$$\begin{aligned} t^*-\Sigma_n = t^*-\Sigma_{n-1}-r_n &= t^*-\Sigma_{n-1}-\gamma^*(t^*-\Sigma_{n-1})+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right)\\ &= (1-\gamma^*)(t^*-\Sigma_{n-1})+\mathcal{O}\left((t^*-\Sigma_{n-1})^2\right). \end{aligned}$$

Because Σ_n converges to t^*, dividing both sides by $(t^*-\Sigma_{n-1})$ and taking the limit as $n\to\infty$ concludes the proof. □

COROLLARY 4.2. *The length of the interval* $[r_n, s_n]$ *goes to zero at the same rate as* $\Sigma_n\to t^*$, *i.e.*,

$$\lim_{n\to\infty}\frac{s_{n+1}-r_{n+1}}{s_n-r_n} = 1-\gamma^*.$$

Proof. The proof is rather straightforward. Using (7) and (8) from Theorem 4.1, one has

$$\begin{aligned} \frac{s_{n+1}-r_{n+1}}{s_n-r_n} &= \frac{(s_{n+1}-t^*)+(t^*-r_{n+1})}{(s_n-t^*)+(t^*-r_n)}\\ &= \frac{\nu^*t^*-\nu^*\Sigma_n+\gamma^*t^*-\gamma^*\Sigma_n}{\nu^*t^*-\nu^*\Sigma_{n-1}+\gamma^*t^*-\gamma^*\Sigma_{n-1}}+\mathcal{O}(t^*-\Sigma_n)+\mathcal{O}(t^*-\Sigma_{n-1})\\ &= \frac{t^*-\Sigma_n}{t^*-\Sigma_{n-1}}+\mathcal{O}(t^*-\Sigma_n)+\mathcal{O}(t^*-\Sigma_{n-1}). \end{aligned}$$

Taking the limit as $n \to \infty$ on both sides and using (9) from Theorem 4.1 completes the proof. □

The asymptotic order of convergence results derived here is used in the next section to construct an accelerated algorithm.

5. Acceleration of the basic algorithm. A natural question is whether a better approximation to t^* could be obtained by adding to Σ_{n-1} not r_n, but a convex combination of r_n and s_n. For the sake of argument, suppose that we create a sequence $\{T_n\}$ such that $T_n = \Sigma_{n-1} + \epsilon r_n + \eta s_n$ and with $\epsilon + \eta = 1$, $\epsilon \geq 0$, $\eta \geq 0$. We shall investigate if there are any values for ϵ and η that might lead to an improvement in our approximation to t^*.

With (7) and (8) we have

$$\begin{aligned} t^* - T_n &= t^* - \Sigma_{n-1} - \epsilon r_n - \eta s_n \\ &= (1 - \epsilon\gamma^* - \eta\nu^*)(t^* - \Sigma_{n-1}) + \mathcal{O}\left((t^* - \Sigma_{n-1})^2\right). \end{aligned} \tag{16}$$

Dividing by $(t^* - \Sigma_{n-1})$ and taking the limit as $n \to \infty$ yields

$$\lim_{n\to\infty} \frac{t^* - T_n}{t^* - \Sigma_{n-1}} = 1 - \epsilon\gamma^* - \eta\nu^* \ . \tag{17}$$

The meaning of this expression is that the sequence $\{T_n\}$ might converge to t^* faster than the sequence $\{\Sigma_n\}$ if ϵ and η, in addition to being positive and adding up to one, also satisfy $\epsilon\gamma^* + \eta\nu^* = 1$, in which case the right-hand side in (17) becomes zero. The unique ϵ^* and η^* satisfying these conditions are given by

$$\epsilon^* = \frac{\nu^* - 1}{\nu^* - \gamma^*}, \qquad \eta^* = \frac{1 - \gamma^*}{\nu^* - \gamma^*} \ .$$

The case $\gamma^* = \nu^*$ can safely be excluded since for any $z \geq 0$ the case $\gamma(z) = \nu(z)$ is trivial. When this happens, the function $f(t)$ coincides with both its upper bound $R(t)$ and its lower bound $S(t)$ and t^* is computed in one iteration.

In reality however, γ^* and ν^* are unknown, so let us now consider the sequence $\{\Omega'_n\}$, defined as follows:

$$\Omega'_n = \Sigma_{n-1} + \epsilon_{n-1} r_n + \eta_{n-1} s_n \ ,$$

where

$$\epsilon_{n-1} = \frac{\nu(\Sigma_{n-1}) - 1}{\nu(\Sigma_{n-1}) - \gamma(\Sigma_{n-1})}, \qquad \eta_{n-1} = \frac{1 - \gamma(\Sigma_{n-1})}{\nu(\Sigma_{n-1}) - \gamma(\Sigma_{n-1})} \ .$$

Note that Ω'_n always lies inside the interval $[\Sigma_{n-1} + r_n, \Sigma_{n-1} + s_n]$ so that $\{\Omega'_n\}$ converges to t^*. From the preceding discussion, we see that this new sequence has a good chance to converge faster than $\{\Sigma_n\}$. Of course, to make use of the accelerated iterates Ω'_n in an efficient way, it is not sufficient to run the basic algorithm and at each iteration compute Ω'_n. Instead, one should restart the algorithm from the new accelerated iterate. However, the basic algorithm generates approximations from below to the root t^* of f, and it does not provide a direct means to continue from any point greater than t^*. This problem will be addressed in the next section. In the meantime, let us assume that the new accelerated iterate does lie to the left of the root and that it forms the starting point for the next accelerated iterate. This defines a sequence $\{\Omega_n\}$ as follows:

$$\Omega_n = \Omega_{n-1} + \epsilon_{n-1} r_n + \eta_{n-1} s_n,$$

where r_n, s_n, ϵ_{n-1}, and η_{n-1} are now computed using Ω_{n-1} instead of Σ_{n-1} and where $\Omega_1 = \Sigma_1$. For this sequence, the following theorem holds.

THEOREM 5.1. *If the sequence* $\{\Omega_n\}$ *lies to the left of* t^*, *then it converges quadratically to* t^*, *i.e.,*

$$\limsup_{n\to\infty} \frac{|t^* - \Omega_n|}{|t^* - \Omega_{n-1}|^2} < \infty .$$

Proof. The convergence follows from the fact that

$$\Omega_{n-1} + r_n \le \Omega_n \le \Omega_{n-1} + s_n$$

and from the convergence of the basic algorithm, under the assumption that the sequence $\{\Omega_n\}$ lies to the left of t^*.

With (13), after replacing Σ_{n-1} with Ω_{n-1}, we have

$$\epsilon_{n-1} = \epsilon^* + \mathcal{O}(t^* - \Omega_{n-1}), \qquad \eta_{n-1} = \eta^* + \mathcal{O}(t^* - \Omega_{n-1}).$$

We therefore obtain

$$\begin{aligned} t^* - \Omega_n &= t^* - \Omega_{n-1} - \epsilon_{n-1} r_n - \eta_{n-1} s_n \\ &= t^* - \Omega_{n-1} - (\epsilon^* + t^* - \Omega_{n-1})\gamma^*(t^* - \Omega_{n-1}) \\ &\qquad -(\eta^* + t^* - \Omega_{n-1})\nu^*(t^* - \Omega_{n-1}) + \mathcal{O}\left((t^* - \Omega_{n-1})^2\right) \\ &= t^* - \Omega_{n-1} - (\epsilon^*\gamma^* + \eta^*\nu^*)(t^* - \Omega_{n-1}) + \mathcal{O}\left((t^* - \Omega_{n-1})^2\right) \\ &= \mathcal{O}\left((t^* - \Omega_{n-1})^2\right). \end{aligned} \tag{18}$$

In the last step, we used the fact that $\epsilon^*\gamma^* + \eta^*\nu^* = 1$. Dividing both sides by $(t^* - \Omega_{n-1})^2$ and taking the limit as $n \to \infty$ completes the proof. □

This method of acceleration is of course not the only one possible. If we take, e.g., $\epsilon = \frac{1}{\gamma^*}$ and $\eta = 0$ in the expression for T_n, then we can see from (16) that this new sequence also converges faster to t^* than the original one. However, this is on the condition that it converges, since global convergence is no longer assured in this case. This way of accelerating the algorithm bears a close resemblance to Aitken's Δ^2-method (see, e.g., §5.10 in [19]). In that method, a new iterate A_n is computed based on Σ_{n-1}, Σ_n, and Σ_{n+1} as follows:

$$A_n = \frac{\Sigma_{n+1}\Sigma_{n-1} - \Sigma_n^2}{\Sigma_{n+1} - 2\Sigma_n + \Sigma_{n-1}} .$$

With $\Sigma_{n+1} = \Sigma_{n-1} + r_n + r_{n+1}$ and $\Sigma_n = \Sigma_{n-1} + r_n$ and after some algebra, we can rewrite this expression as

$$A_n = \Sigma_{n-1} + \frac{r_n}{1 - \frac{r_{n+1}}{r_n}} .$$

We recall that as $n \to \infty$

$$\frac{r_{n+1}}{r_n} \approx \frac{\gamma^*(t^* - \Sigma_n)}{\gamma^*(t^* - \Sigma_{n-1})} \approx 1 - \gamma^*,$$

so that $A_n \to \Sigma_{n-1} + \frac{1}{\gamma^*} r_n$, exactly as we proposed. In fact, preliminary numerical evidence shows that Aitken's method, combined with the basic algorithm, performs quite well. Since our algorithm always provides an interval around t^*, we can easily check whether the iterates obtained by Aitken's method converge without having to compute any function values.

6. Practical implementations of the accelerated algorithm. We now describe two practical implementations of the algorithm. As was mentioned in the previous section, to make use of the accelerated iterate Ω'_n in an efficient way, one should restart the algorithm from this new iterate. However, since some of the accelerated iterations could lie to the right of the root t^* of f and since the basic algorithm needs a point on the left-hand side of that root, we adopt, as a first implementation, the following modifications:

(1) if an accelerated iterate falls to the left of t^*, continue in the usual way, i.e., compute the next r_n, s_n and the new accelerated iteration;

(2) if an accelerated iterate falls to the right of the root, perform one step of the secant method (see §5.9 in [19]) with this point and the last point on the left of t^*;

(3) if this new point lies to the left of t^*, continue as usual with the accelerated algorithm and if it falls to the right of t^*, continue with the secant method either until convergence or until a point on the left of t^* is generated, whichever comes first;

(4) if we do obtain a point on the left of t^* that is better than the last iterate on the left of t^*, continue with the accelerated algorithm; otherwise perform a secant step with the best points on the left- and right-hand sides of t^* and continue as in (3).

This implementation ensures global convergence. As far as the order of convergence is concerned, only the following three scenarios are encountered.

(1) The new accelerated iterate always lies to the left of the root. In this case, it follows from Theorem 5.1 that the algorithm converges quadratically.

(2) Some accelerated iterate falls to the right of the root and from this point on the secant method generates iterates which always lie to the right of the root. The order of convergence will be that of the secant method, which is $\frac{1+\sqrt{5}}{2} = 1.618\ldots$.

(3) The new accelerated iterate falls to the right of the root, the following secant iteration then throws it back to the left of the root, and the iterates continue to oscillate in this way. Here the order of convergence will be $\sqrt{1+\sqrt{5}} = 1.798\ldots$.

There are only these three cases to consider because from some point on, the next iterate will consistently lie either to the left or to the right of the root. This is true both for our accelerated algorithm and for the secant method.

To understand this for our algorithm, it is sufficient to look at (18). The factor in front of $(t^* - \Sigma_{n-1})^2$ determines on which side of t^* the next iterate will lie asymptotically. (In case this factor is zero; the same is true for higher-order terms.) This factor depends only on t^* and therefore does not change. A similar reasoning holds for the secant method based on (5.9.9) in [19]. If m is the total number of terms in the expression for f, then the accelerated algorithm must perform about $6m$ basic operations per iteration. Of course, whenever a secant step is taken, no derivatives must be computed.

A modification of this algorithm is obtained by approximating γ^* using the ratio of the lengths of two succesive intervals and by approximating ν^* using the approximation to γ^* and the ratio $\frac{s_n}{r_n}$, which converges to $\frac{\nu^*}{\gamma^*}$. This modified algorithm, which we call the *approximate accelerated algorithm*, maintains global convergence as long as the approximations to γ^* and to ν^* are less than one, and greater than one, respectively. Whenever these conditions are not satisfied, $\gamma(z)$ and $\nu(z)$ must be computed. Here at each iteration only about $4m$ basic operations must be executed, except in the case where $\gamma(z)$ and $\nu(z)$ must be computed, in which case the number of operations is about $6m$.

A second implementation consists of reversing the search direction whenever a

point falls to the right of the root. The accelerated algorithm can then be applied to the negative of the function with $-h$ (the negative of the largest negative α_i) playing the role of u (the largest positive α_i). This method converges quadratically, but derivatives must be computed at every step.

7. Other interior-point methods. In this section we briefly describe the application of the philosophy of the previously developed method to the linesearch for other interior-point methods, namely, the methods based on the *center function*, on the *Carroll*-type barrier function, and on *modified barrier functions*. We will not go into great detail because the development is basically a repetition of that of the logarithmic barrier function case.

7.1. The center function. With L an upper bound to the optimal cost for the linear programming (LP) problem, n the dimension of the problem, and $q \geq n$ constant, the center function [6] is given by

$$\psi(x, L) = -q \ln(L - c^T x) - \sum_{i=1}^{m} \ln(b_i - a_i^T x) .$$

Assume now, as before, that d is a descent direction for the function G defined by

$$G(t) = \psi(x + td).$$

From the definition of $G(t)$, we have

$$G(t) = -q \ln(L - c^T(x + td)) - \sum_{i=1}^{m} \ln(b_i - a_i^T(x + td)) .$$

The derivative of this function is given by

$$G'(t) = \frac{q(c^T d)}{(L - c^T x) - (c^T d)t} + \sum_{i=1}^{m} \frac{a_i^T d}{(b_i - a_i^T x) - (a_i^T d)t} ,$$

which can be rewritten as

$$G'(t) = \frac{q\beta}{1 - \beta t} + \sum_{i=1}^{m} \frac{\alpha_i}{1 - \alpha_i t} ,$$

where $\beta = \frac{c^T d}{L - c^T x}$ and $\alpha_i = \frac{a_i^T d}{b_i - a_i^T x}$. In what follows we write $F(t)$ for $G'(t)$.

The function F is analytic and monotonic increasing for $t \geq 0$, with a simple pole at the inverse of the maximum between β and the largest positive α_i.

We can proceed exactly as in the barrier function case, with $\frac{q\beta}{1-\beta t}$ replacing K. At each iteration we still must solve a quadratic equation, and the same convergence analysis as before can be applied here. The same is true for the acceleration techniques we used before.

7.2. The Carroll barrier function. For the Carroll barrier function [2], which uses the reciprocal instead of the logarithm, the function to be minimized in the linesearch is given by

$$g(t) = \frac{c^T(x + td)}{\mu} + \sum_{i=1}^{m} \frac{1}{b_i - a_i^T(x + td) - \frac{1}{2}(x + td)^T Q_i (x + td)} .$$

Again, by taking the derivative, a function of the following form is obtained:

$$g'(t) = K + \sum_{i=1}^{m} \frac{\beta_i}{(1-\alpha_i t)^2}, \tag{19}$$

where α_i and β_i have the same sign. Bounds can be found for this function analogously to the logarithmic case. The difference is that now a fourth-order equation must be solved, a computation that is still negligible, compared to the computation of one function value.

7.3. Modified barrier functions. The *modified logarithmic barrier function* [15] for our problem is given by

$$\phi(x,u,k) = c^T x - \frac{1}{k}\sum_{i=1}^{m} u_i \ln\left(k\left(b_i - a_i^T x - \frac{1}{2}x^T Q_i x\right) + 1\right), \tag{20}$$

where k is the barrier parameter (which is taken to $+\infty$) and the positive numbers $\{u_i\}$ are approximations to the Lagrange multipliers. For the linesearch a function of the following form is obtained:

$$K + \sum_{i=1}^{m} \frac{\beta_i \alpha_i}{1-\alpha_i t},$$

where $\beta_i > 0$. The presence of the β_i's does not hinder the application of our methods.

The *modified Carroll barrier function* can be treated similarly.

8. Other applications.

8.1. Minimization over the cone of positive semidefinite matrices. A linesearch problem of the same structure appears when using a logarithmic barrier function for minimization over the cone of positive semidefinite matrices (see, e.g., [8] [9]). Let $A(x) = A^{(0)} + \sum_{i=1}^{m} x_i A^{(i)}$ be positive definite for some x and let the barrier function be given by $\phi(x) = -\log\det(A(x))$. It can then be shown that

$$\phi'(x+td) = -\sum_{i=1}^{n} \frac{\lambda_i}{1+\lambda_i t},$$

where λ_i are the eigenvalues of $A(x)^{-1/2}\left(\sum_{i=1}^{m} d_i A^{(i)}\right) A(x)^{-1/2}$. This leads to the same problem considered earlier.

8.2. Perturbed eigenvalues. Let us now briefly consider an entirely different kind of problem from matrix theory. Suppose that a matrix $A \in \mathbb{R}^{(n,n)}$ is perturbed by the addition of a matrix of rank one. The eigenvalues of this new matrix will lie between the eigenvalues of A and, as can readily be seen from [12, p. 63], are given by the roots of a function of the form

$$K + \sum_{i=1}^{n} \frac{\beta_i \alpha_i}{1-\alpha_i t},$$

where $\beta_i > 0$. This function is discontinuous at $t = \frac{1}{\alpha_i}$ for $i = 1, \ldots, n$, but continuous and ascending on the intervals between those points. The roots of this function are to be found on these intervals. All one has to do for each root is find a point on each interval where the function value is negative, perform a translation of the Y-axis to make this point the new zero, and then proceed exactly as in the linesearch problem.

9. Numerical results. We now present some numerical results obtained with the algorithms we have developed and compare them to three methods: the so-called *reduced Newton* method, the safeguarded Newton method, and a method based on [13], [14]. The reduced Newton method is Newton's method with a specific stepsize rule, the basic iteration of which, with $\Delta x = \frac{f(x)}{f'(x)}$, is given by

$$x_{i+1} = x_i - \frac{\Delta x_i}{1 + |\Delta x_i| \sqrt{f'(x_i)}}.$$

Its global convergence was proved in [11]. The asymptotic rate of convergence is quadratic. The safeguarded Newton method we use here consists of a Newton step if this yields a point in the current interval containing the root; otherwise it consists of a bisection step. The method based on [13], [14], which will be denoted "M-W," approximates the given function at a current point by a function of the form

$$a + \frac{b}{1 - ut},$$

where a and b are chosen such that the approximation and its derivative agree with the given function and its derivative, respectively, at the current point. The next iterate is determined by the root of the approximation. All three methods perform roughly $6m$ basic operations per iteration.

In what follows, the number of iterations for the aforementioned algorithms are compared, starting from zero (except for the reduced Newton method, which was started from the middle of the feasible interval), when applied to the following functions:

$$F_1(t) = -5 \times 10^3 + \sum_{i=1}^{2500} \frac{i^{10}}{(2500)^{10} - (i)^{10}t} - \sum_{i=1}^{1500} \frac{i^2}{1500 + (i)^2 t},$$

$$F_2(t) = -5 \times 10^3 + \sum_{i=1}^{4500} \frac{i}{4500 - it} - \sum_{i=1}^{300} \frac{i}{300 + it},$$

$$F_3(t) = -5 + \sum_{i=1}^{1500} \frac{i}{1500 - it} - \sum_{i=1}^{3500} \frac{i}{3500 + it},$$

$$F_4(t) = -5 \times 10^3 + \sum_{i=1}^{3000} \frac{\sqrt{i}}{3000 - (\sqrt{i})t} - \sum_{i=1}^{3500} \frac{i^2}{3500 + (i)^2 t},$$

$$F_5(t) = -5 \times 10^2 + \sum_{i=1}^{3000} \frac{i}{2 - it} - \sum_{i=1}^{3500} \frac{i^{\frac{3}{2}}}{1 + \left(i^{\frac{3}{2}}\right) t},$$

$$F_6(t) = -5 \times 10^6 + \sum_{i=1}^{3000} \frac{i}{2 - it} - \sum_{i=1}^{3500} \frac{i^{\frac{3}{2}}}{1 + \left(i^{\frac{3}{2}}\right) t},$$

$$F_7(t) = -5 \times 10^6 + \sum_{i=1}^{3000} \frac{i^{10}}{(2500)^{10} - (i)^{10}t} - \sum_{i=1}^{3000} \frac{i^{\frac{3}{2}}}{1 + \left(i^{\frac{3}{2}}\right) t},$$

$$F_8(t) = -5 \times 10^3 + \sum_{i=1}^{2500} \frac{i^{10}}{(2500)^{10} - (i)^{10}t} - \sum_{i=1}^{300} \frac{i}{300 + it},$$

$$F_9(t) = -5 \times 10^6 + \sum_{i=1}^{5000} \frac{7i}{10 - 7it} - \sum_{i=1}^{5000} \frac{i}{2 + it},$$

$$F_{10}(t) = -5 \times 10^7 + \sum_{i=1}^{5000} \frac{\sqrt{i}}{3000 - (\sqrt{i})t} - \sum_{i=1}^{5000} \frac{i}{5 + it}.$$

These test functions range from relatively well-behaved convex functions such as in Figures 1 and 2 to rather pathological ones as in Figure 3. The computations were carried out in double precision until the function value at the root was approximately $F(t^*) \approx 10^{-9}$. The results are summarized in Table 2, where in the column for the approximate accelerated algorithm we have listed the total number of operations, divided by $6m$. This is to facilitate the comparison to the other methods, which execute about $6m$ basic operations per iteration, as opposed to $4m$ in the approximate accelerated algorithm. The actual number of iterations for this method appears in parentheses. The letters "NC" mean that no convergence was achieved after 150 iterations. In addition, the convergence rate for the basic algorithm was checked and found to be in agreement with its theoretical value from Theorem 4.1.

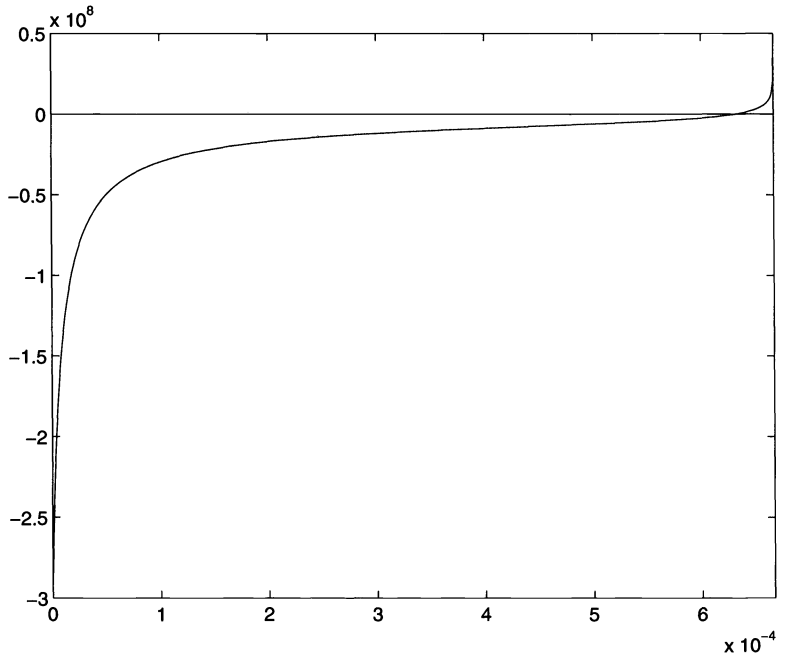


FIG. 3. *The function* F_6.

We find that the approximate accelerated algorithm performs slightly better than the pure accelerated algorithm. It should be noted that in all these examples the accelerated iterates consistently fell to the left of the root, so it was never necessary for the accelerated algorithm to perform any secant steps or reversions of the search direction nor for the approximate accelerated algorithm to compute $\gamma(z)$ or $\nu(z)$. We see that, in general, our methods perform better than the reduced and safeguarded Newton methods and are at least comparable with the M-W method. The sometimes rather excessive number of iterations in the safeguarded Newton method are caused

TABLE 2
Comparison of computational performance.

Function	Red. Newton	Sfg. Newton	M-W	Acc. Alg.	Aprx. Acc. Alg.
F_1	39	19	15	6	5.3 (8)
F_2	28	8	7	13	12 (18)
F_3	11	7	8	11	10 (15)
F_4	104	31	19	7	6 (9)
F_5	44	14	13	12	12 (18)
F_6	65	16	15	16	15.3 (23)
F_7	NC	32	13	4	3.3 (5)
F_8	31	19	8	7	7.3 (11)
F_9	23	7	7	13	8.7 (13)
F_{10}	NC	38	6	4	2.7 (4)

by having to perform many bisection steps to ensure convergence. However, when the Newton iteration is rejected, a bisection step is often worse than the current best point, causing subsequent Newton iterates also to be rejected. This fact was recognized in a report by P. E. Gill and W. Murray [3] in which they developed a more elaborate safeguarding scheme. Although this might slightly reduce the number of iterations for the safeguarded Newton method, there is little doubt that it would not materially alter the results.

The numerical experiments were run on a SUN 4/370.

Acknowledgments. I would like to thank Professor Ariela Sofer and the referees (one of whom, Professor Walter Murray, chose to identify himself) for their helpful comments, which improved the paper considerably.

REFERENCES

[1] A. BEN-TAL AND M. P. BENDSŒ, *A new method for optimal truss topology design*, SIAM J. Optim., 3 (1993), pp. 322–358.

[2] C. W. CARROLL, *The created response surface technique for optimizing nonlinear restrained systems*, Oper. Res., 9 (1961), pp. 169–184.

[3] P. E. GILL AND W. MURRAY, *Safeguarded Steplength Algorithms for Optimization Using Descent Methods*, Report NAC 37, National Physical Laboratory, Teddington, England, 1974.

[4] P. E. GILL, W. MURRAY, M. A. SAUNDERS, J. A. TOMLIN, AND M. H. WRIGHT, *On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method*, Math. Programming, 36 (1986), pp. 183–209.

[5] C. C. GONZAGA, *An Algorithm for Solving Linear Programming Problems in $\mathcal{O}(n^3L)$ Operations*, Memorandum UCB/ERL M87/10, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, 1987.

[6] P. HUARD, *Resolution of mathematical programming with nonlinear constraints by the methods of centers*, in Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1967.

[7] F. JARRE, *The Method of Analytic Centers for Smooth Convex Programs*, Ph.D. thesis, Institut fur Angewandte Mathematik und Statistik, University of Würzburg, Würzburg, Germany, 1989.

[8] ———, *An Efficient Line-search for Logarithmic Barrier Functions*, Tech. report 188, Institut fur Angewandte Mathematik, University of Würzburg, Würzburg, Germany, 1992.

[9] ———, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[10] N. KARMARKAR, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[11] YU.E. NESTEROV AND A.S. NEMIROVSKY, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Central Economic and Mathematical Institute, Moscow, USSR, 1989.

[12] G. P. McCormick, *Nonlinear Programming*, John Wiley & Sons, New York, 1983.
[13] W. Murray and M. H. Wright, *Efficient Linear Search Algorithms for the Logarithmic Barrier Function*, Report SOL 76-18, Department of Operations Research, Stanford University, Stanford, CA, 1976.
[14] ———, *Line search procedures for the logarithmic barrier function*, SIAM J. Optim., 4 (1994), pp. 229–246.
[15] R. Polyak, *Modified barrier functions (theory and methods)*, Math. Programming, 54 (1992), pp. 177–222.
[16] J. Renegar, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–94.
[17] J. Renegar and M. Shub, *Simplified Complexity Analysis for Newton LP Methods*, Tech. report 807, School of Operations Research and Industrial Engineering, College of Engineering, Cornell University, Ithaca, NY, 1988.
[18] C. Roos and J.-Ph. Vial, *Long Steps With the Logarithmic Penalty Barrier Function in Linear Programing*, Tech. report 89-44, Faculty of Mathematics and Informatics/Computer Science, Delft University of Technology, Delft, The Netherlands, 1989.
[19] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 299–315, May 1996
© 1996 Society for Industrial and Applied Mathematics
003

DIFFERENTIABLE PIECEWISE QUADRATIC EXACT PENALTY FUNCTIONS FOR QUADRATIC PROGRAMS WITH SIMPLE BOUND CONSTRAINTS*

WU LI†

Abstract. The quadratic program with simple bound constraints is reformulated as unconstrained minimization of a differentiable piecewise quadratic function. The two problems have the same set of local solutions, the same set of isolated local solutions, and the same set of global solutions. Unlike other penalty functions, a parameter involved in the unconstrained reformulation can be easily determined by the spectrum radius of the Hessian of the objective function in the original quadratic program. Moreover, the exact penalty function can also be derived from the Hestenes–Powell–Rockafellar augmented Lagrangian function for two-sided inequality constrained minimization problems by using Fletcher's multiplier function.

Key words. differentiable exact penalty functions, quadratic splines, quadratic programs with simple bound constraints, (isolated) local solutions, global solutions

AMS subject classifications. Primary 90C20; Secondary 90C30

1. Introduction.

Consider the following quadratic program with simple bound constraints:

$$\min_{l\le x\le u} \frac{1}{2}x^T Mx - b^T x, \tag{1}$$

where M is an $n \times n$ symmetric matrix, $b \in \mathbf{R}^n$ (a vector of n components), and l, u are vectors of n components with $l < u$. (Note that if $l_i = u_i$, then $x_i \equiv l_i$ and one can replace x_i by l_i in (1) and reformulate the problem so that $l < u$.) Some components of l or u may be $-\infty$ or $+\infty$.

One can take advantage of the special structure of (1) to design numerical algorithms for solving (1) [2]–[9], [11, 12, 16, 14, 18], [20]–[24], [28]–[30], [32], [34]–[36], [41]–[45], [50]–[52]. The article by Moré and Toraldo [42] contains references on quadratic programs with simple bound constraints in engineering applications.

When M is positive semidefinite, Li and Swetits reformulated (1) as the following unconstrained minimization problem [32, 33]:

$$\min_{x\in\mathbf{R}^n} \Psi_0(x), \tag{2}$$

where $\Psi_0(x)$ is a convex quadratic spline defined as follows:

$$\begin{aligned}\Psi_0(x) := {} & \frac{1}{2}x^T(E - E^2)x - x^T Eh \\ & + \frac{1}{2}\|(l - (Ex + h))_+\|^2 + \frac{1}{2}\|((Ex + h) - u)_+\|^2.\end{aligned} \tag{3}$$

Here α is a positive constant such that $0 < \alpha\|M\| < 1$, $\|M\|$ is the 2-norm of the matrix M, $E := I - \alpha M$, and $h := \alpha b$. It was proved that x^* is a solution to (1) if and

* Received by the editors April 27, 1994; accepted for publication (in revised form) September 14, 1994.

† Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529 (wuli@math.odu.edu). The research of this author was partially supported by National Aeronautics and Space Administration/Langley Research Center grant NCC-1-68 supplement-15 and Air Force Office of Scientific Research grant 94-NM-390.

only if x^* is a minimizer of $\Psi_0(x)$ [32, 33]. Therefore, algorithms for unconstrained minimization of the convex quadratic spline $\Psi_0(x)$ can be used to find a solution to (1).

When M is a positive definite matrix, $\Psi_0(x)$ is actually a strictly convex quadratic spline. In this case, one can use either a Newton method with line search to find the unique solution x^* of (1) in finite iterations [32, 33] or a conjugate gradient method to generate a sequence of iterates which converge linearly to x^* [30]. Moreover, the conjugate gradient method finds x^* in finite iterations if x^* is nondegenerate [30]. In general, it is very easy to design a linearly convergent descent method for finding a minimizer of a convex quadratic spline which is bounded below on $\mathbf{R}^n$, even if the set of minimizers of the convex quadratic spline is unbounded [29]. As a consequence, by the unconstrained reformulation (2), one can easily generate a sequence of iterates which converges linearly to a solution of (1) when M is positive semidefinite and (1) has a solution [29].

The unconstrained reformulation of (1) was based solely on the observation that x^* is a stationary point of $\Psi_0(x)$ (i.e., $\Psi_0'(x^*) = 0$) if and only if x^* satisfies the first-order optimality conditions of (1), i.e., $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point of (1) [32, 33]. Therefore, it is unclear whether the minimizers of $\Psi_0(x)$ correspond to the solutions of (1) when M is not positive semidefinite. To establish the correspondence without the convexity assumption, we have to relate the second-order optimality conditions of (1) to those of (2).

When $0 < 2\alpha\|M\| < 1$, based on careful manipulations of the second-order optimality conditions and an observation that $\Psi_0(x)$ is actually the sum of the objective function of (1) and a quadratic spline penalty term, we prove that x^* is a local solution (or an isolated local solution) of (1) if and only if x^* is a local minimizer (or an isolated local minimizer) of $\Psi_0(x)$. Moreover, x^* is a global solution of (1) if and only if x^* is a global minimizer of $\Psi_0(x)$. In other words, the unconstrained reformulation (2) is "absolutely" equivalent to (1) when $0 < 2\alpha\|M\| < 1$. We want to point out that $\Psi_0(x)$ can also be derived from Hestenes–Powell–Rockafellar's augmented Lagrangian function for two-sided inequality constraints by using Fletcher's multiplier function. See §2 for details.

In general, one can use exact penalty functions or exact augmented Lagrangian functions to derive unconstrained reformulations of a constrained minimization problem (cf. [1, 13]). Consider the following constrained minimization problem:

$$\min_{x \in X} f(x), \tag{4}$$

where $X \subset \mathbf{R}^n$ is the feasible set defined by some linear or nonlinear constraints. To reformulate (4) as an unconstrained minimization problem, we introduce an exact penalty function $F(x, \epsilon)$ which contains a penalty parameter ϵ and consider the following unconstrained minimization problem:

$$\min_{x \in \mathbf{R}^n} F(x, \epsilon). \tag{5}$$

The most important property of an exact penalty function is that its local (or global) minimizers are local (or global) solutions of the original constrained minimization problem, which is the main focus of most of the literature on this subject, as Han and Mangasarian note in [25]. This property ensures that any unconstrained minimization technique for finding a local (or global) minimizer of the penalty function yields a local (or global) solution of the original constrained problem. Usually, there is a threshold

ϵ^* for the penalty parameter such that $F(x,\epsilon)$ is exact if $0 < \epsilon \leq \epsilon^*$. However, in almost all cases, there is no simple way to compute the value of ϵ^*. Also, as pointed out by Di Pillo and Grippo [13], in practice the threshold ϵ^* could only be established with reference to some compact subset $\mathcal{D}$ of $\mathbf{R}^n$. That is, in practice, one could get only a threshold ϵ^* such that, for $0 < \epsilon \leq \epsilon^*$, $F(x,\epsilon)$ is exact for x in $\mathcal{D}$. To measure the degree of exactness of a penalty function, formal definitions of exactness were introduced by Di Pillo and Grippo [13]. By Di Pillo and Grippo's definition [13], for $0 < 2\alpha\|M\| < 1$, $\Psi_0(x)$ is a strongly exact penalty function for (1) with respect to the set $\mathcal{D} \equiv \mathbf{R}^n$.

There are two other unconstrained reformulations of (1): one given by Grippo and Lucidi [20] and one by Coleman and Hulbert [5]. Both reformulations are valid only for the case that the feasible region $\{x : l \leq x \leq u\}$ is compact.

In [20], Grippo and Lucidi started from the definition of differentiable multiplier functions that yield an estimate of the Karush–Kuhn–Tucker multipliers as explicit functions of x. Then they constructed a penalty function $P(x,\epsilon)$ containing barrier terms on a perturbation of the constraints. The penalty function $P(x,\epsilon)$ is a differentiable piecewise rational function in an open neighborhood $\mathcal{D}$ of the compact feasible region $\{x : l \leq x \leq u\}$. The definition of the threshold ϵ^* involves the maximum value of $\|Mx - b\|_\infty$ (the supremum norm of the vector $(Mx - b)$) over the feasible region $\{x : l \leq x \leq u\}$ and the maximum value of some nonlinear expression of x and ϵ. For $0 < \epsilon \leq \epsilon^*$, they proved that x^* is a global minimizer of $P(x,\epsilon)$ in $\mathcal{D}$ if and only if x^* is a global solution of (1). Moreover, any local minimizer x^* of $P(x,\epsilon)$ is a local solution of (1). The converse is also true when $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point satisfying strict complementarity conditions. In a separate paper [21] based on the penalty function $P(x,\epsilon)$, they proposed Newton-type algorithms to solve (1). Under suitable assumptions, finite termination of a Newton method was established. In [14], Facchinei and Lucidi adapted Grippo and Lucidi's approach to tackle unbounded feasible sets.

Coleman and Hulbert's penalty function is a nondifferentiable piecewise quadratic function. They started with an ℓ_1 penalty term and the transform $y := -(Mx - b)$ that yield a penalty function $f(y)$. Under the assumption that M is positive definite and the unique solution x^* of (1) is nondegenerate, they proved that a Newton-type method generates a sequence of iterates $\{y^k\}$ which converge superlinearly to $y^* := -(Mx^* - b)$ [5]. Note that one must solve the linear system $Mx = (b - y^*)$ to get the solution x^*. One interesting feature of Coleman and Hulbert's penalty function is that there is no penalty parameter involved.

In comparison with Grippo and Lucidi's penalty function $P(x,\epsilon)$ and Coleman and Hulbert's penalty function $f(y)$, $\Psi_0(x)$ is simpler than $P(x,\epsilon)$: a differentiable piecewise quadratic function versus a differentiable piecewise rational function. $\Psi_0(x)$ is "smoother" than $P(x,\epsilon)$ or $f(y)$: differentiability on $\mathbf{R}^n$ versus singularity of $P(x,\epsilon)$ on the boundary of $\mathcal{D}$ (induced by the barrier terms) or nondifferentiability of $f(y)$. $\Psi_0(x)$ is "more exact" than $P(x,\epsilon)$ or $f(y)$: equivalence of local solutions without strict complementarity conditions for $\Psi_0(x)$ versus local solutions with strict complementarity conditions for $P(x,\epsilon)$ or with a strict convexity assumption on (1) for $f(y)$. The penalty parameter for $\Psi_0(x)$ is easier to compute than that for $P(x,\epsilon)$: the trivial estimate of the threshold versus the complicated estimate of the threshold by maximization of linear and nonlinear functions. $\Psi_0(x)$ preserves the convexity of the original minimization problem (i.e., if the original quadratic program (1) is convex, then $\Psi_0(x)$ is also convex), while $P(x,\epsilon)$ or $f(y)$ does not. $\Psi_0(x)$ is valid for any

feasible set while $P(x,\epsilon)$ or $f(y)$ is defined only for compact feasible sets.

The paper is organized as follows. In §2, we outline how (1) can be reformulated as unconstrained minimization of a quadratic spline penalty function $\Psi(x)$ and show that the penalty function $\Psi(x)$ can be derived from Hestenes–Powell–Rockafellar's augmented Lagrangian function by using Fletcher's multiplier function. An important relation between the penalty function $\Psi(x)$ and the objective function of (1) is also given. In §3, we relate the second-order optimality conditions for (1) to those for unconstrained minimization of $\Psi(x)$. In §4, we prove that (1) and the unconstrained reformulation have the same set of local solutions, the same set of isolated local solutions, and the same set of global solutions. Conclusions and final remarks are given in §5.

Now we conclude this section by giving some terminology and notation used in the paper.

For simplicity, we use $f'(x)$ to denote the gradient of $f(x)$ (as a column vector) and use $f''(x)$ to denote the Hessian of $f(x)$. A real-valued function $f(x)$ on $\mathbf{R}^n$ is said to be a quadratic spline if the gradient $f'(x)$ of $f(x)$ is a piecewise linear mapping from $\mathbf{R}^n$ to $\mathbf{R}^n$. That is, a quadratic spline is a continuously differentiable piecewise quadratic function. The 2-norm $\|\cdot\|$ on $\mathbf{R}^n$ is defined as $\|x\| := \left(\sum_{i=1}^n x_i^2\right)^{1/2}$ and the 2-norm of an $n \times n$ matrix B is defined as $\|B\| := \sup\{\|Bx\| : x \in \mathbf{R}^n \text{ with } \|x\| = 1\}$. The transpose of a matrix B or a vector x is denoted by B^T or x^T. For $x, y \in \mathbf{R}^n$, $x \le y$ means $x_i \le y_i$ for $1 \le i \le n$, where x_i or y_i denotes the ith component of x or y. Let $(z)_l$ (or $(z)^u$) be the lower (or upper) truncation of z by l (or u), whose ith component is $\max\{l_i, z_i\}$ (or $\min\{u_i, z_i\}$). By convention, z_+ is a vector whose ith component is $\max\{z_i, 0\}$. For convenience, we use Λ to denote a diagonal matrix whose diagonal entries are either 0 or 1. The $n \times n$ identity matrix is written as I and $\Lambda_c := I - \Lambda$. A vector $x^* \in X$ is said to be a local solution of (4) if there exists a positive constant δ such that $f(x) \ge f(x^*)$ for $x \in X$ with $\|x - x^*\| \le \delta$. A vector $x^* \in X$ is said to be an isolated local solution of (4) if there exists a positive constant δ such that $f(x) > f(x^*)$ for $x \in X$ and $0 < \|x - x^*\| \le \delta$. A vector $x^* \in X$ is said to be a global solution of (4) if $f(x) \ge f(x^*)$ for all $x \in X$.

2. Unconstrained reformulations. In this section we outline how the quadratic program with simple bound constraints, (1), can be reformulated as unconstrained minimization of a quadratic spline $\Psi(x)$ [32, 33] and show that $\Psi(x)$ can be derived from Hestenes–Powell–Rockafellar's augmented Lagrangian function by using Fletcher's multiplier function. Finally, an important relation between $\Psi(x)$ and the objective function of (1) is given.

It is well known from the Karush–Kuhn–Tucker conditions that if x is a solution of (1), then there exists $w \in R^n$ such that, for $1 \le i \le n$,

$$\begin{aligned} & Mx - b - w = 0, \\ & x_i = l_i \quad \text{if} \quad w_i > 0, \\ & x_i = u_i \quad \text{if} \quad w_i < 0, \\ & l_i \le x_i \le u_i \quad \text{if} \quad w_i = 0. \end{aligned} \tag{6}$$

One can verify that (x^*, w^*) is a solution of (6) if and only if

$$w^* = Mx^* - b \quad \text{and} \quad x^* = (x^* - \alpha w^*)_l^u, \tag{7}$$

where α is any positive constant (cf. [33]). By substituting $w^* = Mx^* - b$ into (7), we observe that (6) is equivalent to a system of piecewise linear equations. However,

by multiplying a special nonsingular matrix on both sides of (7), one can prove that (7) is equivalent to the normal equation of a quadratic spline (cf. [33]). The following reformulation of (1) follows from Theorem 2.6 in [33].

LEMMA 1. *Suppose that M is a symmetric matrix. Then $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point for* (1) *if and only if x^* satisfies the following piecewise linear equation:*

$$x = (Ex + h)_l^u, \tag{8}$$

where $E := I - \alpha M$, α is any positive constant, and $h := \alpha b$. Moreover, $\frac{1}{\alpha}E(x - (Ex + h)_l^u)$ is the gradient of the following quadratic spline:

$$\begin{aligned}\Psi(x) := {} & \frac{1}{2\alpha}x^T(E - E^2)x - \frac{1}{\alpha}x^T Eh - \frac{\alpha}{2}\|b\|^2 \\ & + \frac{1}{2\alpha}\|(l - (Ex + h))_+\|^2 + \frac{1}{2\alpha}\|((Ex + h) - u)_+\|^2.\end{aligned} \tag{9}$$

As a consequence, for $0 < \alpha\|M\| < 1$, $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point of (1) *if and only if $\Psi'(x^*) = 0$.*

Remark. The reformulation of (6) as a system of piecewise linear equations $x = (Ex + h)_l^u$ is a generalization of Mangasarian's idea of reformulating a linear complementarity problem as a system of piecewise linear equations. Mangasarian's reformulation led to matrix splitting algorithms for solving the linear complementarity problem [38].

Note that we scale the original $\Psi_0(x)$ in (2) by $\frac{1}{\alpha}$ and also add an extra term $-\frac{\alpha}{2}\|b\|^2$. Lemma 1, not including the last statement, follows directly from Theorem 2.6 in [33]. When $0 < \alpha\|M\| < 1$, $E := I - \alpha M$ is a positive definite matrix. Therefore, $\Psi'(x^*) = 0$ if and only if $\frac{1}{\alpha}E(x^* - (Ex^* + h)_l^u) = 0$, which is equivalent to $x^* - (Ex^* + h)_l^u = 0$. From the first statement of Lemma 1 we know that $\Psi'(x^*) = 0$ if and only if $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point of (1).

Now let us consider the following unconstrained minimization problem:

$$\min_{x \in \mathbf{R}^n} \Psi(x), \tag{10}$$

which is equivalent to (2). If M is positive semidefinite and $0 < \alpha\|M\| < 1$, then $\Psi(x)$ is a convex function. As a consequence, x^* is a solution of (1) if and only if x^* is a minimizer of $\Psi(x)$. Therefore, (1) can be reformulated as the unconstrained minimization problem (10) when M is positive semidefinite. However, the reformulation preserves only the first-order optimality conditions. When M is not positive semidefinite, it becomes difficult to relate the second-order optimality conditions for (1) with the second-order optimality conditions for (10). Since (9) does not clearly show how $\Psi(x)$ is related to the original objective function $\frac{1}{2}x^T Mx - b^T x$ and the constraints $l \le x \le u$, we rewrite $\Psi(x)$ by substituting $E = I - \alpha M$ and $h = \alpha b$ into (9). Using simple algebraic manipulations, one can get the following explicit formula for $\Psi(x)$.

LEMMA 2. *For any $\alpha > 0$,*

$$\begin{aligned}\Psi(x) = {} & \left(\frac{1}{2}x^T Mx - b^T x\right) - \frac{\alpha}{2}\|(Mx - b)\|^2 \\ & + \frac{1}{2\alpha}\left\|((l - x) + \alpha(Mx - b))_+\right\|^2 + \frac{1}{2\alpha}\left\|((x - u) - \alpha(Mx - b))_+\right\|^2.\end{aligned}$$

Remark. The previous penalty function can also be derived from Hestenes–Powell–Rockafellar's quadratic augmented Lagrangian function. Consider the following constrained minimization problem:

$$\min_x f(x) \quad \text{subject to} \quad l \le g(x) \le u, \tag{11}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}^m$ are twice continuously differentiable functions. The corresponding augmented Lagrangian function $L(x, y, \alpha)$ introduced independently by Hestenes [26, 27] and Powell [46] for equality constraints and by Rockafellar [47]–[49] for inequality constraints can be written in the following unified way:

$$\begin{aligned} L(x, y, \alpha) := f(x) &+ \frac{\alpha}{2} \left\| \left(\frac{1}{\alpha}(g(x) - u) + y \right)_+ \right\|^2 \\ &+ \frac{\alpha}{2} \left\| \left(\frac{1}{\alpha}(l - g(x)) - y \right)_+ \right\|^2 - \frac{\alpha}{2}\|y\|^2, \end{aligned} \tag{12}$$

where y is the Lagrange multiplier corresponding to two-sided inequality constraints and α is a penalty parameter. If $f(x) = \frac{1}{2}x^T M x - b^T x$ and $g(x) = x$, then the corresponding augmented Lagrangian function has the following expression:

$$\begin{aligned} L(x, y, \alpha) := \frac{1}{2}x^T M x - b^T x &+ \frac{\alpha}{2} \left\| \left(\frac{1}{\alpha}(x - u) + y \right)_+ \right\|^2 \\ &+ \frac{\alpha}{2} \left\| \left(\frac{1}{\alpha}(l - x) - y \right)_+ \right\|^2 - \frac{\alpha}{2}\|y\|^2. \end{aligned} \tag{13}$$

Note that the Lagrange multiplier y in (13) should satisfy the following equation:

$$y = -(Mx - b). \tag{14}$$

As a consequence, we have the following relation between $\Psi(x)$ and the augmented Lagrangian function $L(x, y, \alpha)$:

$$\Psi(x) \equiv L(x, -(Mx - b), \alpha). \tag{15}$$

This is actually Fletcher's idea for obtaining an exact penalty function depending only on x [15]. For example, suppose the gradients

$$\{g_1'(x), g_2'(x), \ldots, g_m'(x)\}$$

of the constraints are linearly independent for any x. Let $g'(x)$ be the $n \times m$ matrix whose ith column is the gradient $g_i'(x)$ of $g_i(x)$. Then the Lagrange multiplier y in (12) should satisfy the following equation:

$$f'(x) + g'(x)y = 0. \tag{16}$$

If (16) has a solution $y(x)$, then $y(x)$ has the following expression:

$$y(x) = -\left((g'(x))^T g'(x)\right)^{-1}(g'(x))^T f'(x),$$

which is Fletcher's multiplier estimate. The function $L(x, y(x), \alpha)$ can be used as an exact penalty function for the two-sided inequality constrained minimization problem

[31]. Fletcher [15] initially used $y(x)$ to eliminate the Lagrange multiplier in the augmented Lagrangian function for equality constrained minimization problems. Our derivation of the penalty function $\Psi(x)$ indicates that Fletcher's multiplier function can also be used for the inequality constrained minimization problem.

Now, one may consider $\Psi(x)$ to be the sum of the objective function

$$\frac{1}{2}x^T Mx - b^T x$$

and a quadratic spline penalty term

$$P(x) := -\frac{\alpha}{2}\|(Mx-b)\|^2 + \frac{1}{2\alpha}\left\|((l-x)+\alpha(Mx-b))_+\right\|^2 + \frac{1}{2\alpha}\left\|((x-u)-\alpha(Mx-b))_+\right\|^2. \tag{17}$$

The penalty function has some important properties that relate $\Psi(x)$ to the objective function of (1).

LEMMA 3. *For any* $\alpha > 0$,

$$\Psi(x) \le \left(\frac{1}{2}x^T Mx - b^T x\right) \quad \text{for } x \in \mathbf{R}^n \text{ with } l \le x \le u. \tag{18}$$

Moreover, the equality in (18) *holds if* $(x, Mx-b)$ *is a Karush–Kuhn–Tucker point of* (1).

Proof. Let $l \le x \le u$. Define $J_1 := \{i : ((l-x)+\alpha(Mx-b))_i > 0\}$. Then, for $i \in J_1$,

$$(\alpha(Mx-b))_i \ge ((l-x)+\alpha(Mx-b))_i > 0$$

and

$$((x-u)-\alpha(Mx-b))_i = (l-u)_i - ((l-x)+\alpha(Mx-b))_i < 0.$$

Thus,

$$\alpha^2(Mx-b)_i^2 \ge (((l-x)+\alpha(Mx-b))_+)_i^2 + (((x-u)-\alpha(Mx-b))_+)_i^2. \tag{19}$$

Similarly, for $i \in J_2 := \{i : ((x-u)-\alpha(Mx-b))_i > 0\}$, (19) holds. The above argument also shows that

$$((l-x)+\alpha(Mx-b))_+^T((x-u)-\alpha(Mx-b))_+ = 0 \tag{20}$$

and $J_1 \cap J_2 = \emptyset$ (i.e., J_1 and J_2 are disjoint). Therefore, by (19),

$$P(x) = -\frac{\alpha}{2}\sum_{i=1}^{n}(Mx-b)_i^2 + \frac{1}{2\alpha}\sum_{i\in J_1}(((l-x)+\alpha(Mx-b))_+)_i^2 + \frac{1}{2\alpha}\sum_{i\in J_2}(((x-u)-\alpha(Mx-b))_+)_i^2 \le -\frac{\alpha}{2}\sum_{i\notin J_1\cup J_2}(Mx-b)_i^2 \le 0.$$

Thus, for $l \le x \le u$,

$$\Psi(x) = \left(\frac{1}{2}x^T Mx - b^T x\right) + P(x) \le \left(\frac{1}{2}x^T Mx - b^T x\right).$$

Now suppose that $(x, Mx - b)$ is a Karush–Kuhn–Tucker point of (1). It follows from (20) that

$$\begin{aligned} &\frac{1}{2\alpha}\left\|((l-x)+\alpha(Mx-b))_+\right\|^2 + \frac{1}{2\alpha}\left\|((x-u)-\alpha(Mx-b))_+\right\|^2 \\ &\quad = \frac{1}{2\alpha}\left\|((l-x)+\alpha(Mx-b))_+ - ((x-u)-\alpha(Mx-b))_+\right\|^2. \end{aligned} \tag{21}$$

One can easily verify that $z_l^u = z + (l-z)_+ - (z-u)_+$ for any $z \in \mathbf{R}^n$. Let $z := x - \alpha(Mx - b) \equiv Ex + h$. Then

$$\begin{aligned} &((l-x)+\alpha(Mx-b))_+ - ((x-u)-\alpha(Mx-b))_+ \\ &\quad = (Ex+h)_l^u - (Ex+h) = \alpha(Mx-b) \end{aligned} \tag{22}$$

because $(Ex+h)_l^u - x = 0$ by Lemma 1. By (21) and (22), $P(x) = 0$ and the equality holds in (18). □

3. Second-order optimality conditions. All the results in this section are attempts to establish relationships between the second-order optimality conditions of (1) and (10).

From the definition of $\Psi(x)$ it is easy to verify that, if $\Psi(\cdot)$ is twice differentiable at x, then the Hessian $\Psi''(x)$ has the following expression:

$$\Psi''(x) = \frac{1}{\alpha}(E - E\Lambda E), \tag{23}$$

where $\Lambda := \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with

$$\lambda_i := \begin{cases} 1 & \text{if } l_i \le (Ex+h)_i \le u_i, \\ 0 & \text{otherwise.} \end{cases}$$

Even when $\Psi(\cdot)$ is not differentiable at x, we still have a similar expression.

THEOREM 4. *Suppose that $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point of (1). Then there exists a positive constant δ such that, for any $x \in \mathbf{R}^n$ with $\|x - x^*\| \le \delta$,*

$$\Psi(x) - \Psi(x^*) = \frac{1}{2\alpha}(x - x^*)^T(E - E\Lambda E)(x - x^*), \tag{24}$$

where $\Lambda := \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with

$$\lambda_i := \begin{cases} 1 & \text{if } l_i < (Ex^*+h)_i < u_i, \\ 1 & \text{if } l_i = (Ex^*+h)_i \text{ and } (E(x-x^*))_i \ge 0, \\ 1 & \text{if } u_i = (Ex^*+h)_i \text{ and } (E(x-x^*))_i \le 0, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Proof. There exists a positive constant δ such that

$$
\begin{aligned}
& l_i < (Ex+h)_i < u_i && \text{if } l_i < (Ex^*+h)_i < u_i, \\
& l_i > (Ex+h)_i && \text{if } l_i > (Ex^*+h)_i, \\
& u_i < (Ex+h)_i && \text{if } u_i < (Ex^*+h)_i
\end{aligned} \tag{26}
$$

whenever $\|x-x^*\| \le \delta$. Let $\|x-x^*\| \le \delta$ and $\Lambda := \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with λ_i given in (25). Then for $x_\theta := \theta x + (1-\theta)x^*$ with $0 \le \theta \le 1$, $g(\theta) := \Psi(x_\theta)$ is a differentiable function of θ and

$$
\begin{aligned}
g'(\theta) &= (x-x^*)^T \Psi'(x_\theta) \\
&= \frac{1}{\alpha}(x-x^*)^T (Ex_\theta - E(Ex_\theta + h)_l^u) \\
&= \frac{1}{\alpha}(x-x^*)^T (Ex_\theta - E\Lambda(Ex_\theta + h) - E\Lambda_c(Ex^* + h)_l^u),
\end{aligned}
$$

where the last equality follows from the definition of Λ. Thus, for $0 \le \theta \le 1$, $\Psi(x_\theta)$ is a quadratic polynomial of θ and

$$
g''(\theta) = \frac{1}{\alpha}(x-x^*)^T (E - E\Lambda E)(x-x^*).
$$

By the Taylor expansion and $\Psi'(x^*) = 0$, we have

$$
\begin{aligned}
\Psi(x) - \Psi(x^*) = g(1) - g(0) &= g'(0) + \frac{1}{2}g''(0) \\
&= (x-x^*)^T \Psi'(x^*) + \frac{1}{2\alpha}(x-x^*)^T (E - E\Lambda E)(x-x^*) \\
&= \frac{1}{2\alpha}(x-x^*)^T (E - E\Lambda E)(x-x^*). \qquad \square
\end{aligned}
$$

Note that the Hessian of the objective function of (1) is M. Therefore, to relate the second-order optimality conditions of (1) to those of (10), we must study how $(E - E\Lambda E)$ is related to M.

Recall that we use Λ to denote a diagonal matrix whose diagonal entries are either 0 or 1, and $\Lambda_c = I - \Lambda$. Since $E = I - \alpha M$, by simple algebraic manipulations, one can verify the following matrix identities.

LEMMA 5. *For any* α,

$$
E - E\Lambda E = \Lambda_c - \alpha \Lambda_c M \Lambda_c + \alpha(\Lambda M \Lambda - \alpha M \Lambda M) \tag{27}
$$

and

$$
\begin{aligned}
E\Lambda M \Lambda E = {} & \Lambda M \Lambda - \alpha M \Lambda M - \alpha(\Lambda M \Lambda)^2 \\
& + \alpha \Lambda_c M \Lambda M \Lambda_c + \alpha^2 (M \Lambda M \Lambda M).
\end{aligned} \tag{28}
$$

To derive the second-order optimality conditions of (10) from those of (1), we give a lower bound estimate of $x^T(E - E\Lambda E)x$. This estimate will allow us to prove that local solutions (or isolated local solutions) of (1) are also local minimizers (or isolated local minimizers) of $\Psi(x)$.

LEMMA 6. *Suppose that* $0 < 2\alpha\|M\| < 1$. *Then there exists a positive constant* β *such that*

$$
x^T(E - E\Lambda E)x \ge \beta(\|\Lambda_c x\|^2 + \|\Lambda M \Lambda x\|^2) + \alpha x^T E \Lambda M \Lambda E x. \tag{29}
$$

Proof. By the definition of the 2-norm of an $n \times n$ matrix Q, for any $y \in \mathbf{R}^n$, we have

$$\|Qy\| \le \|Q\|\|y\|. \tag{30}$$

By (30) and the Cauchy–Schwarz inequality we get

$$y^T Qy \le \|y\|\|Qy\| \le \|Q\|\|y\|^2. \tag{31}$$

Let γ be any positive number. Then

$$\begin{aligned}
\|\Lambda M x\|^2 &= \|\Lambda M \Lambda x + \Lambda M \Lambda_c x\|^2 \\
&\le (\|\Lambda M \Lambda x\| + \|\Lambda M \Lambda_c x\|)^2 \\
&\le (\|\Lambda M \Lambda x\| + \|\Lambda\|\|M \Lambda_c x\|)^2 \\
&\le (\|\Lambda M \Lambda x\| + \|M\|\|\Lambda_c x\|)^2 \\
&= \|\Lambda M \Lambda x\|^2 + \|M\|^2\|\Lambda_c x\|^2 + 2\|M\|\|\Lambda_c x\|\|\Lambda M \Lambda x\| \\
&\le (1+\gamma^{-1})\|\Lambda M \Lambda x\|^2 + \|M\|^2(1+\gamma)\|\Lambda_c x\|^2,
\end{aligned} \tag{32}$$

where the first inequality is by the triangle inequality, the second inequality is from (30), the third inequality follows from (30) and $\|\Lambda\| \le 1$, and the last inequality follows from $2st \le \gamma^{-1}s^2 + \gamma t^2$. By (31) and (32) we find that

$$\begin{aligned}
\alpha^3 x^T M \Lambda M \Lambda M x &\le \alpha^3 \|M\|\|\Lambda M x\|^2 \\
&\le \alpha^3 \|M\| \left((1+\gamma^{-1})\|\Lambda M \Lambda x\|^2 + \|M\|^2(1+\gamma)\|\Lambda_c x\|^2\right).
\end{aligned} \tag{33}$$

Similarly, by (31), (30), and $\|\Lambda\| \le 1$, we can derive that

$$\alpha x^T \Lambda_c M \Lambda_c x \le \alpha \|M\|\|\Lambda_c x\|^2 \tag{34}$$

and

$$\alpha^2 x^T \Lambda_c M \Lambda M \Lambda_c x \le \alpha^2 \|M\|^2 \|\Lambda_c x\|^2. \tag{35}$$

By (27) and (28), we find that

$$\begin{aligned}
x^T(E - E\Lambda E)x - \alpha x^T E \Lambda M \Lambda E x &= x^T \Lambda_c x - \alpha x^T \Lambda_c M \Lambda_c x \\
&+ \alpha^2 x^T (\Lambda M \Lambda)^2 x - \alpha^2 x^T \Lambda_c M \Lambda M \Lambda_c x - \alpha^3 x^T M \Lambda M \Lambda M x.
\end{aligned} \tag{36}$$

Finally, by (36), (33), (34), and (35), we get the following estimate:

$$\begin{aligned}
x^T(E - E\Lambda E)x &\ge (1 - \alpha\|M\| - \alpha^2\|M\|^2 - \alpha^3\|M\|^3(1+\gamma))\|\Lambda_c x\|^2 \\
&+ \alpha^2(1 - \alpha\|M\|(1+\gamma^{-1}))\|\Lambda M \Lambda x\|^2 + \alpha x^T E \Lambda M \Lambda E x.
\end{aligned} \tag{37}$$

Let

$$\gamma := \frac{1}{2}\left(\frac{\alpha\|M\|}{1 - \alpha\|M\|} + \frac{1 - 2\alpha\|M\| + \alpha^4\|M\|^4}{\alpha^3\|M\|^3(1 - \alpha\|M\|)}\right)$$

and

$$\beta := \min\{1 - \alpha\|M\| - \alpha^2\|M\|^2 - \alpha^3\|M\|^3(1+\gamma), \alpha^2(1 - \alpha\|M\|(1+\gamma^{-1}))\}.$$

Since $0 < 2\alpha\|M\| < 1$, one can verify that

$$0 < \frac{\alpha\|M\|}{1-\alpha\|M\|} < \gamma < \frac{1-2\alpha\|M\|+\alpha^4\|M\|^4}{\alpha^3\|M\|^3(1-\alpha\|M\|)}. \tag{38}$$

Since

$$1-\alpha\|M\|-\alpha^2\|M\|^2-\alpha^3\|M\|^3(1+\gamma) = \frac{1-2\alpha\|M\|+\alpha^4\|M\|^4}{1-\alpha\|M\|} - \gamma\alpha^3\|M\|^3,$$

it is easy to verify that (38) implies that $\beta > 0$. Finally, (29) follows from (37). □

4. Equivalence of the quadratic program and its unconstrained reformulation. By using characterizations of local solutions and isolated local solutions of (1), we prove that a local solution (or an isolated local solution) of (1) is also a local minimizer (or an isolated local minimizer) of $\Psi(x)$. By Lemma 3, it is easy to show that the converse also holds. Moreover, x^* is a global solution of (1) if and only if x^* is a global minimizer of $\Psi(x)$.

LEMMA 7 (see [37]). *Let $(x^*, Mx^* - b)$ be a Karush–Kuhn–Tucker point of (1). Then x^* is a local solution of (1) if and only if $\bar{x}^T M\bar{x} \geq 0$ for $\bar{x} \in \mathbf{R}^n$, which satisfies the following conditions:*

$$\begin{aligned} &\bar{x}_i = 0 \quad \text{if } (Ex^* + h)_i < l_i \text{ or } (Ex^* + h)_i > u_i, \\ &\bar{x}_i \geq 0 \quad \text{if } l_i = (Ex^* + h)_i, \\ &\bar{x}_i \leq 0 \quad \text{if } u_i = (Ex^* + h)_i. \end{aligned}$$

LEMMA 8 (see [39]). *Let $(x^*, Mx^* - b)$ be a Karush–Kuhn–Tucker point of (1). Then x^* is an isolated local solution of (1) if and only if $\bar{x}^T M\bar{x} > 0$ for any nonzero vector $\bar{x}$ in $\mathbf{R}^n$, which satisfies the following conditions:*

$$\begin{aligned} &\bar{x}_i = 0 \quad \text{if } (Ex^* + h)_i < l_i \text{ or } (Ex^* + h)_i > u_i, \\ &\bar{x}_i \geq 0 \quad \text{if } l_i = (Ex^* + h)_i, \\ &\bar{x}_i \leq 0 \quad \text{if } u_i = (Ex^* + h)_i. \end{aligned}$$

Remark. The condition given in Lemma 7 is a special case of McCormick's second-order necessary condition for a local minima and the condition given in Lemma 8 is a special case of McCormick's second-order sufficient condition for a local minima [40].

THEOREM 9. *Suppose that $0 < 2\alpha\|M\| < 1$. If x^* is a local solution (or an isolated local solution) of (1), then x^* is a local minimizer (or an isolated local minimizer) of $\Psi(x)$.*

Proof. If x^* is a local solution (or an isolated local solution) of (1), then $(x^*, Mx^* - b)$ satisfies the Karush–Kuhn–Tucker conditions. By Theorem 4, there exist positive constants δ and β such that (24) and (29) hold. Let $x \in \mathbf{R}^n$ with $0 < \|x - x^*\| \leq \delta$.

Note that x^* is always a local solution of (1). (An isolated local solution is a local solution.) By Lemma 7 and the definition of Λ in (25), for $\bar{x} := \Lambda E(x - x^*)$, we have $\bar{x}^T M\bar{x} \geq 0$, i.e., $(x - x^*)^T E\Lambda M\Lambda E(x - x^*) \geq 0$. By (24) and (29),

$$\Psi(x) - \Psi(x^*) \geq \frac{\beta}{2\alpha}(\|\Lambda M\Lambda(x - x^*)\|^2 + \|\Lambda_c(x - x^*)\|^2) \geq 0. \tag{39}$$

Therefore, x^* is a local minimizer of $\Psi(x)$.

Now assume that x^* is an isolated local solution of (1). We want to show that $\Psi(x) - \Psi(x^*) > 0$. If $\Lambda_c(x - x^*) \neq 0$ or $\Lambda M \Lambda(x - x^*) \neq 0$, then it follows from (39) that $\Psi(x) - \Psi(x^*) > 0$. Otherwise,

$$\Lambda E(x - x^*) = \Lambda(I - \alpha M)(x - x^*) = \Lambda(I - \alpha M)\Lambda(x - x^*) = \Lambda(x - x^*).$$

Since $x - x^* \neq 0$ and $\Lambda_c(x - x^*) = 0$, we get $\Lambda E(x - x^*) = \Lambda(x - x^*) \neq 0$. By Lemma 8 and the definition of Λ in (25), for $\bar{x} := \Lambda E(x - x^*)$, we have $\bar{x}^T M \bar{x} > 0$, i.e., $(x - x^*)^T E \Lambda M \Lambda E(x - x^*) > 0$. Therefore, by (24) and (29), $\Psi(x) - \Psi(x^*) > 0$. Hence, x^* is an isolated local minimizer of $\Psi(x)$. □

THEOREM 10. *Suppose that* $0 < \alpha\|M\| < 1$. *If* x^* *is a local minimizer (or an isolated local minimizer) of* $\Psi(x)$, *then* x^* *is a local solution (or an isolated local solution) of* (1).

Proof. Let $f(x) := \frac{1}{2}x^T M x - b^T x$. Since x^* is a local minimizer of $\Psi(x)$, $\Psi'(x^*) = 0$ and there exists a positive constant δ such that

$$\Psi(x) \geq \Psi(x^*) \quad \text{for } x \in \mathbf{R}^n \text{ with } 0 < \|x - x^*\| \leq \delta. \tag{40}$$

By Lemma 1, $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point of (1). Thus, by Lemma 3, $\Psi(x^*) = f(x^*)$. It follows from Lemma 3 that

$$f(x) \geq \Psi(x) \geq \Psi(x^*) = f(x^*)$$

whenever $0 < \|x - x^*\| \leq \delta$ and $l \leq x \leq u$. Therefore, x^* is a local solution of (1).

If x^* is an isolated local minimizer of $\Psi(x)$, then the strict inequality holds in (40) and, as a consequence,

$$f(x) \geq \Psi(x) > \Psi(x^*) = f(x^*)$$

whenever $0 < \|x - x^*\| \leq \delta$ and $l \leq x \leq u$. Therefore, x^* is an isolated local solution of (1). □

Finally, we discuss the equivalence of global solutions of (1) and global minimizers of $\Psi(x)$. First we have the following implication on the existence of global minimizers of $\Psi(x)$.

THEOREM 11. *Suppose that* $0 < 2\alpha\|M\| < 1$. *If* (1) *has a global solution, then* $\Psi(x)$ *is bounded below on* $\mathbf{R}^n$ *and has a global minimizer.*

Proof. Assume the contrary—that $\Psi(x)$ is not bounded below on $\mathbf{R}^n$. Then there exist x^k's such that

$$\lim_{k \to \infty} \Psi(x^k) = -\infty. \tag{41}$$

Define

$$\begin{aligned} J_l^k &:= \{i : (Ex^k + h)_i \leq l_i\}, \\ J_u^k &:= \{i : (Ex^k + h)_i \geq u_i\}, \\ J_0^k &:= \{i : l_i < (Ex^k + h)_i < u_i\}. \end{aligned}$$

Since the index set $\{1, 2, \ldots, n\}$ has finitely many different subsets, we can select a subsequence $\{k_j\}$ such that $J_l^{k_j} \equiv J_l^{k_1}$, $J_u^{k_j} \equiv J_u^{k_1}$, and $J_0^{k_j} \equiv J_0^{k_1}$. Without loss of

generality, we may replace the sequence by the subsequence and assume that there exist three subsets J_l, J_u, and J_0 of $\{1, 2, \ldots, n\}$ such that, for all k,

$$\begin{aligned} J_l &= \{i : (Ex^k + h)_i \le l_i\}, \\ J_u &= \{i : (Ex^k + h)_i \ge u_i\}, \\ J_0 &= \{i : l_i < (Ex^k + h)_i < u_i\}. \end{aligned} \tag{42}$$

Let $\Lambda := \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, where

$$\lambda_i := \begin{cases} 1 & \text{if } i \in J_0, \\ 0 & \text{otherwise.} \end{cases}$$

Since E is nonsingular, there exists $x^0 \in \mathbf{R}^n$ such that

$$\begin{aligned} (Ex^0 + h)_i &= l_i && \text{if } i \in J_l, \\ (Ex^0 + h)_i &= u_i && \text{if } i \in J_u, \\ (Ex^0 + h)_i &= (Ex^1 + h)_i && \text{if } i \in J_0. \end{aligned} \tag{43}$$

As in the proof of Theorem 4, $\Psi(\theta x^k + (1-\theta)x^0)$ is a quadratic function of θ for $0 \le \theta \le 1$ and

$$\Psi(x^k) - \Psi(x^0) = (x^k - x^0)^T \Psi'(x^0) + \frac{1}{2\alpha}(x^k - x^0)^T (E - E\Lambda E)(x^k - x^0). \tag{44}$$

By Lemma 6, there exists a positive constant β such that

$$\begin{aligned} \frac{1}{2\alpha}(x^k - x^0)^T (E - E\Lambda E)(x^k - x^0) &\ge \frac{\beta}{2\alpha}\|\Lambda_c(x^k - x^0)\|^2 \\ &\quad + \frac{\beta}{2\alpha}\|\Lambda M \Lambda (x^k - x^0)\|^2 + \frac{1}{2}(x^k - x^0)^T E\Lambda M \Lambda E (x^k - x^0). \end{aligned} \tag{45}$$

In the remaining part of the proof, we want to show that

$$(x^k - x^0)^T \Psi'(x^0) + \frac{1}{2}(x^k - x^0)^T E\Lambda M \Lambda E (x^k - x^0)$$

is bounded below by a linear function of $(\Lambda_c(x^k - x^0))$ and $(\Lambda M \Lambda (x^k - x^0))$. As a consequence, $\Psi(x^k)$ is bounded below by a strictly convex quadratic function of $(\Lambda_c(x^k - x^0))$ and $(\Lambda M \Lambda (x^k - x^0))$.

Since $l \le (Ex^0 + h) \le u$ and $l \le (Ex^0 + h) + \Lambda E(x^k - x^0) \le u$, we have

$$\alpha \Psi'(x^0) = E(x^0 - (Ex^0 + h)_l^u) = E(x^0 - (Ex^0 + h)) = \alpha E(Mx^0 - b) \tag{46}$$

and

$$\begin{aligned} -\infty < \kappa &\le f((Ex^0 + h) + \Lambda E(x^k - x^0)) - f(Ex^0 + h) \\ &= (x^k - x^0)^T E\Lambda (M(Ex^0 + h) - b) \\ &\quad + \frac{1}{2}(x^k - x^0)^T E\Lambda M \Lambda E (x^k - x^0), \end{aligned} \tag{47}$$

where $f(x) := \frac{1}{2}x^T Mx - b^T x$, $\kappa := \inf_{l \le x \le u} f(x) - f(Ex^0 + h)$, and the equality in (47) is the Taylor expansion for $f(x)$. By (46) and

$$M(Ex^0 + h) - b = E(Mx^0 - b)$$

we get

$$\Psi'(x^0) - E\Lambda(M(Ex^0 + h) - b) = (E - E\Lambda E)(Mx^0 - b). \tag{48}$$

By simple algebraic manipulations, one can derive from (27) that

$$E - E\Lambda E = \Lambda_c(E\Lambda_c - \alpha^2 M\Lambda M) + \alpha\Lambda M\Lambda E. \tag{49}$$

It follows from (48) and (49) that

$$\begin{aligned}(x^k - x^0)^T\Psi'(x^0) - (x^k - x^0)^T E\Lambda(M(Ex^0 + h) - b)\\ = p^T\Lambda_c(x^k - x^0) + q^T\Lambda M\Lambda(x^k - x^0),\end{aligned} \tag{50}$$

where $p := (E\Lambda_c - \alpha^2 M\Lambda M)(Mx^0 - b)$ and $q := \alpha E(Mx^0 - b)$. From (47) and (50) we obtain

$$\begin{aligned}(x^k - x^0)^T\Psi'(x^0) + \frac{1}{2}(x^k - x^0)^T E\Lambda M\Lambda E(x^k - x^0)\\ \geq p^T\Lambda_c(x^k - x^0) + q^T\Lambda M\Lambda(x^k - x^0) + \kappa.\end{aligned} \tag{51}$$

Finally, by (44), (45), and (51), we find that

$$\Psi(x^k) - \Psi(x^0) \geq h(\Lambda_c(x - x^0), \Lambda M\Lambda(x - x^0)),$$

where

$$h(y, z) := \frac{\beta}{2\alpha}(\|y\|^2 + \|z\|^2) + p^T y + q^T z + \kappa.$$

Since $h(y, z)$ is a strictly convex quadratic function of (y, z), there exists a constant η such that $h(y, z) \geq \eta$ for all $y, z \in \mathbf{R}^n$. As a consequence, $\Psi(x^k) \geq \Psi(x^0) + \eta > -\infty$. The contradiction proves that $\Psi(x)$ is bounded below on $\mathbf{R}^n$.

By the Frank–Wolfe theorem [17, 10], we see that if a piecewise quadratic function is bounded below on $\mathbf{R}^n$, it has a global minimizer. Therefore, $\Psi(x)$ has a global minimizer. □

THEOREM 12. *Let* $0 < 2\alpha\|M\| < 1$. *Then* x^* *is a global solution of* (1) *if and only if* x^* *is a global minimizer of* $\Psi(x)$.

Proof. Assume that x^* is a global solution of (1). Then $(x^*, Mx^* - b)$ is a Karush–Kuhn–Tucker point of (1) and, by Lemma 3, $\Psi(x^*) = f(x^*)$, where $f(x) := \frac{1}{2}x^T Mx - b^T x$. By Theorem 11, $\Psi(x)$ also has a global minimizer $\hat{x}$. Since $\Psi'(\hat{x}) = 0$, by Lemmas 1 and 3, $(\hat{x}, M\hat{x} - b)$ is a Karush–Kuhn–Tucker point of (1) and

$$\Psi(\hat{x}) = f(\hat{x}) \geq f(x^*) = \Psi(x^*).$$

Thus, x^* is a global minimizer of $\Psi(x)$.

Now suppose that x^* is a global minimizer of $\Psi(x)$. By Lemma 3, $f(x^*) = \Psi(x^*)$ and

$$f(x) \geq \Psi(x) \geq \Psi(x^*) = f(x^*) \quad \text{for } x \in \mathbf{R}^n \text{ with } l \leq x \leq u.$$

Therefore, x^* is also a global solution of (1). □

5. Conclusions. We were able to show that, for $0 < 2\alpha\|M\| < 1$, the quadratic program (1) is "absolutely" equivalent to the unconstrained minimization of the quadratic spline $\Psi(x)$ in every conceivable way: the two problems have the same set of local solutions, the same set of isolated local solutions, and the same set of global solutions. Moreover, the exact penalty function $\Psi(x)$ is convex if (1) is a convex quadratic program. Majthay and Mangasarian's characterizations of local solutions [37] and isolated local solutions [39], respectively, of (1) were essential in our analysis.

Since M is symmetric, $\|M\|_1 = \|M\|_\infty$ and we have the following upper bound of $\|M\|$:

$$\|M\| \le \sqrt{\|M\|_1\|M\|_\infty} = \max_{1\le i\le n} \sum_{j=1}^{n} |m_{ij}|,$$

where m_{ij} is the (i,j) entry of M. See §2.2 of [19] for details. Therefore, $0 < 2\alpha\|M\| < 1$ if

$$0 < \alpha < \left(2 \cdot \max_{1\le i\le n} \sum_{j=1}^{n} |m_{ij}| \right)^{-1}.$$

This provides a simple and explicit estimate for the threshold of the penalty parameter α.

From the discussion in §2, we realize that the exact penalty function $\Psi(x)$ could actually be derived from Hestenes–Powell–Rockafellar's augmented Lagrangian function for two-sided inequality constrained problems by using Fletcher's multiplier function. The process indicates that we may obtain an exact penalty function by eliminating the multiplier y from the augmented Lagrangian function simply by using the equations in the Karush–Kuhn–Tucker conditions for the constrained minimization problem. At least the approach works for the quadratic program with simple bound constraints. Further results for general nonlinear programming problems can be found in [31].

Algorithms based on the exact penalty function $\Psi(x)$ to solve (1) were discussed in [32, 33, 29, 30] when M is positive semidefinite. It would be interesting to study how the exact penalty function $\Psi(x)$ can be used to design new numerical algorithms for solving nonconvex quadratic programs with simple bound constraints.

Acknowledgment. The author would like to thank O. L. Mangasarian for providing references [37, 39] on characterizations of local solutions and isolated local solutions of a quadratic program.

REFERENCES

[1] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[2] ———, *Projected Newton methods for optimization methods for optimization problems with simple bounds*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[3] A. Björck, *A direct method for sparse least squares problems with lower and upper bounds*, Numer. Math., 54 (1988), pp. 19–32.

[4] T. F. Coleman and L. A. Hulbert, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Math. Programming, 45 (1990), pp. 180–201.

[5] ———, *A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds*, SIAM J. Optim., 3 (1993), pp. 298–321.

[6] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[7] ———, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comput., 50 (1988), pp. 399–430.

[8] R. W. Cottle and M. S. Goheen, *A special class of large quadratic programs*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1978.

[9] R. W. Cottle, G. H. Golub, and R. S. Sacher, *On the solution of large, structured linear complementarity problems: The block partitioned case*, Appl. Math. Optim., 4 (1978), pp. 347–363.

[10] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[11] A. Dax, *On computational aspects of bounded linear least squares problems*, ACM Trans. Math. Software, 17 (1991), pp. 64–73.

[12] M. A. Diamond, *The solution of a quadratic programming problem using fast methods to solve systems of linear equations*, Internat. J. Systems Sci., 5 (1974), pp. 131–136.

[13] G. Di Pillo and L. Grippo, *Exact penalty functions in constrained optimization*, SIAM J. Control Optim., 27 (1989), pp. 1333–1360.

[14] F. Facchinei and S. Lucidi, *A class of penalty functions for optimization problems with bound constraints*, Optimization, 26 (1992), pp. 239–259.

[15] R. Fletcher, *A class of methods for nonlinear programming with termination and convergence properties*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1970, pp. 157–173.

[16] R. Fletcher and M. P. Jackson, *Minimization of a quadratic function of many variables subject to only lower and upper bounds*, J. IMA, 14 (1974), pp. 159–174.

[17] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logist., 3 (1956), pp. 95–110.

[18] A. Friedlander and J. M. Martínez, *On the maximization of a concave quadratic function with box constraints*, SIAM J. Optim., 4 (1994), pp. 177–192.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, MD, 1983.

[20] L. Grippo and S. Lucidi, *A differentiable exact penalty function for bound constrained quadratic programming problems*, Optimization, 22 (1991), pp. 557–578.

[21] ———, *On the solution of a class of quadratic programs using a differentiable exact penalty function*, in System Modelling and Optimization (Leipzig, 1989), H. J. Sebastian and K. Tammer, eds., Lecture Notes in Control and Inform. Sci., 143, Springer-Verlag, Berlin, 1990, pp. 764–773.

[22] C.-H. Han, P. M. Pardalos, and Y. Ye, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, in Large-scale Numerical Optimization, Thomas F. Coleman and Yuying Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 92–112.

[23] ———, *An interior point algorithm for large-scale quadratic problems with box constraints*, Springer-Verlag Lecture Notes in Control and Information, Vol. 144, A. Bensoussan and J. L. Lions, eds., 1990, pp. 413–422.

[24] ———, *On the solution of indefinite quadratic problems using an interior point algorithm*, Informatica, 3 (1992), pp. 474–496.

[25] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Programming, 17 (1979), pp. 251–269.

[26] M. R. Hestenes, *Multiplier and gradient methods*, in Computing Methods in Optimization Problems—2, L. A. Zadeh, L. W. Neustadt, and A. V. Balakrishnan, eds., Academic Press, New York, 1969, pp. 143–164.

[27] ———, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.

[28] J. J. Júdice and F. M. Pires, *Bard-type methods for the linear complementarity problem with symmetric positive definite matrices*, IMA J. Math. Appl. Bus. Indust., 2 (1989), pp. 51–68.

[29] W. Li, *Linearly convergent descent methods for unconstrained minimization of a convex quadratic spline*, J. Optim. Theory Appl., 86 (1995), pp. 145–172.

[30] ———, *A conjugate gradient method for the unconstrained minimization of strictly convex quadratic splines*, Math. Programming, to appear.

[31] ———, *Differentiable piecewise quadratic penalty functions for quadratic programs via Hestenes–Powell–Rockafellar's augmented Lagrangian function*, Department of Math-

ematics and Statistics, Old Dominion University, Norfolk, VA, April 1994, revised July 1995, manuscript.

[32] W. Li and J. Swetits, *A Newton method for convex regression, data smoothing, and quadratic programming with bounded constraints*, SIAM J. Optim., 3 (1993), pp. 466–488.

[33] ———, *A new algorithm for solving strictly convex quadratic programs*, Tech. report TR92-1, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, Nov. 1992, SIAM J. Optim., submitted.

[34] Y. Lin and C. W. Cryer, *An alternating direction implicit algorithm for the solution of linear complementarity problems arising from free boundary problems*, Appl. Math. Optim., 13 (1985), pp. 1–17.

[35] Y. Y. Lin and J.-S. Pang, *Iterative methods for large convex quadratic programs: A survey*, SIAM J. Control Optim., 25 (1987), pp. 383–411.

[36] P. Lotstedt, *Solving the minimal least squares problem subject to bounds on the variables*, BIT, 24 (1984), pp. 206–224.

[37] A. Majthay, *Optimality conditions for quadratic programming*, Math. Programming, 1 (1971), pp. 359–365.

[38] O. L. Mangasarian, *Solution of symmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl., 22 (1977), pp. 465–485.

[39] ———, *Locally unique solutions of quadratic programs, linear and nonlinear complementarity problems*, Math. Programming, 19 (1980), pp. 200–212.

[40] G. P. McCormick, *Second order conditions for constrained minima*, SIAM J. Appl. Math., 15 (1967), pp. 641–652.

[41] J. J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[42] ———, *On the solutions of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[43] D. P. O'Leary, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[44] P. M. Pardalos, C. G. Han, and Y. Ye, *Interior-point algorithms for solving nonlinear optimization problems*, Math. Progr. Society, COAL Newsletter, 19 (1991), pp. 45–54.

[45] ———, *An interior-point algorithm for large-scale quadratic problems with box constraints*, in Analysis and Optimization of Systems, A. Bensoussan and J. L. Lions, eds., Lecture Notes in Control and Inform. Sci., 144, Springer-Verlag, New York, 1990, pp. 413–422.

[46] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1972, pp. 283–298.

[47] R. T. Rockafellar, *New applications of duality in convex programming*, in Proc. Confer. Probab. 4th, Brasov, Romania, 1971, pp. 73–81.

[48] ———, *The multiplier method of Hestenes and Powell applied to convex programming*, J. Optim. Theory Appl., 12 (1973), pp. 555–562.

[49] ———, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Math. Programming, 5 (1973), pp. 354–573.

[50] A. Stachurski, *Monotone sequences of feasible solutions for quadratic programming problems with M-matrices and box constraints*, in System Modelling and Optimization (Budapest, 1985), A. Prekopa, J. Szelezsan, and B. Strazicky, eds., Lecture Notes in Control and Inform. Sci., 84, Springer-Verlag, New York, 1985, pp. 896–902.

[51] P. Tseng, D. P. Bertsekas, and J. N. Tsitsiklis, *Partially asynchronous, parallel algorithms for network flow and other problems*, SIAM J. Control Optim., 28 (1990), pp. 678–710.

[52] E. K. Yang and J. W. Tolle, *A class of methods for solving large, convex quadratic programs subject to box constraints*, Math. Programming, 51 (1991), pp. 223–228.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 316–325, May 1996
© 1996 Society for Industrial and Applied Mathematics
004

CONVERGENCE OF AN INFEASIBLE INTERIOR-POINT ALGORITHM FROM ARBITRARY POSITIVE STARTING POINTS*

STEPHEN C. BILLUPS† AND MICHAEL C. FERRIS†

Abstract. An important advantage of infeasible interior-point methods over feasible interior-point methods is their ability to be warm-started from approximate solutions. It is therefore important for the convergence theory of these algorithms not to depend on the ability to alter the starting point. In a recent paper [*SIAM J. Optim.*, 4 (1994), pp. 208–227], Zhang proves a global linear convergence rate for an infeasible interior-point method for the horizontal linear complementarity problem, which unfortunately places a restriction on the starting point. It is easy to meet the restriction by altering the starting point, but this may take the point farther away from the solution, thus removing the advantage of warm-starting the algorithm. In this paper we extend Zhang's results to apply to arbitrary strictly positive starting points. We also show how the extended results can be used to prove convergence for an algorithm for box-constrained linear complementarity problems.

Key words. infeasible interior-point methods, linear complementarity, global convergence

AMS subject classification. 90C33

1. Introduction.

Quite often, in using an iterative method to solve a problem, it is possible to use a previously derived approximate solution as a starting point. Such a starting point may be available, for example, as the result of solving a "nearby" problem. When such an approximate solution is used as a starting point, we say that the algorithm has been *warm-started.* Warm-starting is particularly important in the context of sequential quadratic programming where a nonlinear problem is tackled by solving a sequence of quadratic subproblems. Typically, the solution of each subproblem provides an excellent starting point for the next subproblem in the sequence. It is therefore important to consider how well an algorithm can be warm-started.

Until recently, interior-point methods were not amenable to warm-starting because they required the starting point to be strictly feasible. This was an unfortunate constraint since a solution to a nearby problem would not, in general, be feasible. Thus, it would have to be modified to bring it into the feasible region—a process which would typically carry it farther away from the solution.

More recently, infeasible interior-point methods which differ from (feasible) interior-point methods have been developed. Specifically, they allow the iterates to violate the equality constraints while strictly satisfying the nonnegativity constraints. The relaxation of the feasibility requirements allows these infeasible interior-point methods to handle warm starts quite effectively. Typically, the problems can be formulated in such a way that the only difference between the current problem and the nearby problem is in the equality constraints. Thus, with only a slight modification, the solution to the nearby problem can be used as the starting point for the current problem. This slight modification is needed simply to move the point off the boundary of the region defined by the nonnegativity constraints and can be arbitrarily small.

A significant amount of work has been devoted to the development of infeasible interior-point algorithms. This line of research first produced practical algorithms

* Received by the editors October 15, 1993; accepted for publication (in revised form) December 6, 1994. This material is based on research supported by National Science Foundation grant CCR-9157632 and Air Force Office of Scientific Research grant F49620-94-1-0036.

† Computer Sciences Department, University of Wisconsin, Madison, WI 53706.

along with numerical tests and comparisons, which demonstrated the superior practical performance of this class of algorithms. See, for example, [1, 2, 9, 10, 11, 12, 13, 14, 16, 18].

Recently, a number of theoretical papers have been written which analyze convergence and the complexity behavior of these algorithms. See, for example, [8, 20, 22, 23, 24]. Of particular importance is the paper by Zhang [23], where global Q-linear convergence and polynomial complexity is demonstrated for a class of infeasible interior-point methods for a generalization of the linear complementarity problem (LCP) called the horizontal linear complementarity problem (HLCP). This work is significant because the class of algorithms Zhang analyzes is closely related to already existing algorithms with proven effectiveness.

Unfortunately, Zhang's analysis places a restriction on the starting point, which poses problems for warm-starting. The restriction is very easy to satisfy if we are allowed to vary the starting point. However, this completely defeats the purpose of warm-starting, since changing the starting point may take us farther away from the solution. Fortunately, the restriction on the starting point is unnecessary. Demonstrating that fact is the purpose of this paper.

Since Zhang's algorithm was first introduced, a number of algorithms have been proposed based on Zhang's work. In [22], Wright proposes an infeasible interior-point method for the monotone LCP, which has global Q-linear convergence and local Q-subquadratic convergence. Wright's analysis of this algorithm depends on the same restriction on the starting point that was used by Zhang. This restriction can be removed from Wright's convergence results as well, using the results in this paper, as is shown in [3].

The issue of relaxing the restrictions on the starting points for infeasible interior-point algorithms has been addressed in several other papers. For linear programming, global convergence from any positive starting point is proved by Potra in [17]. A similar result is given in [19] for the LCP. In [15], Monteiro and Wright propose an algorithm for the LCP that achieves superlinear convergence from any positive starting point. Their analysis is significant because they prove superlinear convergence without assuming the existence of a strictly feasible point. Another very recent algorithm by Wright [21] achieves similar convergence results for the mixed monotone LCP.

In [7], Güler proves that any interior-point algorithm for the LCP can be extended to an interior-point algorithm for the HLCP with identical convergence results. Thus, in principle, any of the methods cited above could be extended to the HLCP to provide an algorithm that converges from an arbitrary positive starting point. However, because of the fundamental importance of Zhang's algorithm, a direct generalization of his analysis is appropriate. This is the approach we take in this paper.

Some words about notation are in order. Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm of a vector. Iteration numbers appear as superscripts on vectors and matrices and as subscripts on scalars. A subscript on a vector (or matrix) represents either a subvector (submatrix) or a component of the vector (matrix). In expressions, vectors are assumed to be column vectors unless explicitly transposed. We use the notation $(\cdot)_+$, $(\cdot)_-$, and $|\cdot|$ to represent the plus, minus, and absolute value operators, respectively, for vectors, that is, $x_+ := (\max(x_1, 0), \ldots, \max(x_n, 0))$, $x_- := (\max(-x_1, 0), \ldots, \max(-x_n, 0))$, and $|x| := (|x_1|, \ldots, |x_n|)$. We also refer to the sets $\mathbf{R}_+$ and $\mathbf{R}_{++}$, which represent the nonnegative real numbers and the positive real numbers, respectively.

2. Zhang's algorithm for the HLCP. The HLCP [4] can be stated as follows:

$$\text{HCLP} \qquad \begin{aligned} &\text{Find } (x,y) \in \mathbf{R}^n \times \mathbf{R}^n \text{ such that} \\ &F(x,y) := \begin{bmatrix} Mx+Ny-h \\ XYe \end{bmatrix} = 0, \quad (x,y) \geq 0, \end{aligned}$$

where $M, N \in \mathbf{R}^{n\times n}, e, h \in \mathbf{R}^n, X = \operatorname{diag}(x), Y = \operatorname{diag}(y), e = (1,1,\ldots,1)^\top$. Note that if $N = -I$, this is exactly the LCP.

In his paper [23], Zhang presents two algorithms for solving the HLCP. The first algorithm is a very general algorithm about which a number of useful lemmas can be proved. The second algorithm is a special case of the first for which Zhang proves a global Q-linear convergence result.

For convenience of discussion, Zhang defines the following sets:

$\mathcal{S} = \{(x,y) \in \mathbf{R}^{2n} : h = Mx+Ny, (x,y) \geq 0, x^\top y = 0\}$, i.e., the solution set,
$\mathcal{A} = \{(x,y) \in \mathbf{R}^{2n} : h = Mx+Ny\}$,
$\mathcal{F} = \{(x,y) \in \mathcal{A} : (x,y) \geq 0\}$, i.e., the set of feasible points.

Zhang's algorithms can be described as centered and damped Newton methods that work as follows. Given a starting point $(x^0,y^0) > 0$, both algorithms generate a sequence of strictly positive iterates $\{(x^k,y^k)\}$ that, under appropriate assumptions, converge to a solution (x^*,y^*) of the HLCP.

To prove his results, Zhang makes the following assumptions on the problem.

ASSUMPTION 2.1. *For any $(x,y) \in \mathcal{A}$ and $(\hat{x},\hat{y}) \in \mathcal{A}, (x-\hat{x})^\top(y-\hat{y}) \geq 0$, i.e., $\mathcal{A}$ is the graph of a monotone operator.*

ASSUMPTION 2.2. *$\mathcal{F} \neq \emptyset$, i.e., a feasible point exists.*

It is known that Assumptions 2.1–2.2 imply the existence of a solution (x^*,y^*) to the HLCP (see [7, Thm. 3.1]). It is also well known that Assumption 2.1 is satisfied by linear programs, convex quadratic programs, and monotone LCPs.

In addition to these two explicit assumptions, Zhang also makes an implicit assumption about the starting point. Given a point $(u^0,v^0) \in \mathcal{A}$ (such a point exists by Assumption 2.2), Zhang proves his convergence results by choosing a starting point $(x^0,y^0) > 0$ that satisfies $(x^0,y^0) \geq (u^0,v^0)$. It is easy to find such an (x^0,y^0) (simply choose $x^0 = \max(\zeta,u^0)$ and $y^0 = \max(\zeta,v^0)$ for some $\zeta > 0$). However, since we are interested in warm-starting the algorithm, we do not want to change the starting point. Thus, given a *fixed* starting point (x^0,y^0), Zhang's results are based on the following implicit assumption.

ASSUMPTION 2.3. *There exists $(u^0,v^0) \in \mathcal{A}$ such that $(x^0,y^0) \geq (u^0,v^0)$.*

We now prove Zhang's results without this implicit assumption. We start with Zhang's first algorithm.

Algorithm 1– Zhang's first algorithm

Given $(x^0,y^0) > 0$, for $k = 0,1,2,\ldots,$ do

1. Choose $\sigma_k \in [0,1)$ and let $\mu_k = \frac{1}{n}{x^k}^\top y^k$. Solve the following linear system for $(\Delta x^k, \Delta y^k)$:

$$\begin{bmatrix} M & N \\ Y & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} h - Mx^k - Ny^k \\ -X^kY^k + \sigma_k\mu_k e \end{bmatrix}. \tag{1}$$

2. Choose a steplength $\alpha_k \in (0,1]$ and

$$\alpha_k < \hat{\alpha}_k = \frac{-1}{\min((X^k)^{-1}\Delta x^k, (Y^k)^{-1}\Delta y^k, -1/2)}.$$

Let $x^{k+1} = x^k + \alpha_k \Delta x^k$ and $y^{k+1} = y^k + \alpha_k \Delta y^k$.

We shall prove a number of technical lemmas about Algorithm 1, which we will then use to prove Q-linear convergence of Zhang's second algorithm, which is a special case of Algorithm 1.

We begin as Zhang does by constructing an auxiliary sequence $\{(u^k, v^k)\}$. Given a pair $(u^0, v^0) \in \mathcal{A}$, for $k = 0, 1, \ldots$ we define

$$u^{k+1} := u^k + \alpha_k(\Delta x^k + x^k - u^k), \quad v^{k+1} := v^k + \alpha_k(\Delta y^k + y^k - v^k), \tag{2}$$

where x^k, y^k, Δx^k, and Δy^k are defined in Algorithm 1. The sequence is strictly a tool for analysis and is not actually computed. The following lemma summarizes some of the properties of the auxiliary sequence.

LEMMA 2.4. *Let* $\{(x^k, y^k)\}$ *and* $\{\alpha_k\}$ *be generated by Algorithm* 1, $\{(u^k, v^k)\}$ *be given by* (2), *and* $\nu_k := \prod_{j=0}^{k-1}(1 - \alpha_j)$. *Then for* $k \geq 0$,

1. $(u^k, v^k) \in \mathcal{A}$, *i.e.*, $h = Mu^k + Nv^k$;
2. $x^k - u^k = \nu_k(x^0 - u^0)$ *and* $y^k - v^k = \nu_k(y^0 - v^0)$;
3. $|x^k - u^k| = \nu_k|x^0 - u^0| \leq |x^0 - u^0|$ *and* $|y^k - v^k| = \nu_k|y^0 - v^0| \leq |y^0 - v^0|$.

Proof. Statements 1 and 2 are proven in [23, Lem. 4.1]. Statement 3 follows immediately from Statement 2 and the fact that $0 \leq \nu_k < 1$. □

LEMMA 2.5. *Let* $(x^0, y^0) \in \mathbf{R}^n_{++} \times \mathbf{R}^n_{++}, (u^0, v^0) \in \mathcal{A}, (\hat{x}, \hat{y}) \in \mathcal{F}$, *and let* $\{(x^k, y^k)\}$ *be generated by Algorithm* 1. *Then, under Assumptions* 2.1–2.2, *for all* $k \geq 0$,

$$\begin{aligned}
&(\hat{x} - (x^0 - u^0)_-)^\top y^k + (\hat{y} - (y^0 - v^0)_-)^\top x^k \\
&\leq \hat{x}^\top \hat{y} + {x^k}^\top y^k + \nu_k \left(|x^0 - u^0|^\top \hat{y} + |y^0 - v^0|^\top \hat{x} + \nu_k |x^0 - u^0|^\top |y^0 - v^0| \right).
\end{aligned}$$

Proof. Define $\{(u^k, v^k)\}$ according to (2). Then, by Lemma 2.4(1), $(u^k, v^k) \in \mathcal{A}$, so $(\hat{x} - u^k)^\top (\hat{y} - v^k) \geq 0$ by Assumption 2.1.

Using this fact,

$$\begin{aligned}
&\hat{x}^\top y^k + \hat{y}^\top x^k + (x^k - u^k)^\top y^k + (y^k - v^k)^\top x^k \\
&\leq \hat{x}^\top y^k + \hat{y}^\top x^k + (x^k - u^k)^\top y^k + (y^k - v^k)^\top x^k + (\hat{x} - u^k)^\top (\hat{y} - v^k) \\
&= \hat{x}^\top \hat{y} + {x^k}^\top y^k + (x^k - u^k)^\top \hat{y} + (y^k - v^k)^\top \hat{x} + (x^k - u^k)^\top (y^k - v^k).
\end{aligned} \tag{3}$$

Thus,

$$\begin{aligned}
&(\hat{x} - (x^0 - u^0)_-)^\top y^k + (\hat{y} - (y^0 - v^0)_-)^\top x^k \\
&= \hat{x}^\top y^k + \hat{y}^\top x^k - (x^0 - u^0)_-^\top y^k - (y^0 - v^0)_-^\top x^k \\
&\leq \hat{x}^\top y^k + \hat{y}^\top x^k - (x^k - u^k)_-^\top y^k - (y^k - v^k)_-^\top x^k \quad \text{(by Lemma 2.4(2))} \\
&\leq \hat{x}^\top y^k + \hat{y}^\top x^k + (x^k - u^k)^\top y^k + (y^k - v^k)^\top x^k \\
&\leq \hat{x}^\top \hat{y} + {x^k}^\top y^k + (x^k - u^k)^\top \hat{y} + (y^k - v^k)^\top \hat{x} + (x^k - u^k)^\top (y^k - v^k) \quad \text{(by (3))} \\
&\leq \hat{x}^\top \hat{y} + {x^k}^\top y^k + |x^k - u^k|^\top \hat{y} + |y^k - v^k|^\top \hat{x} + |x^k - u^k|^\top |y^k - v^k| \\
&\qquad \text{(since } (\hat{x}, \hat{y}) \geq 0) \\
&= \hat{x}^\top \hat{y} + {x^k}^\top y^k + \nu_k(|x^0 - u^0|^\top \hat{y} + |y^0 - v^0|^\top \hat{x} + \nu_k |x^0 - u^0|^\top |y^0 - v^0|) \\
&\qquad \text{(by Lemma 2.4(3)).} \qquad \square
\end{aligned}$$

LEMMA 2.6. *Let $\{(x^k, y^k)\}$ be generated by Algorithm 1 in such a way that $\phi_0 \geq {x^k}^\top y^k$ for some $\phi_0 > 0$ and let (x^*, y^*) be a solution to the HLCP. If for some i, $x_i^* > 0$, then the sequence $\{y_i^k\}$ is bounded. Similarly, if $y_i^* > 0$, then $\{x_i^k\}$ is bounded.*

Proof. Define $\tilde{x} := x^* - (x^0 - x^*)_-$ and $\tilde{y} := y^* - (y^0 - y^*)_-$. By applying Lemma 2.5 with $(\hat{x}, \hat{y}) = (u^0, v^0) = (x^*, y^*)$ and noting that ${x^*}^\top y^* = 0$, we get

$$\begin{aligned}
&\tilde{x}^\top y^k + \tilde{y}^\top x^k \\
&\leq {x^k}^\top y^k + \nu_k(|x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + \nu_k |x^0 - x^*|^\top |y^0 - y^*|) \\
&\leq \phi_0 + |x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + |x^0 - x^*||y^0 - y^*| =: C.
\end{aligned}$$

Thus,

$$\sum_{i=1}^{n} (\tilde{x}_i y_i^k + x_i^k \tilde{y}_i) \leq C. \tag{4}$$

Now, $\tilde{y}_i = \min(y_i^0, y_i^*) \geq 0$ and $\tilde{x}_i = \min(x_i^0, x_i^*) \geq 0$. So each term on the left side of (4) is nonnegative. Therefore, for all i,

$$x_i^k \tilde{y}_i \leq C \quad \text{and} \quad \tilde{x}_i y_i^k \leq C.$$

Thus, we see that if $\tilde{y}_i > 0$, then $\{x_i^k\}$ is bounded. Similarly, if $\tilde{x}_i > 0$, then $\{y_i^k\}$ is bounded. □

The next lemma is the counterpart to [23, Lem. 6.1].

LEMMA 2.7. *Let $\{(x^k, y^k)\}$ be generated by Algorithm 1 in such a way that $\phi_0 \geq {x^k}^\top y^k \geq \beta \nu_k$ for some $\phi_0, \beta > 0$. For any $(x^*, y^*) \in \mathcal{S}$, let $(u^0, v^0) := (x^*, y^*)$ and generate $\{(u^k, v^k)\}$ according to (2). Then there exists $K > 0$ such that*

$$\frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k} \leq K.$$

Proof. Partition the indices $\{1, \ldots, n\}$ as follows: $H_1 := \{i : x_i^0 \geq x_i^*, y_i^0 \geq y_i^*\}, H_2 := \{i : x_i^0 < x_i^*\}, H_3 := \{i : y_i^0 < y_i^*\}$. Note that $H_2 \bigcap H_3 = \emptyset$ since one of x_i^* and y_i^* is zero for each i and (x^0, y^0) is strictly positive.

By Lemma 2.4(2), $x_i^k - u_i^k = \nu_k(x_i^0 - x_i^*)$ and $y_i^k - v_i^k = \nu_k(y_i^0 - y_i^*)$, so

$$|x_i^k - u_i^k| = \begin{cases} (x_i^k - u_i^k) & \text{for } i \in H_1 \bigcup H_3, \\ (u_i^k - x_i^k) & \text{for } i \in H_2, \end{cases}$$
$$|y_i^k - v_i^k| = \begin{cases} (y_i^k - v_i^k) & \text{for } i \in H_1 \bigcup H_2, \\ (v_i^k - y_i^k) & \text{for } i \in H_3. \end{cases}$$

By Lemma 2.6, $i \in H_2 \Rightarrow x_i^* > 0 \Rightarrow \{y_i^k\}$ is bounded $\Rightarrow \{v_i^k\}$ is bounded. Similarly, for $i \in H_3$ $\{x_i^k\}$ and $\{u_i^k\}$ are bounded. Thus, there exists $K_1 > 0$ such that $|y_i^k + v_i^k| < K_1$ for $i \in H_2$ and $|x_i^k + u_i^k| < K_1$ for $i \in H_3$. Thus,

$$\begin{aligned}
&|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k \\
&= \sum_{i \in H_1} \{(x_i^k - u_i^k) y_i^k + (y_i^k - v_i^k) x_i^k\} + \sum_{i \in H_2} \{(u_i^k - x_i^k) y_i^k + (y_i^k - v_i^k) x_i^k\} \\
&\quad + \sum_{i \in H_3} \{(x_i^k - u_i^k) y_i^k + (v_i^k - y_i^k) x_i^k\}
\end{aligned}$$

$$
\begin{aligned}
(5) \quad &= {x^k}^\top y^k - {u^k}^\top v^k + \sum_{i\in H_1} (x_i^k - u_i^k)(y_i^k - v_i^k) \\
&\quad + \sum_{i\in H_2} (u_i^k - x_i^k)(y_i^k + v_i^k) + \sum_{i\in H_3} (x_i^k + u_i^k)(v_i^k - y_i^k) \\
&\le {x^k}^\top y^k - {u^k}^\top v^k + \sum_{i\in H_1} |x_i^k - u_i^k||y_i^k - v_i^k| \\
&\quad + K_1 \left(\sum_{i\in H_2} |x_i^k - u_i^k| + \sum_{i\in H_3} |y_i^k - v_i^k| \right).
\end{aligned}
$$

Note also that

$$
\begin{aligned}
0 &\le {x^*}^\top y^k + {y^*}^\top x^k + (x^* - u^k)^\top (y^* - v^k) \quad \text{(by Assumption 2.1)} \\
(6) \qquad &= {x^*}^\top y^* + (x^k - u^k)^\top y^* + (y^k - v^k)^\top x^* + {u^k}^\top v^k \\
&\le |x^k - u^k|^\top y^* + |y^k - v^k|^\top x^* + {u^k}^\top v^k \quad (\text{since } {x^*}^\top y^* = 0).
\end{aligned}
$$

Combining (5) and (6), we get

$$
\begin{aligned}
&|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k \\
&\le {x^k}^\top y^k + |x^k - u^k|^\top y^* + |y^k - v^k|^\top x^* + \sum_{i\in H_1} |x_i^k - u_i^k||y_i^k - v_i^k| \\
&\quad + \sum_{i\in H_2} |x_i^k - u_i^k| K_1 + \sum_{i\in H_3} K_1 |y_i^k - v_i^k| \\
&\le {x^k}^\top y^k + \nu_k \left(|x^0 - u^0|^\top y^* + |y^0 - v^0|^\top x^* + \sum_{i\in H_1} |x_i^0 - u_i^0||y_i^0 - v_i^0| \right. \\
&\quad \left. + \sum_{i\in H_2} |x_i^0 - u_i^0| K_1 + \sum_{i\in H_3} K_1 |y_i^0 - v_i^0| \right) \quad \text{(by Lemma 2.4(3))}.
\end{aligned}
$$

So,

$$
\begin{aligned}
&\frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k} \\
&\le 1 + \frac{1}{\beta} \left(|x^0 - u^0|^\top y^* + |y^0 - u^0|^\top x^* + \sum_{i\in H_1} |x_i^0 - u_i^0||y_i^0 - v_i^0| \right. \\
&\quad \left. + \sum_{i\in H_2} |x_i^0 - u_i^0| K_1 + \sum_{i\in H_3} K_1 |y_i^0 - v_i^0| \right) =: K. \qquad \square
\end{aligned}
$$

We are now ready to discuss Algorithm 2. This algorithm is identical to Algorithm 1 except that the steplength α_k is defined more precisely. We use the following merit function:

$$
\phi(x, y) := x^\top y + \|Mx + Ny - h\|.
$$

For convenience we note several additional definitions:

$$
x(\alpha) := x^k + \alpha \Delta x^k, \quad y(\alpha) := y^k + \alpha \Delta y^k,
$$

$$\phi(\alpha) := \phi(x(\alpha), y(\alpha)), \quad \phi_k := \phi(x^k, y^k).$$

The steplength α_k is chosen so as to minimize the merit function $\phi(\alpha)$ subject to the following constraints:

$$\alpha \in [0,1], \tag{7a}$$

$$x(\alpha) > 0, \quad y(\alpha) > 0, \tag{7b}$$

$$x(\alpha)^\top y(\alpha) \geq (1-\alpha)\nu_k {x^0}^\top y^0, \tag{7c}$$

$$x(\alpha)_i y(\alpha)_i \geq \left(\frac{\gamma}{n}\right) x(\alpha)^\top y(\alpha), \quad i = 1, \ldots, n, \tag{7d}$$

where $\gamma \in (0,1)$ is chosen so that $\gamma \leq \min(X^0 Y^0 e)/({x^0}^\top y^0/n)$.

Condition (7c) implies that

$$\frac{{x^k}^\top y^k}{{x^0}^\top y^0} \geq \nu_k. \tag{8}$$

The algorithm can now be stated as Algorithm 2.

Algorithm 2– Zhang's second algorithm

Given $(x^0, y^0) > 0$, for $k = 0, 1, 2, \ldots$, do

1. Choose $\sigma_k \in [0,1)$ and let $\mu_k = \frac{1}{n}{x^k}^\top y^k$. Solve the linear system (1) for $(\Delta x^k, \Delta y^k)$.
2. Set the steplength α_k by minimizing $\phi(\alpha)$ subject to the constraints (7). Let $x^{k+1} = x^k + \alpha_k \Delta x^k$ and $y^{k+1} = y^k + \alpha_k \Delta y^k$.

Note that Algorithm 2 is a special case of Algorithm 1, so all the lemmas proved for Algorithm 1 also apply to Algorithm 2. In particular, since ϕ_k is a decreasing sequence, it follows that ${x^k}^\top y^k \leq \phi_0$, for all k. We now show that Algorithm 2 has global Q-linear convergence from any strictly positive starting point.

THEOREM 2.8. *Let $\{\phi^k\}$ be generated by Algorithm 2 with σ^k satisfying $0 < \sigma \leq \sigma^k \leq 1/2$. Then $\{\phi^k\}$ converges to zero at a global Q-linear rate, i.e., there exists $\delta \in (0,1)$ such that*

$$\phi^{k+1} \leq (1-\delta)\phi^k, \quad k = 0, 1, 2, \ldots.$$

Proof. The proof is identical to Zhang's proof except that we can no longer depend on $(x^k - u^k)$ and $(y^k - v^k)$ being nonnegative. To compensate, we redefine Zhang's constants ξ_k and η_k using absolute values as follows:

$$\xi_k := \left(\frac{n}{\gamma}\right)^{1/2} \frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k}, \tag{9}$$

$$\eta_k := 1 - 2\sigma_k + \frac{{\sigma_k}^2}{\gamma} + \frac{2\nu_k |x^0 - u^0|^\top |y^0 - v^0|}{{x^0}^\top y^0}. \tag{10}$$

With these definitions, the remainder of the proof is identical to Zhang's except that Lemma 2.7 is used in place of [23, Lem. 6.1]. □

3. Applications. Lemma 2.7 has applications beyond the analysis of Zhang's algorithms. In [22], Wright proposes a Q-subquadratically convergent algorithm for the LCP which is based closely on Zhang's work. In fact, Wright's algorithm is a

special case of Algorithm 1. And like Zhang, Wright places an identical restriction on the starting point to prove his convergence results. However, by using Lemma 2.7, Wright's results can easily be extended to apply to arbitrary positive starting points. Details of this are given in [3].

Wright's algorithm is the basis of an algorithm for the box-constrained linear complementarity problem (BLCP) [3]. This problem is a natural extension of the LCP that incorporates upper and lower bounds on the variables. Define the box $\mathbf{B} := \prod_{i=1}^{n}[l_i, u_i]$, where $l_i \in [-\infty, \infty)$ and $u_i \in (l_i, \infty]$. The BLCP is then stated as

BLCP
$$\begin{aligned} &\text{Given } M \in \mathbf{R}^{n\times n}, q \in \mathbf{R}^n, \text{find a vector} \\ &\text{triple } (x, w, v) \in \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R}^n \text{ such that} \\ &\quad w - v = Mx + q, \\ &\quad x \in \mathbf{B}, \quad w \geq 0, \quad v \geq 0, \\ &\quad w^{\mathsf{T}}(x - l) + v^{\mathsf{T}}(u - x) = 0, \end{aligned}$$

where we agree that $\pm\infty \times 0 = 0$. This problem form arises in many practical instances and is of particular interest in the field of economics [6]. A library of examples from various disciplines is also described in [5]. The BLCP format is obtained as subproblems in Newton-type methods for rectangular variational inequalities and mixed complementarity problems. The use of lower and upper bounds is crucial in these nonlinear models since many of the functions are not defined outside of the box.

Note that the LCP is a special case of the BLCP (simply set $l = 0$ and $u = \infty$). Conversely, the BLCP can be reformulated as an LCP with higher dimension. As an example, consider the case when both l and u are finite (other cases are discussed in [3]). Let

$$\hat{x} = \begin{bmatrix} x - l \\ v \end{bmatrix}, \quad \hat{y} = \begin{bmatrix} w \\ r \end{bmatrix},$$

$$\hat{M} = \begin{bmatrix} M & I \\ -I & 0 \end{bmatrix}, \quad \hat{q} = \begin{bmatrix} q + Ml \\ u - l \end{bmatrix}.$$

With these definitions the BLCP can be rewritten as the following LCP:

$$\begin{aligned} &\hat{y} = \hat{M}\hat{x} + \hat{q}, \\ &\hat{x}, \hat{y} \geq 0, \quad \hat{x}^{\mathsf{T}}\hat{y} = 0. \end{aligned} \tag{11}$$

Note that to perform the reformulation, we introduced the variable r, which at the solution will satisfy $r = u - x$.

Hence, from a theoretical perspective, the two problems (BLCP and LCP) are equivalent. In fact, one approach to solving the BLCP is simply to reformulate it as an LCP, which can then be solved by an LCP algorithm. This approach has two drawbacks. First, it is inconvenient to perform the reformulation. Second, after the BLCP is reformulated, the resulting LCP has a great deal of structure that should be exploited to solve the problem more efficiently. Thus, from a practical standpoint, it makes sense to develop an algorithm that directly solves the BLCP.

In [3], such an algorithm for BLCP is derived directly from Wright's algorithm for the LCP. Global Q-linear and local Q-subquadratic convergence results are proven for the BLCP algorithm as direct consequences of Wright's convergence analysis, with one important caveat—in order to extend Wright's results to the BLCP algorithm, it

is essential that they be valid from arbitrary positive starting points. This in fact was the original motivation behind the efforts to extend Wright's and Zhang's results to arbitrary starting points. We now describe why this caveat is needed.

The BLCP algorithm is derived from the LCP algorithm by substituting the reformulation of the BLCP directly into the LCP algorithm. The resulting algorithm is then simplified to produce an algorithm that solves the BLCP. For example, to calculate the search direction, Wright's algorithm solves the equation

$$\begin{bmatrix} \hat{M} & -I \\ \hat{Y}^k & \hat{X}^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = \begin{bmatrix} \hat{y}^k - \hat{M}\hat{x}^k - \hat{q} \\ -\hat{X}^k\hat{Y}^k + \sigma_k\mu_k e \end{bmatrix}.$$

With the reformulation defined above, this equation becomes

$$\begin{bmatrix} M & I & -I & 0 \\ -I & 0 & 0 & -I \\ W^k & 0 & X^k - L & 0 \\ 0 & R^k & 0 & V^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta v^k \\ \Delta w^k \\ \Delta r^k \end{bmatrix} = \begin{bmatrix} w^k - v^k - Mx^k - q \\ r^k - u + x^k \\ -(X^k - L)W^k e + \sigma_k\mu_k e \\ -V^k R^k e + \sigma_k\mu_k e \end{bmatrix}, \tag{12}$$

where $W^k := \operatorname{diag}(w^k), V^k := \operatorname{diag}(v^k)$, $R^k := \operatorname{diag}(r^k)$, and $L := \operatorname{diag}(l)$.

Note that if $r^0 = u - x^0$, this equation can be significantly simplified. In particular, the second row of the equation shows that $\Delta r^0 = -\Delta x^0$ and by induction $r^k = u - x^k$ for all subsequent iterations k. (12) then reduces to

$$\begin{bmatrix} M & -I & I \\ W^k & X^k - L & 0 \\ -V^k & 0 & U - X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta w^k \\ \Delta v^k \end{bmatrix} = \begin{bmatrix} -q - Mz^k - v^k + w^k \\ -(X^k - L)W^k e + \sigma_k\mu_k e \\ -(U - X^k)V^k e + \sigma_k\mu_k e \end{bmatrix}.$$

This equation is used to calculate the search direction for the BLCP algorithm. It is straightforward to show that there is a 1–1 map $T : \mathbf{R}^{3n} \to \mathbf{R}^{4n}$ which maps the iterates (x^k, w^k, v^k) generated by the BLCP algorithm to the iterates $(\hat{x}^k, \hat{y}^k)$ generated by the LCP algorithm started from the point $\hat{x}^0 = (x^0 - l, v^0)$, $\hat{y}^0 = (w^0, u - x^0)$. This map is given by the equation

$$T(x, w, v) := (x - l, v, w, u - x).$$

Thus, to extend Wright's convergence results to the BLCP algorithm, it is sufficient to show that the iterates $(\hat{x}^k, \hat{y}^k)$ converge at the appropriate rate. But now the restriction on the starting point becomes significant. The restriction is easily satisfied for the LCP algorithm by making the starting point $(\hat{x}^0, \hat{y}^0)$ large in every component. However, this cannot be done for the analysis of the BLCP algorithm because we require $(\hat{x}^0, \hat{y}^0)$ to lie in the range of T. In particular, any increase in the ith component of $\hat{x}^0$ must be offset by a decrease in the $(i+n)$th component of $\hat{y}^0$. Thus, it may not be possible to find a starting point (x^0, w^0, v^0) for the BLCP for which $(\hat{x}^0, \hat{y}^0) := T(x^0, w^0, v^0)$ satisfies Assumption 2.3. However, by extending Zhang's and Wright's convergence results to arbitrary positive starting points, the difficulty is removed.

REFERENCES

[1] K. ANSTREICHER, *A combined phase* I*–phase* II *projective algorithm for linear programming*, Math. Programming, 43 (1989), pp. 209–223.

[2] K. ANSTREICHER, *A combined phase I–phase II scaled potential algorithm for linear programming*, Math. Programming, 52 (1991), pp. 429–440.

[3] S. C. BILLUPS AND M. C. FERRIS, *Convergence of infeasible interior-point algorithms from arbitrary starting points*, Tech. report 1180, Computer Sciences Department, University of Wisconsin, Madison, WI, 1993.

[4] R. W. COTTLE, J. S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[5] S. P. DIRKSE AND M. C. FERRIS, *MCPLIB: A collection of nonlinear mixed complementarity problems*, Optim. Meth. Software, 5 (1995), pp. 319–345.

[6] ———, *A pathsearch damped Newton method for computing general equilibria*, Mathematical Programming Tech. report 94-03, Computer Sciences Department, University of Wisconsin, Madison, WI, 1994.

[7] O. GÜLER, *Generalized linear complementarity problems*, Math. Oper. Res., 20 (1995), pp. 441–448.

[8] M. KOJIMA, N. MEGGIDO, AND S. MIZUNO, *A primal-dual exterior point algorithm for linear programming*, Tech. report RJ 8500, IBM Alamaden Research Center, San Jose, CA, 1991.

[9] M. KOJIMA, S. MIZUNO, AND M. TODD, *Infeasible interior-point primal-dual potential-reduction algorithms for linear programming*, Tech. report 1023, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1992.

[10] I. LUSTIG, R. MARSTEN, AND D. SHANNO, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[11] ———, *On implementing Mehrotra's predictor-corrector interior point method for linear programming*, SIAM J. Optim., 2 (1992), pp. 435–449.

[12] K. MCSHANE, C. MONMA, AND D. SHANNO, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Comput., 1 (1989), pp. 70–83.

[13] S. MEHROTRA, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[14] S. MIZUNO, *Polynomiality of Kojima–Meggido–Mizuno infeasible-interior-point algorithm for linear programming*, Tech. report 1006, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1993.

[15] R. D. C. MONTEIRO AND S. J. WRIGHT, *A superlinear infeasible-interior-point affine scaling algorithm for LCP*, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1993, preprint MCS-P361-0693.

[16] F. A. POTRA, *An infeasible interior-point predictor-corrector algorithm for linear programming*, Tech. Report 26, Department of Mathematics, University of Iowa, Iowa City, IA, 1992.

[17] ———, *On a predictor-corrector method for solving linear programs from infeasible starting points*, Reports on Computational Mathematics 34, Department of Mathematics, University of Iowa, Iowa City, IA, 1992.

[18] ———, *A quadratically convergent infeasible interior-point algorithm for linear programming*, Tech. report 28, Department of Mathematics, University of Iowa, Iowa City, IA, 1992.

[19] F. A. POTRA AND R. SHENG, *A modified O(nL) infeasible-interior-point algorithm for LCP with quadratic convergence*, Reports on Computational Mathematics 54, Department of Mathematics, University of Iowa, Iowa City, IA, 1994.

[20] S. J. WRIGHT, *A path-following infeasible-interior-point algorithm for linear complementarity problems*, Optim. Meth. Software, 2 (1993), pp. 79–106.

[21] ———, *A path-following interior-point algorithm for linear and quadratic problems*, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1993, preprint MCS-P401-1293.

[22] ———, *An infeasible-interior-point algorithm for linear complementarity problems*, Math. Programming, 67 (1994), pp. 29–51.

[23] Y. ZHANG, *On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

[24] Y. ZHANG AND D. ZHANG, *Superlinear convergence of infeasible interior-point methods for linear programming*, Tech. report 92-15, Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD, 1992.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 326–341, May 1996
© 1996 Society for Industrial and Applied Mathematics
005

GLOBAL CONVERGENCE PROPERTIES OF SOME ITERATIVE METHODS FOR LINEAR COMPLEMENTARITY PROBLEMS*

CHRISTIAN KANZOW†

Abstract. The subject of this work is a class of iterative methods for solving the linear complementarity problem (LCP). These methods are based on a reformulation of the LCP consisting of a (usually) differentiable system of nonlinear equations, to which Newton's method is applied. Thus, the algorithms are locally Q-quadratically convergent. Furthermore, global convergence results for these methods are proved for LCPs associated with R_0-, nondegenerate, and P-matrices. Finally, some promising numerical results are reported for both constructed and randomly generated problems.

Key words. linear complementarity, Newton's method, global convergence

AMS subject classifications. 90C33, 65H10

1. Introduction.

For an arbitrary function $F : \Re^n \to \Re^n$, the nonlinear complementarity problem NCP(F) is the following: Find a vector $x \in \Re^n$ such that

$$x \geq 0, F(x) \geq 0, x^T F(x) = 0. \tag{1}$$

In case F is an affine function, i.e., $F(x) = Mx + q$ for a matrix $M \in \Re^{n \times n}$ and a vector $q \in \Re^n$, this problem is called a linear complementarity problem (LCP) and is denoted by LCP(M,q). Usually, the LCP is stated in the following equivalent way:

$$x \geq 0, w \geq 0, x^T w = 0, w = Mx + q. \tag{2}$$

In this paper, we derive some methods for solving problem (2) by transforming the complementarity conditions into an equivalent system of nonlinear equations, to which Newton's method is applied. Of course, this idea is not new. To the author's knowledge, the first person to characterize (1), and therefore also (2), as a system of equations was Mangasarian [17]. His approach is described in more detail in §2. Several authors present algorithms based on the result of Mangasarian; see, e.g., Watson [27], Subramanian [24], and Ferris and Lucidi [5].

While these authors make use of a (Frechet-) differentiable characterization of NCP(F), the recently proposed methods of Pang [20, 21] and Harker and Xiao [11] employ nondifferentiable characterizations. More precisely, Pang applies the min–map approach

$$\min\{x, F(x)\} = 0,$$

whereas Harker and Xiao use the Minty–map approach

$$F(\max\{0, x\}) + \min\{0, x\} = 0;$$

here the min- and max-operators are taken componentwise. In both papers, the theory of B-differentiable equations (see Pang [20]) is used, and a generalization of Newton's method is applied to the B-differentiable equations $\min\{x, F(x)\} = 0$ and

* Received by the editors July 6, 1993; accepted for publication (in revised form) November 11, 1994.

† Institute of Applied Mathematics, University of Hamburg, Bundesstrasse 55, D-20146 Hamburg, Germany (kanzow@math.uni-hamburg.de).

$F(\max\{0,x\}) + \min\{0,x\} = 0$, respectively. A specialization of Pang's algorithm for LCPs is presented in Harker and Pang [10].

The main motivation for this paper is the work [14] of the author, who derives several methods by utilizing functions $\varphi : \Re^2 \to \Re$ having the property

$$\varphi(a,b) = 0 \iff a \geq 0, b \geq 0, ab = 0, \tag{3}$$

so that

$$\Psi(x) := \begin{pmatrix} \varphi(x_1, F_1(x)) \\ \vdots \\ \varphi(x_n, F_n(x)) \end{pmatrix} = 0 \iff x \text{ solves NCP(F).}$$

While the paper [14] deals mainly with the nonlinear complementarity problem (NCP), some numerical results for an LCP show that at least two members of the class of algorithms presented there lead to very good results for LCPs. Therefore, we continue to investigate these algorithms for problem LCP(M,q) in this paper. Related approaches to the constrained optimization problem are given by Purtov [22]; see also [16].

In §2, some background material and some preliminary results are presented. In §3, we state our class of algorithms and prove that they are (locally) well defined under certain conditions. Section 4 contains global convergence results for our class of algorithms. Numerical results are presented in §5, and we conclude this paper with some final remarks in §6.

To conclude this section, we briefly discuss notation. The symbol $\|.\|$ denotes the Euclidian norm of a vector or its associated matrix norm. A superscript "k" is always used as iteration index, while a subscript "i" usually denotes the ith component of a vector. For $x, w \in \Re^n$, the vector $(x^T, w^T)^T \in \Re^{2n}$ is often written as (x,w). For any matrix $M \in \Re^{n\times n}$ and any subsets $I, J \subseteq \{1,\ldots,n\}$, let M_I denote the matrix which is generated by the ith rows of M $(i \in I)$, i.e., $M_I := (m_{ij})_{i\in I, 1\leq j\leq n}$. Furthermore, we write M_{IJ} for the matrix consisting of the elements $m_{ij}, i \in I, j \in J$. Similarly, for a vector $q \in \Re^n, q_I$ stands for the vector having the components q_i $(i \in I)$. (The elements of the index sets I and J are always arranged in increasing order.)

2. Preliminaries. Following [16], we call any function $\varphi : \Re^2 \to \Re$ satisfying the complementarity condition (3) an *NCP-function.*

Example 2.1. The following functions are NCP-functions:

$$\begin{aligned}
&1.\ \varphi(a,b) := -ab + \frac{1}{2}\min^2\{0, a+b\},\\
&2.\ \varphi(a,b) := -ab + \min^2\{0,a\} + \min^2\{0,b\},\\
&3.\ \varphi(a,b) := \min\{a,b\},\\
&4.\ \varphi(a,b) := \sqrt{a^2+b^2} - a - b,\\
&5.\ \varphi(a,b) := \theta(|a-b|) - \theta(a) - \theta(b), \text{ where}\\
&\qquad \theta(t) := \begin{cases} t|t| & \text{if } |t| \leq \tau,\\ 2\tau t - \tau^2 + c(t-\tau)^2 & \text{if } t > \tau,\\ 2\tau t + \tau^2 - c(t+\tau)^2 & \text{if } t < -\tau \end{cases}\\
&\qquad \text{for constants } \tau > 0 \text{ and } c \geq 0.
\end{aligned}$$

The first two NCP-functions mentioned in Example 2.1 were introduced in Evtushenko and Purtov [3] and the third one was used by Pang in his min–map approach

(cf. §1). The fourth example is due to Fischer, who first used it in connection with nonlinear programming problems [6] and recently in connection with LCPs [7, 8]. The last example was suggested in [13], likewise in connection with nonlinear programming problems.

Note that the examples 3 and 4 are not differentiable everywhere. Nevertheless, as shown in the papers of Fischer [6, 7], his function 4 leads to algorithms which converge locally Q-quadratically even if the underlying problem is degenerate. This is not true for the algorithms presented here.

While it can easily be verified that the functions 1–4 of Example 2.1 are NCP-functions, the same holds for the last function because of the following result, which is the main theorem in the paper of Mangasarian [17].

THEOREM 2.1. *Let $\theta : \Re \to \Re$ be any strictly increasing function with $\theta(0) = 0$. Define*

$$\varphi(a,b) := \theta(|a-b|) - \theta(a) - \theta(b). \tag{4}$$

Then φ is an NCP-function.

Remark. a. Theorem 2.1 remains true if θ is any strictly decreasing function with $\theta(0) = 0$.

b. The NCP-functions 2, 3, and 5 are particular members of Mangasarian's class (4); for functions 2 and 3, just take $\theta(t) = 0.5t|t|$ and $\theta(t) = -0.5t$, respectively.

Several authors present algorithms based on this theorem. For example, Watson [27] describes a continuation method using $\theta(t) = t^3$, and Subramanian [24] considers a Gauss–Newton method using $\theta(t) = t|t|$. The same function is used by Ferris and Lucidi [5], who give numerical results for a stabilized Newton method. The approach of this paper, however, is more general and is not limited to Mangasarian's class of NCP-functions.

Let $(x^*, w^*) \in \Re^{2n}$ denote a solution of LCP(M,q). Under the nondegeneracy assumption

$$x_i^* + w_i^* > 0 \ \ (i \in I := \{1, \dots, n\}), \tag{A}$$

each of our NCP-functions stated in Example 2.1 is at least locally continuously differentiable and has the following properties:

$$\left.\begin{array}{lcll} \frac{\partial\varphi}{\partial b}(x_i^*, w_i^*) & \neq & 0 & (i \in I^{**} := \{i \in I | x_i^* > 0\}), \\ \frac{\partial\varphi}{\partial a}(x_i^*, w_i^*) & = & 0 & (i \in I^{**}), \\ \frac{\partial\varphi}{\partial b}(x_i^*, w_i^*) & = & 0 & (i \notin I^{**}), \\ \frac{\partial\varphi}{\partial a}(x_i^*, w_i^*) & \neq & 0 & (i \notin I^{**}). \end{array}\right\} \tag{P.1}$$

Some of these properties result directly from the definition of an NCP-function, as was shown in [16, Lem. 2.4].

LEMMA 2.2. *Let $(x^*, w^*) \in \Re^{2n}$ be a solution of LCP(M,q), and let φ be any continuously differentiable NCP-function. Then the following assertions hold:*

$$\frac{\partial\varphi}{\partial a}(x_i^*, w_i^*) = 0 \ \ (i \in I^{**}),$$
$$\frac{\partial\varphi}{\partial b}(x_i^*, w_i^*) = 0 \ \ (i \notin I^* := \{i \in I | w_i^* = 0\}).$$

The remaining properties of (P.1) are additional assumptions which we require for our NCP-functions (see Mangasarian and Solodov [18] or [15] for some NCP-functions not satisfying all the conditions of (P.1)). For the special case in which the NCP-function is defined as in Theorem 2.1, the following lemma was shown in [13, Lem. 2.7].

LEMMA 2.3. *Let $(x^*, w^*) \in \Re^{2n}$ be a solution of LCP(M,q) and suppose that the nondegeneracy assumption* (A) *holds. Let $\theta : \Re \to \Re$ be a continuously differentiable and strictly increasing (or decreasing) function with $\theta(0) = 0$. Furthermore, assume that*

$$\theta'(x_i^*) + \theta'(w_i^*) \neq 0 \quad (i \in I).$$

Then φ as defined in (4) *has all the properties stated in* (P.1).

To prove our global convergence results in §4, we need the following property of NCP-functions:

$$\text{(P.2)} \qquad \left.\begin{array}{l}(a \to +\infty \text{ and } b \to +\infty) \\ (a \to -\infty \text{ and } b \to -\infty)\end{array}\right\} \text{ or } \Longrightarrow |\varphi(a,b)| \to +\infty.$$

A stronger version of (P.2) is the following property:

$$\text{(P.3)} \qquad (|a| \to \infty \text{ and } |b| \to \infty) \Longrightarrow |\varphi(a,b)| \to +\infty.$$

LEMMA 2.4. *Let $\theta : \Re \to \Re$ be any strictly increasing function with $\theta(0) = 0$. Define $\varphi : \Re^2 \to \Re$ as in* (4). *If*

$$\text{(5)} \qquad \lim_{t \to +\infty} \theta(t) = +\infty \text{ and } \lim_{t \to -\infty} \theta(t) = -\infty,$$

φ satisfies (P.3) *and therefore also* (P.2).

Proof. Case 1: $a \to +\infty$ and $b \to +\infty$.

Without loss of generality, let $a \geq 0$ and $b \geq 0$. Then $|a - b| \leq \max\{a, b\}$, and consequently we have $\theta(|a - b|) \leq \theta(\max\{a, b\})$ because of the strict monotonicity of θ. Thus, we obtain

$$\varphi(a,b) \leq \theta(\max\{a,b\}) - \theta(a) - \theta(b) = -\theta(\min\{a,b\}) \to_{a,b \to +\infty} -\infty.$$

Case 2: $(a \to -\infty$ and $b \to -\infty)$ or $(a \to +\infty$ and $b \to -\infty)$.

With no loss of generality, assume that $b \leq 0$. Since θ is an increasing function, we have $\theta(a) \leq \theta(a - b) \leq \theta(|a - b|)$. Thus, we obtain

$$\varphi(a,b) = \theta(|a-b|) - \theta(a) - \theta(b) \geq \theta(|a-b|) - \theta(|a-b|) - \theta(b) = -\theta(b) \to_{b \to -\infty} +\infty.$$

Case 3: $a \to -\infty$ and $b \to +\infty$.

Here, the proof is similar to the case $a \to \infty$ and $b \to -\infty$ because of the symmetry of the function φ in its arguments a and b. □

Obviously, an analogous result holds for strictly decreasing functions $\theta : \Re \to \Re$.

Example 2.2. Lemma 2.4 is not true if condition (5) is not satisfied; for example, consider the function $\theta(t) := \exp(t) - 1$. If we define φ as in (4), we obtain for $a = b \to -\infty$: $\varphi(a,b) = \theta(|a-b|) - \theta(a) - \theta(b) \to 2$. Therefore, neither (P.3) nor (P.2) is satisfied.

Now we are ready to prove that each of our NCP-functions defined in Example 2.1 satisfies (P.3).

LEMMA 2.5. *Let φ be any NCP-function of Example* 2.1. *Then φ has the property* (P.3).

Proof. For the NCP-functions 2, 3, and 5, the assertion follows directly from Lemma 2.4 and the remark after Theorem 2.1. Now consider the first function $\varphi(a,b) = -ab + \frac{1}{2}\min^2\{0, a+b\}$. If $a, b \to +\infty$, then obviously $\varphi(a,b) \to -\infty$. If $a, b \to -\infty$, then, assuming without loss of generality $a, b \le 0$, $\varphi(a,b) = -ab + \frac{1}{2}(a+b)^2 = \frac{1}{2}(a^2+b^2) \to +\infty$. If $a \to +\infty$ and $b \to -\infty$ or vice versa, the assertion follows immediately. Finally, let $\varphi(a,b) = \sqrt{a^2+b^2} - a - b$. Since $\varphi(a,b) \ge -\min\{a,b\}$ $\forall$ $a, b \in \Re$, the result is true if $a \to -\infty$ or $b \to -\infty$. Therefore, consider the case $a \to \infty$ and $b \to \infty$. Then the assertion follows from the easily established inequality $\varphi(a,b) \le -0.5\min\{a,b\}$, which is true for all $a, b \ge 0$. □

In §4, we also need the following property:

(P.4) $(a \to -\infty)$ or $(b \to -\infty)$ or $(a \to \infty$ and $b \to \infty) \Longrightarrow |\varphi(a,b)| \to +\infty$.

This property is satisfied by all our NCP-functions because of the following result.

LEMMA 2.6. *Any of our NCP-functions defined in Example* 2.1 *has the property* (P.4).

Proof. Because of Lemma 2.5 and the symmetry of the functions φ in their arguments a and b, it is sufficient to consider the case $a \to -\infty$ and $\{b\}$ bounded. In this case, however, the assertion follows directly from the definitions of our NCP-functions. □

3. The algorithm. Consider the problem LCP(M,q). Let φ be any NCP-function and define the operator $F_\varphi : \Re^{2n} \to \Re^{2n}$ as follows:

$$F_\varphi(x,w) := \begin{pmatrix} Mx + q - w \\ \hline \varphi(x_1, w_1) \\ \vdots \\ \varphi(x_n, w_n) \end{pmatrix}. \tag{6}$$

Then we have the equivalence

$$F_\varphi(x^*, w^*) = 0 \Longleftrightarrow (x^*, w^*) \text{ solves LCP(M,q)}.$$

Thus, to solve LCP(M,q), we can apply to F_φ any method for solving systems of nonlinear equations. Since the convergence theories of these methods usually depend on the existence and nonsingularity of the Jacobian $F'_\varphi(x,w)$, at least in a neighbourhood of a solution (x^*, w^*) of LCP(M,q), we first prove the following result.

THEOREM 3.1. *Let M be a nondegenerate matrix, i.e.,* $\det(M_{JJ}) \ne 0$ *for any subset $J \subseteq \{1, \dots, n\}$. Let (x^*, w^*) be a solution of LCP(M,q), and let φ be a continuously differentiable NCP-function having the property* (P.1). *Define F_φ as in* (6). *Then the Jacobian matrix $F'_\varphi(x^*, w^*)$ is nonsingular.*

Proof. The Jacobian of F_φ at (x^*, w^*) is given by

$$\left(\begin{array}{ccc|ccc} & M & & & -I & \\ \hline \frac{\partial\varphi}{\partial a}(x_1^*, w_1^*) & & 0 & \frac{\partial\varphi}{\partial b}(x_1^*, w_1^*) & & 0 \\ & \ddots & & & \ddots & \\ 0 & & \frac{\partial\varphi}{\partial a}(x_n^*, w_n^*) & 0 & & \frac{\partial\varphi}{\partial b}(x_n^*, w_n^*) \end{array}\right). \tag{7}$$

Consider $F'_\varphi(x^*, w^*)q = 0$ for an arbitrary vector $q \in \Re^{2n}$. Partition $q = (q^{(1)}, q^{(2)})$, with $q^{(i)} \in \Re^n, i = 1, 2$. Then we get

$$Mq^{(1)} - q^{(2)} = 0, \tag{8}$$

$$\frac{\partial \varphi}{\partial a}(x_i^*, w_i^*)q_i^{(1)} + \frac{\partial \varphi}{\partial b}(x_i^*, w_i^*)q_i^{(2)} = 0 \ \ (i \in I). \tag{9}$$

Using (P.1), we obtain from the second equation

$$q_i^{(1)} = 0 \ \ (i \notin I^{**}),$$
$$q_i^{(2)} = 0 \ \ (i \in I^{**}).$$

Using this, it follows from (8) with $J := I^{**}$ that

$$M_{JJ}q_J^{(1)} = 0.$$

Since M is nondegenerate, M_{JJ} is nonsingular. Thus, we get

$$q_J^{(1)} = 0, \text{ i.e., } q_i^{(1)} = 0 \ \ (i \in I^{**}).$$

Using (8) again, we also obtain $q_i^{(2)} = 0$ $(i \notin I^{**})$. Thus $q = 0$ and $F'_\varphi(x^*, w^*)$ is nonsingular. □

We now state a result concerning the nonsingularity of the Jacobian $F'_\varphi(x, w)$ at an arbitrary point (x, w). Similar results are known for interior-point methods; see, e.g., Zhang, Tapia, and Potra [28].

LEMMA 3.2. *Consider problem LCP(M,q) with M being positive semidefinite. Let φ be a continuously differentiable function, and define F_φ as in* (6). *Let* $(x, w) \in \Re^{2n}, D_a := \operatorname{diag}(\ldots, \frac{\partial \varphi}{\partial a}(x_i, w_i), \ldots)$, *and* $D_b := \operatorname{diag}(\ldots, \frac{\partial \varphi}{\partial b}(x_i, w_i), \ldots)$. *Suppose that D_aD_b is positive definite. Then the Jacobian matrix $F'_\varphi(x, w)$ is nonsingular.*

Proof. By our assumption, the matrices D_a and D_b are nonsingular. Since

$$\left(\begin{array}{c|c} I & -MD_a^{-1} \\ \hline 0 & I \end{array}\right) F'_\varphi(x, w) = \left(\begin{array}{c|c} 0 & -I - MD_a^{-1}D_b \\ \hline D_a & D_b \end{array}\right),$$

the result follows if $I + MD_a^{-1}D_b$ is nonsingular. The latter is true because the matrix $(I + MD_a^{-1}D_b)D_b^{-1}D_a = D_b^{-1}D_a + M$ is positive definite. □

Note that if (x, w) is not a solution of LCP(M,q), the assumptions made for the diagonal matrices D_a and D_b defined in Lemma 3.2 are very likely to hold for the NCP-function 4 of Example 2.1, whereas they never hold for the NCP-function 3. Next, we formally state our class of algorithms, which result from Newton's method being applied to $F_\varphi(x, w) = 0$. This method is globalized using the nonmonotone Armijo line search rule first introduced by Grippo, Lampariello, and Lucidi [9].

ALGORITHM 3.1.

(S.0): Let $\varphi : \Re^2 \to \Re$ be an NCP-function satisfying (P.1). Define $F_\varphi(x, w)$ as in (6). Choose $(x^0, w^0) \in \Re^{2n}, \beta, \sigma \in (0, 1), p \in \mathcal{N}$, and let $k := 0$.

(S.1): If $\|F_\varphi(x^k, w^k)\| = 0$, stop: (x^k, w^k) solves LCP(M,q).

(S.2): Compute $(\Delta x^k, \Delta w^k) \in \Re^{2n}$ as the solution of the linear system

$$F'_\varphi(x^k, w^k)\begin{pmatrix} \Delta x^k \\ \Delta w^k \end{pmatrix} = -F_\varphi(x^k, w^k).$$

(S.3): Define $p_k := \min\{k, p\}$ and let $t^k = \beta^{m_k}$, where m_k is the smallest nonnegative integer m which satisfies the nonmonotone Armijo-condition

$$\|F_\varphi(x^k + \beta^m \Delta x^k, w^k + \beta^m \Delta w^k)\|^2 \le \max_{j=k-p_k+1,\dots,k} \|F_\varphi(x^j, w^j)\|^2 - 2\sigma\beta^m \|F_\varphi(x^k, w^k)\|^2.$$

(S.4): Set $x^{k+1} := x^k + t^k \Delta x^k, w^{k+1} := w^k + t^k \Delta w^k, k := k+1$, and go to (S.1).

Remark. Let $\varphi(x^k, w^k) := (\varphi(x_1^k, w_1^k), \dots, \varphi(x_n^k, w_n^k))^T \in \Re^n$ and define $D_a^k := \operatorname{diag}(\frac{\partial\varphi}{\partial a}(x_1^k, w_1^k), \dots, \frac{\partial\varphi}{\partial a}(x_n^k, w_n^k)), D_b^k := \operatorname{diag}(\frac{\partial\varphi}{\partial b}(x_1^k, w_1^k), \dots, \frac{\partial\varphi}{\partial b}(x_n^k, w_n^k))$. Using these abbreviations, it is easy to see that the linear system in step (S.2), which is of dimension $2n$, can be solved as follows: Compute Δx^k from the $n \times n$-system

$$(D_a^k + D_b^k M)\Delta x^k = -\varphi(x^k, w^k) - D_b^k(Mx^k + q - w^k) \tag{10}$$

and set

$$\Delta w^k = M(x^k + \Delta x^k) + q - w^k.$$

If $F'_\varphi(x^k, w^k)$ is nonsingular and $\frac{\partial\varphi}{\partial b}(x_i^k, w_i^k) = 0$ for an index $i \in I$, then $\frac{\partial\varphi}{\partial a}(x_i^k, w_i^k) \neq 0$, and we obtain from the ith row of (10)

$$\Delta x_i^k = -\frac{\varphi(x_i^k, w_i^k)}{\frac{\partial\varphi}{\partial a}(x_i^k, w_i^k)}.$$

Hence, the dimension of the $n \times n$-system (10) can be reduced further. A more detailed description is given in [14, §3].

Remark. Let φ be any of the NCP-functions defined in Example 2.1 and let $(x^*, w^*) \in \Re^{2n}$ denote a solution of LCP(M,q). Assume that the nondegeneracy condition (A) holds at (x^*, w^*). Then it is not difficult to show that the Jacobian F'_φ is a locally Lipschitz-continuous function in (x^*, w^*), i.e.,

$$\|F'_\varphi(x, w) - F'_\varphi(x^*, w^*)\| \le L \left\| \begin{pmatrix} x \\ w \end{pmatrix} - \begin{pmatrix} x^* \\ w^* \end{pmatrix} \right\|$$

for some $L > 0$ and for all (x, w) in a neighbourhood of (x^*, w^*). Therefore, under the assumptions of Theorem 3.1, Algorithm 3.1 is locally well defined and Q-quadratically convergent.

4. Global convergence. In this section, we prove that the function $\|F_\varphi(x, w)\|$ has bounded level sets under certain assumptions. In Theorem 4.1 this is shown under the assumptions that M is a nondegenerate matrix and the NCP-function φ satisfies (P.3). The case in which M is a P-matrix and φ fulfills (P.2) is handled in Theorem 4.2. Theorem 4.3 shows that the assertion is also true for R_0-matrices M and NCP-functions φ satisfying (P.4). According to these results, any sequence $\{(x^k, w^k)\}_{k\in\mathcal{N}}$ generated by our algorithm must have at least one accumulation point.

THEOREM 4.1. *Let M be a nondegenerate matrix. Let $(x^0, w^0) \in \Re^{2n}$ be an arbitrary vector, and let $F_\varphi(x, w)$ be defined as in* (6). *Assume that φ is a continuous NCP-function having the property* (P.3). *Then the level set $L(x^0, w^0) := \{(x, w) \in \Re^{2n} \mid \|F_\varphi(x, w)\| \le \|F_\varphi(x^0, w^0)\|\}$ is bounded and therefore compact.*

Proof. Using the triangle inequality, we get the following for all $(x, w) \in L(x^0, w^0)$:

$$\big|\|Mx + q\| - \|w\|\big| \le \|Mx + q - w\| \le \|F_\varphi(x, w)\| \le \|F_\varphi(x^0, w^0)\|$$

and therefore

$$\|w\| \leq \|F_\varphi(x^0, w^0)\| + \|q\| + \|M\|\,\|x\|. \tag{11}$$

Assume that there exists a sequence $\{(x^k, w^k)\}_{k\in\mathcal{N}} \subseteq L(x^0, w^0)$ such that $\|(x^k, w^k)\| \to \infty$ $(k \in \mathcal{N})$. Then $\|x^k\| \to \infty$ $(k \in \mathcal{N})$; otherwise $\{\|x^k\|\}_{k\in\mathcal{N}}$ and $\{\|w^k\|\}_{k\in\mathcal{N}}$ are bounded because of (11). Let $J \neq \emptyset$ be the following subset of $I = \{1, \ldots, n\}$:

$$J := \{j \in I \mid \{x_j^k\}_{k\in\mathcal{N}} \text{ is unbounded}\}.$$

Let $K := I \setminus J$ be its complementary subset. Let m_i^T denote the ith row vector of the matrix M. Then we have

$$|m_i^T x^k + q_i| - |w_i^k| \leq |m_i^T x^k + q_i - w_i^k| \leq \|Mx^k + q - w^k\| \leq \|F_\varphi(x^0, w^0)\|. \tag{12}$$

Since

$$|\varphi(x_i^k, w_i^k)| \leq \|F_\varphi(x^k, w^k)\| \leq \|F_\varphi(x^0, w^0)\|$$

is bounded and because of (P.3) and (12), it follows that the sequence $\{|m_j^T x^k + q_j|\}_{k\in\mathcal{N}}$ $(j \in J)$ is bounded. Therefore, the sequence $\{\|M_J x^k\|\}_{k\in\mathcal{N}}$ is also bounded because

$$\|M_J x^k\| \leq \|M_J x^k + q_J\| + \|q_J\|.$$

Hence, using the nondegeneracy of the matrix M, the sequence $\{\|x_J^k\|\}_{k\in\mathcal{N}}$ is bounded since

$$\begin{aligned}\|x_J^k\| &\leq \|M_{JJ}^{-1}\|\,\|M_{JJ}x_J^k\| \\ &= \|M_{JJ}^{-1}\|(\|M_{JJ}x_J^k + M_{JK}x_K^k - M_{JK}x_K^k\|) \\ &\leq \|M_{JJ}^{-1}\|(\|M_J x^k\| + \|M_{JK}\|\,\|x_K^k\|)\end{aligned}$$

and because all components x_j^k $(j \in K)$ are bounded. This, however, contradicts the definition of the index set J. Thus, $L(x^0, w^0)$ is bounded and therefore, by continuity, also compact. □

Note that Theorem 4.1 generalizes a result of Harker and Pang [10]. In the next example, we show that Theorem 4.1 is not true if the NCP-function φ fulfills (P.2) but not (P.3).

Example 4.1. Let φ be any NCP-function satisfying (P.2) but not (P.3). (Obviously, such a function exists.) Without loss of generality, we assume that $\varphi(a, b)$ remains bounded for $a \to +\infty$ and $b \to -\infty$. Let $n = 1, M = (-1), q = 0$, and $(x^k, w^k) = (k, -k)$ $(k \in \mathcal{N})$. Then M is a nondegenerate matrix and

$$\|F_\varphi(x^k, w^k)\| = \left\| \begin{pmatrix} -x^k - w^k \\ \varphi(x^k, w^k) \end{pmatrix} \right\| = \left\| \begin{pmatrix} 0 \\ \varphi(k, -k) \end{pmatrix} \right\|$$

remains bounded, but

$$\|(x^k, w^k)\| \to +\infty \quad (k \in \mathcal{N}).$$

Therefore, if (x^0, w^0) is chosen such that $\|F_\varphi(x^k, w^k)\| \leq \|F_\varphi(x^0, w^0)\|$ $(k \in \mathcal{N})$, e.g., $(x^0, w^0) := (k_0, k_0)$ for $k_0 \in \mathcal{N}$ sufficiently large, the level set $L(x^0, w^0)$ is not bounded.

THEOREM 4.2. *Let M be a P-matrix, i.e., let $\det(M_{JJ}) > 0$ for all $J \subseteq \{1,\dots,n\}$. Let $(x^0,w^0) \in \Re^{2n}$ be an arbitrary vector, and let $F_\varphi(x,w)$ be defined as in* (6). *Assume that φ is a continuous NCP-function having the property* (P.2). *Then the level set $L(x^0,w^0)$ is compact.*

Proof. Suppose that there exists a sequence $\{x^k,w^k\}_{k\in\mathcal{N}} \subseteq L(x^0,w^0)$ such that $\|(x^k,w^k)\| \to \infty$. Consider the normalized sequence $\frac{(x^k,w^k)}{\|(x^k,w^k)\|}$, and let (x^*,w^*) be any accumulation point of it. Since

$$\frac{\|Mx^k+q-w^k\|}{\|(x^k,w^k)\|} \le \frac{\|F_\varphi(x^0,w^0)\|}{\|(x^k,w^k)\|} \to 0 \text{ as } k\to\infty,$$

we obtain

$$Mx^* = w^*. \tag{13}$$

From (13) and the fact that $\|(x^*,w^*)\| = 1$, we furthermore get $x^* \neq 0$. Since M is a P-matrix, there exists an index $i \in I$ such that $x_i^* w_i^* = x_i^*(Mx^*)_i > 0$; see, e.g., Cottle, Pang, and Stone [1]. Therefore, we have $x_i^* > 0$ and $w_i^* > 0$ or $x_i^* < 0$ and $w_i^* < 0$. Hence, we have $x_i^k \to +\infty$ and $w_i^k \to +\infty$ or $x_i^k \to -\infty$ and $w_i^k \to -\infty$ on a subsequence. By property (P.2), this yields $|\varphi(x_i^k,w_i^k)| \to +\infty$, which contradicts $|\varphi(x_i^k,w_i^k)| \le \|F_\varphi(x^k,w^k)\| \le \|F_\varphi(x^0,w^0)\|$. □

Because of Example 4.1, Theorem 4.2 is not true if the assumption "M P-matrix" is replaced by "M nondegenerate." Without the assumption (P.2), Theorem 4.2 is also not true; this we will show in the following example.

Example 4.2. Let $n = 1, M = (1), q = 0, \theta(t) := \arctan(t)$, and $\varphi(a,b) := \theta(|a-b|) - \theta(a) - \theta(b)$. Then M is a P-matrix and φ is an NCP-function not satisfying (P.2). Consider the sequence $\{(x^k,w^k)\}_{k\in\mathcal{N}}$ defined by $(x^0,w^0) := (\pi,0)$ and $(x^k,w^k) := (k,k)$ for $k = 1,2,\dots$. Then we have

$$F_\varphi(x^k,w^k) = \begin{pmatrix} x^k - w^k \\ \varphi(x^k,w^k) \end{pmatrix} \to \begin{pmatrix} 0 \\ -\pi \end{pmatrix}$$

and

$$\|F_\varphi(x^k,w^k)\| \le \pi = \|F_\varphi(x^0,w^0)\| \ \ (k\in\mathcal{N}),$$

but

$$\|(x^k,w^k)\| \to +\infty \ \ (k\in\mathcal{N}).$$

Therefore, the level set $L(x^0,w^0)$ is not bounded.

The following theorem is an extension of Lemma 6.1 of Fischer [7].

THEOREM 4.3. *Let M be an R_0-matrix, i.e., let LCP $(M,0)$ have the unique solution $x = 0$. Let φ be a continuous NCP-function having the property* (P.4). *Then the level set $L(x^0,w^0)$ is compact for any given vector $(x^0,w^0) \in \Re^{2n}$.*

Proof. Assume that there exists an unbounded sequence $\{(x^k,w^k)\}_{k\in\mathcal{N}} \subseteq L(x^0,w^0)$. As in the proof of Theorem 4.2, we obtain a vector (x^*,w^*) with $x^* \neq 0$ and $Mx^* = w^*$. We consider the following two cases.

Case 1. There exists an index $i \in I$ such that $x_i^* < 0$ or $w_i^* < 0$.

From the definition of (x^*,w^*), it follows that $x_i^k \to -\infty$ or $w_i^k \to -\infty$ on a subsequence, yielding $|\varphi(x_i^k,w_i^k)| \to +\infty$ because of (P.4). This, however, contradicts $|\varphi(x_i^k,w_i^k)| \le \|F_\varphi(x^0,w^0)\|$.

Case 2. We have $x_i^* \geq 0$ and $w_i^* \geq 0$ for all $i \in I$.

If $(x^*)^T w^* = 0$, then $x^* \neq 0$ is a solution of LCP(M,0), which contradicts the fact that M is an R_0-matrix. Otherwise, if $(x^*)^T w^* > 0$, there exists an index $i \in I$ such that $x_i^* > 0$ and $w_i^* > 0$. Hence, we have $x_i^k \to +\infty$ and $w_i^k \to +\infty$ on a subsequence. Consequently, $|\varphi(x_i^k, w_i^k)| \to +\infty$ because of (P.4), which again contradicts the fact that $|\varphi(x_i^k, w_i^k)| \leq \|F_\varphi(x^0, w^0)\|$. □

Note that the main theoretical results of this paper (Theorems 4.1–4.3) are independent of the differentiability of the NCP-function φ. We finally state a standard result; see, e.g., Stoer and Bulirsch [23, §5.4] or Dennis and Schnabel [2].

THEOREM 4.4. *Let φ be any NCP-function such that F'_φ exists and is Lipschitz continuous. Let $\{(x^k, w^k)\}_{k \in \mathcal{N}}$ be a sequence produced by Algorithm* 3.1. *If (x^*, w^*) is any accumulation point of this sequence such that the Jacobian matrix $F'_\varphi(x^*, w^*)$ is nonsingular, (x^*, w^*) solves LCP(M,q).*

5. Numerical results. In this section, we present some numerical results for Algorithm 3.1, which we have applied to several constructed and randomly generated examples. We have implemented the algorithm in MATLAB and tested it on a 486-type computer. The implementation includes the following two heuristics. To avoid singular matrices during an iteration, the matrix of the linear system (10) is slightly perturbed on its diagonal when its estimated condition number is too bad; in this case, one could have used the steepest descent direction of the objective function $\|F_\varphi\|^2$ instead. Moreover, if the steplength t^k becomes less than $\frac{1}{8}, t^k = \frac{1}{8}$ is accepted even if the nonmonotone Armijo-condition in step (S.3) is not satisfied; also see Fischer [7]. This leads to slightly better results in a very few cases. Usually, the full stepsize $t^k = 1$ has been taken in almost all iterations because of the nonmonotone line search.

The termination criterion used is $\|F_\varphi(x^k, w^k)\| < 10^{-8}$. For the parameters, we have taken the values $\beta = 0.5, \sigma = 10^{-4}$, and $p = 10$. The starting vector $(x^0, w^0) \in \Re^{2n}$ has been chosen as follows: Let $x^0 \in \Re^n$ as in the examples and set $w^0 := Mx^0 + q$. Although the NCP-function 4 of Example 2.1 is not differentiable in $(0,0)^T$, we have tested our algorithm with this function too since almost all solutions obtained in the examples are nondegenerate; thus, the Jacobian F'_φ is well defined at least in a neighbourhood of these solutions, even for this NCP-function. On the other hand, the NCP-function 3 of Example 2.1 is not differentiable at all points $(a, b) \in \Re^2$ with $a = b$, and results for this function are therefore excluded here. However, for some examples, we compare our algorithm with the one of Harker and Pang [10] (denoted by HP in the tables that follow), which is based on the NCP-function 3 and which avoids nondifferentiable points by a perturbation rule.

Example 5.1 (see Murty [19]): n=5.

$$M = \begin{pmatrix} 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 \\ 1 & 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \end{pmatrix}, \quad q = (-1, -1, -1, -1, -1)^T.$$

This example is a reformulation of a bimatrix game as an LCP (see Murty [19], Example 1.9). The matrix M is degenerate. The number of iterations our algorithm needs using the NCP-functions 1, 2, 4, and 5 (with $c = 0, \tau = 1$) of Example 2.1 with different starting vectors x^0 are listed in Table 1. This example has several solutions including two degenerate ones, namely, $(x^1)^* = (0.2, 0.4, 0, 0.5, 0)^T, (w^1)^* = (0, 0, 0, 0, 0.6)^T$ and $(x^2)^* = (0, 1, 0, 0.5, 0)^T, (w^2)^* = (0, 0, 1, 0, 2)^T$. Convergence to these degenerate solutions is indicated by superscripts 1 and 2, respectively.

TABLE 1
Number of iterations for Algorithm 3.1 *being applied to Example* 5.1.

starting vector	1	2	4	5
(0,0,0,0,0)	18^1	18^1	6^2	18^1
(1,1,1,1,1)	7	13	10^1	14^1
(10,...,10)	11	13	7^2	14^1
(100,...,100)	13	15	7^2	14^1
(1000,...,1000)	21	18	7^2	14^1

TABLE 2
Number of iterations for Algorithm 3.1 *being applied to Example* 5.2.

starting vector	1	2	4	5
(0,0,0,0)	9	11	6	8
(1,1,1,1)	6	7	6	6
(10,...,10)	11	12	7	15
(100,...,100)	15	15	7	15
(1000,...,1000)	17	19	7	15

Example 5.2 (see Hock and Schittkowski [12]): n=4.

$$M = \begin{pmatrix} 4 & 2 & 2 & 1 \\ 2 & 4 & 0 & 1 \\ 2 & 0 & 2 & 2 \\ -1 & -1 & -2 & 0 \end{pmatrix}, \quad q = (-8, -6, -4, 3)^T.$$

This example comes from a quadratic programming problem (see Hock and Schittkowski [12, Ex. 35, BEALE'S PROBLEM]). The Karush–Kuhn–Tucker optimality conditions of this problem lead to the LCP described here. The matrix M is degenerate. The solution is $x^* = \left(\frac{4}{3}, \frac{7}{9}, \frac{4}{9}, \frac{2}{9}\right)^T, w^* = (0,0,0,0)^T$. The number of iterations are indicated in Table 2. The parameters for the NCP-function 5 have been taken as $c = 0$ and $\tau = 10^{-2}$.

Example 5.3 (see Hock and Schittkowski [12]): n=7.

$$M = \begin{pmatrix} 2 & 0 & -1 & 0 & 1 & 3 & 0 \\ 0 & 1 & 0 & 0 & 2 & 1 & -1 \\ -1 & 0 & 2 & 1 & 1 & 2 & -4 \\ 0 & 0 & 1 & 1 & 1 & -1 & 0 \\ -1 & -2 & -1 & -1 & 0 & 0 & 0 \\ -3 & -1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad q = (-1, -3, 1, -1, 5, 4, -1.5)^T.$$

Originally, this example was also a quadratic programming problem (see Hock and Schittkowski [12, Ex. 76]). Again, the matrix M is degenerate. According to the results in Table 3, the algorithms using the NCP-functions 1, 2, and (partly) 5 (with $c = 0, \tau = 10^{-3}$) have some difficulties with this example, whereas the algorithm using the NCP-function 4 converges very fast to the nondegenerate solution $x^* \doteq (0.27, 2.09, 0, 0.54, 0.45, 0, 0)^T, w^* \doteq (0, 0, 1.72, 0, 0, 1.63, 0.59)^T$.

Example 5.4 (see Watson [27]): n=5.

$$M = \begin{pmatrix} 1 & -5 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ -3 & -3 & 1 & 2 & -1 \\ -4 & -4 & 2 & 1 & 2 \\ -5 & -5 & -1 & 4 & 3 \end{pmatrix}, \quad q = (-1, 0, 0, 0, 0)^T.$$

TABLE 3
Number of iterations for Algorithm 3.1 *being applied to Example* 5.3.

starting vector	1	2	4	5
(0,...,0)	17	10	7	6
(1,...,1)	31	35	7	8
(10,...,10)	14	51	9	35
(100,...,100)	112	> 300	9	37
(1000,...,1000)	30	64	9	37

TABLE 4
Number of iterations for Algorithm 3.1 *being applied to Example* 5.4.

starting vector	1	2	4	5
(0,...,0)	6	6	5	7
(1,...,1)	8	8	5	3
(10,...,10)	10	11	5	3
(100,...,100)	13	15	5	3
(1000,...,1000)	17	18	5	3

The solution of this example is $x^* = (1, 0, 1, 1.2, 0.4)^T, w^* = (0, 1, 0, 0, 0)^T$, which is nondegenerate. The number of iterations is indicated in Table 4. None of our algorithms has any difficulty solving this problem. Once more, however, the results for the NCP-functions 4 and 5 ($c = 0$ and $\tau = 10^{-3}$ have been used) are better than the ones obtained by the NCP-functions 1 and 2. In the following examples, we will therefore concentrate on these two NCP-functions and compare their results with the ones obtained by Harker and Pang [10].

The starting vector used in all the following examples is $x^0 = (0, \ldots, 0)^T$.

Example 5.5 (see Murty [19]): n variable.

$$M = \begin{pmatrix} 1 & 2 & 2 & \ldots & 2 \\ 0 & 1 & 2 & \ldots & 2 \\ 0 & 0 & 1 & \ldots & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}, \quad q = (-1, \ldots, -1)^T.$$

The matrix M of this example is a P-matrix and is therefore nondegenerate, but not positive definite. The solution is $x^* = (0, \ldots, 0, 1)^T, w^* = (1, \ldots, 1, 0)^T$. For this example, Lemke's complementary pivot algorithm as well as Cottle and Danzig's principal pivoting method are known to run in exponential time (see Murty [19, Chap. 6]). Furthermore, the number of iterations needed by the method of Harker and Pang [10] seems to increase exponentially with the dimension n of this problem. On the other hand, the numerical results obtained with our algorithms based on the NCP-functions 4 and 5 (with $c = 0, \tau = 10^{-4}$) show that the number of iterations seems to increase only linearly with the dimension n; see Table 5.

Example 5.6 (see Fathi [4]): n variable.

$$M = \begin{pmatrix} 1 & 2 & 2 & \ldots & 2 \\ 2 & 5 & 6 & \ldots & 6 \\ 2 & 6 & 9 & \ldots & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 6 & 10 & \ldots & 4(n-1)+1 \end{pmatrix}, \quad q = (-1, \ldots, -1)^T.$$

TABLE 5
Number of iterations for Algorithm 3.1 *being applied to Example* 5.5.

n	8	16	32	64	128
4	12	20	36	69	134
5	10	23	44	106	232
HP	9	20	72	208	> 300

TABLE 6
Number of iterations for Algorithm 3.1 *being applied to Example* 5.6.

n	8	16	32	64	128
4	12	20	36	68	132
5	12	21	37	28	27
HP	8	16	32	65	63

Here the matrix M is positive definite. The solution of this LCP is $x^* = (1, 0, \ldots, 0)^T$, $w^* = (0, 1, \ldots, 1)^T$. As in Example 5.4, both Lemke's complementary pivot algorithm and Cottle and Danzig's principal pivoting method run in exponential time for this example (see Murty [19, Chap. 6]). The numerical results obtained with our algorithms using the NCP-functions 4 and 5 (with $c = 0, \tau = 10^{-4}$) are summarized in Table 6. The results for function 4 are very similar to the ones obtained by Harker and Pang, except for $n = 128$. On the other hand, function 5 clearly outperforms Harker and Pang's method for the larger values of n. We would like to mention that the NCP-functions 1 and 2 lead to very bad results for the Examples 5.5 and 5.6.

In the following randomly generated problems, $c = 0$ and $\tau = 10^{-10}$ have been chosen as parameters for the NCP-function 5.

Example 5.7 (see Harker and Pang [10]). The matrix M is computed as follows: Let $A, B \in \Re^{n \times n}$ and $q, \eta \in \Re^n$ be randomly generated such that $a_{ij}, b_{ij} \in (-5, 5), q_i \in (-500, 500), \eta_i \in (0.0, 0.3)$, with B being skew-symmetric. Define $M = A^T A + B + \operatorname{diag}(\eta)$. Then M is a P-matrix. For several values of n, ten examples have been generated in this way. The maximum, average, and minimum number of iterations needed by our algorithms are summarized in Table 7 and compared with the results obtained by Harker and Pang [10]. The average number of iterations is almost independent of the dimension for all methods. While the algorithm based on function 4 needs more iterations than Harker and Pang's method (although the number of iterations is still relatively small), our results obtained with function 5 are slightly better than theirs.

Example 5.8 (see Harker and Pang [10]). In this example, M is computed in the same way as in the previous example, and $q \in \Re^n$ is randomly generated with entries $q_i \in (-500, 0)$ (so-called "hard examples"). Table 8 contains the numerical results and compares them again with the results obtained by Harker and Pang [10]. For the larger dimension $n = 100$, the results obtained with our algorithms are slightly better than the ones of Harker and Pang.

The numerical results for the previous examples demonstrate the following: The algorithms based on the NCP-functions 4 and 5 are useful tools for solving LCPs. Their performance is often better than the ones of Harker and Pang's method. Moreover, the methods seem to be robust since they usually have no difficulties solving LCPs which do not satisfy any of the assumptions stated in our theoretical results (see the small-dimensional examples) and since they are also able to solve problems that have degenerate solutions (see Example 5.1). Furthermore, we have observed that the

TABLE 7
Number of iterations for Algorithm 3.1 *being applied to Example* 5.7.

n		4	5	HP
50	Max.	14.0	4.0	5.0
	Avg.	11.5	3.0	3.8
	Min.	9.0	2.0	2.0
100	Max.	15.0	6.0	5.0
	Avg.	12.7	3.7	4.3
	Min.	9.0	3.0	4.0
150	Max.	15.0	5.0	7.0
	Avg.	12.8	3.7	4.9
	Min.	10.0	3.0	4.0
175	Max.	17.0	5.0	5.0
	Avg.	13.7	3.3	4.6
	Min.	10.0	3.0	4.0

TABLE 8
Number of iterations for Algorithm 3.1 *being applied to Example* 5.8.

n		4	5	HP
50	Max.	19.0	36.0	18.0
	Avg.	12.4	16.0	12.9
	Min.	11.0	3.0	11.0
100	Max.	18.0	34.0	36.0
	Avg.	16.9	22.1	24.2
	Min.	15.0	4.0	19.0

results obtained with the NCP-function 5 can often be improved for smaller values of τ, which, however, is disadvantageous because φ comes close to a nondifferentiable function.

Finally, we also compared a FORTRAN implementation of our algorithm with Lemke's method, as written by Tomlin [25]. For this, we consider the Examples 5.5 and 5.6 for the dimensions $n = 10, 20, \ldots, 50$. Since our version of Tomlin's implementation is currently not able to solve larger problems, we consider only these dimensions. The CPU times (in seconds) are given in Table 9. All runs were done on an IBM Risc 6000/550 workstation. The starting vector used for all runs is $x^0 := (0, \ldots, 0)$. The results for Algorithm 3.1 are based on Fischer's NCP-function 4. The linear systems are solved using the subroutine F04ARF from the NAG library. Note that this is a solver for general nonsymmetric matrices, whereas the linear systems for Examples 5.5 and 5.6 have a special structure. We have not exploited this special structure because the linear systems in Algorithm 3.1 are nonsymmetric for general complementarity problems.

As can be seen from Table 9, Algorithm 3.1 outperforms Lemke's method for Murty's example 5.5. On the other hand, the outstanding results of Lemke's method on Fathi's test problem 5.6 are due to the fact that it solves this problem in just one iteration for all dimensions n. Taking this into account, the results of Algorithm 3.1 for this example are also very good.

6. Final remarks. In this paper, we have derived several algorithms for problem LCP(M,q). They are the result of Newton's method being applied to a characterization of LCP(M,q) consisting of a nonlinear system of equations. As demonstrated by the numerical experiments in the last section, two methods seem to be quite promising, namely, the methods based on the NCP-functions 4 and 5 of Example 2.1. In particular, the results show that the frequently used member $\theta(t) = t|t|$ of Mangasar-

TABLE 9
A comparison of Algorithm 3.1 *and Lemke's method (CPU times in seconds).*

Example	5.5		5.6	
Method	Lemke	Alg. 3.1	Lemke	Alg. 3.1
n=10	0.01	0.01	0.00	0.01
n=20	0.09	0.02	0.00	0.10
n=30	0.53	0.12	0.00	0.20
n=40	1.01	0.24	0.00	0.30
n=50	1.64	0.51	0.00	0.40

ian's class of NCP-functions (4) is usually not the best one. Thus, we think that the numerical results presented by Ferris and Lucidi [5] could be improved substantially by replacing $\theta(t) = t|t|$ by the θ-function used for our NCP-function 5.

Our methods, however, are not cheap since we must solve, at each iteration, a linear system of the form

$$(D_a^k + D_b^k M)x = b^k,$$

where $b^k \in \Re^n, D_a^k, D_b^k \in \Re^{n\times n}$ are (in general) singular diagonal matrices and where $M \in \Re^{n\times n}$ is a nonsingular and often unsymmetric matrix. While b^k, D_a^k, and D_b^k vary at each iteration, the matrix M remains constant. Because of the special structure of this sequence of linear systems, we believe that it is possible to derive efficient methods for solving them—direct methods as well as iterative methods. However, this is not the subject of this paper, but it might be of interest for future research.

After the first revision of this paper, we received the report [26] of Tseng. In his §2, Tseng independently derives the main results (Theorems 4.1–4.3) of our paper. His interesting proofs, however, are different from the ones given here. Moreover, he shows that the conditions on the matrices M stated in our main results are, in some sense, not only sufficient, but also necessary.

Acknowledgments. The author would like to thank the referees for their helpful comments, which improved the original paper considerably. In particular, the author is very grateful to the referees for the shortened proof of Theorem 4.2 (which had originally been stated for positive definite matrices only) as well as for Theorem 4.3 itself, which generalized a result for strictly copositive matrices in a former version of this paper. Moreover, the author thanks Andreas Fischer for his suggestion to consider property (P.4) as well as Thomas Hitziger for his advice during the FORTRAN implementation of Algorithm 3.1.

REFERENCES

[1] R. W. COTTLE, J.-S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[2] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

[3] Y. G. EVTUSHENKO AND V. A. PURTOV, *Sufficient conditions for a minimum for nonlinear programming problems*, Soviet Math. Dokl., 30 (1984), pp. 313–316.

[4] Y. FATHI, *Computational complexity of LCPs associated with positive definite matrices*, Math. Programming (Series A), 17 (1979), pp. 335–344.

[5] M. C. FERRIS AND S. LUCIDI, *Globally Convergent Methods for Nonlinear Equations*, Tech. report 1030, Computer Sciences Department, University of Wisconsin, Madison, WI, July 1991. J. Optim. Theory Appl., 81 (1994), pp. 53–71.

[6] A. FISCHER, *A special Newton-type optimization method*, Optimization, 24 (1992), pp. 269–284.

[7] ———, *A Newton-type Method for Positive Semidefinite Linear Complementarity Problems*, Tech. report MATH-NM-04-1992, Institute of Numerical Mathematics, University of Technology Dresden, Dresden, Germany. J. Optim. Theory Appl., 86 (1995), pp. 585–608.

[8] ———, *On the Local Superlinear Convergence of a Newton-type Method for LCP Under Weak Conditions*, Tech. report MATH-NM-07-1993, Institute of Numerical Mathematics, University of Technology Dresden, Dresden, Germany, June 1993 (revised March 1994).

[9] L. GRIPPO, F. LAMPARIELLO, AND S. LUCIDI, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[10] P. T. HARKER AND J.-S. PANG, *A damped Newton method for the linear complementarity problem*, in Computational Solution of Nonlinear Systems of Equations, E. L. Allgower and K. Georg, eds., Lectures in Applied Mathematics, Vol. 26, American Mathematical Society, Providence, RI, 1990, pp. 265–284.

[11] P. T. HARKER AND B. XIAO, *Newton's method for the nonlinear complementarity problem: A B-differentiable equation approach*, Math. Programming (Series A), 48 (1990), pp. 339–357.

[12] W. HOCK AND K. SCHITTKOWSKI, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems 187, Springer-Verlag, Berlin, Germany, 1981.

[13] C. KANZOW, *Newton-type Methods for Nonlinearly Constrained Optimization*, Institute of Applied Mathematics, University of Hamburg, Hamburg, Germany, December 1992, preprint 62.

[14] ———, *Some equation-based methods for the nonlinear complementarity problem*, Optim. Meth. Software, 3 (1994), pp. 327–340.

[15] ———, *Nonlinear Complementarity as Unconstrained Optimization*, Institute of Applied Mathematics, University of Hamburg, Hamburg, Germany, March 1993 (revised February and August 1994), preprint 67, J. Optim. Theory Appl., to appear.

[16] C. KANZOW AND H. KLEINMICHEL, *A Class of Newton-type Methods for Equality and Inequality Constrained Optimization*, Institute of Applied Mathematics, University of Hamburg, Hamburg, Germany, November 1992 (revised March and July 1994), preprint 61, Optim. Meth. Software, 5 (1995), pp. 173–198.

[17] O. L. MANGASARIAN, *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM J. Appl. Math., 11 (1976), pp. 89–92.

[18] O. L. MANGASARIAN AND M. V. SOLODOV, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming (Series B), 62 (1993), pp. 277–297.

[19] K. G. MURTY, *Linear Complementarity, Linear and Nonlinear Programming*, Sigma Series in Applied Mathematics 3, Heldermann Verlag, Berlin, Germany, 1988.

[20] J.-S. PANG, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[21] ———, *A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems*, Math. Programming (Series A), 51 (1991), pp. 101–131.

[22] V. A. PURTOV, *Fast convergent nonlinear programming methods*, in Numerical Methods of Nonlinear Programming and their Implementations, C. Richter, H. Hollatz, and D. Pallaschke, eds., Akademie-Verlag, Berlin, Germany, 1991, pp. 87–98.

[23] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, New York, Heidelberg, Berlin, 1980.

[24] P. K. SUBRAMANIAN, *Gauss–Newton methods for the complementarity problem*, J. Optim. Theory Appl., 77 (1993), pp. 467–482.

[25] J. A. TOMLIN, *Users Guide for LCPL. A Program for Solving Linear Complementarity Problems by Lemke's Method*, Tech. report SOL 76–16, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, August 1976.

[26] P. TSENG, *Growth Behaviour of a Class of Merit Functions for the Nonlinear Complementarity Problem*, Tech. report, University of Washington, Seattle, WA, May 1994, J. Optim. Theory Appl., to appear.

[27] L. T. WATSON, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM J. Control Optim., 17 (1979), pp. 36–46.

[28] Y. ZHANG, R. A. TAPIA, AND F. POTRA, *On the superlinear convergence of interior-point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 342–361, May 1996
© 1996 Society for Industrial and Applied Mathematics
006

AN INTERIOR-POINT METHOD FOR SEMIDEFINITE PROGRAMMING*

CHRISTOPH HELMBERG†, FRANZ RENDL†, ROBERT J. VANDERBEI‡, AND HENRY WOLKOWICZ§

Abstract. We propose a new interior-point-based method to minimize a linear function of a matrix variable subject to linear equality and inequality constraints over the set of positive semidefinite matrices. We show that the approach is very efficient for graph bisection problems such as max–cut. Other applications include max–min eigenvalue problems and relaxations for the stable set problem.

Key words. semidefinite programming, interior-point methods, max–cut relaxations, max–min eigenvalue problems

AMS subject classifications. Primary 65F15; Secondary 49M35, 90C48

1. Introduction.

The continuously rising success of interior-point techniques applied to linear programming has stimulated research in various related fields. One possible line of generalization consists of looking at linear programs over nonpolyhedral cones. This type of generalization is studied in the present paper.

To be specific, let $\mathcal{M}_n$ denote the vector space of symmetric $n \times n$ matrices. Suppose $A : \mathcal{M}_n \mapsto \Re^k$, $B : \mathcal{M}_n \mapsto \Re^m$ are two linear operators and $C \in \mathcal{M}_n, a \in \Re^k, b \in \Re^m$.

We study the following optimization problem:

$$\text{(SDP)} \qquad \begin{array}{ll} \text{maximize} & \operatorname{tr} CX \\ \text{subject to} & a - A(X) = 0, \\ & b - B(X) \geq 0, \\ & X \succeq 0. \end{array}$$

This is a *semidefinite linear program* because we optimize a linear function subject to linear inequality and equality constraints over positive semidefinite matrices X. We note that the inequality constraints $b - B(X) \geq 0$ could be transformed into equality constraints by introducing slack variables. In this case X must be replaced by a direct sum of X and 1×1 matrices, corresponding to the slacks.

Our main motivation to study this kind of problem comes from applications in discrete optimization. In particular, we will investigate a new, powerful, and tractable relaxation of the *max–cut* problem, as well as other graph bisection problems. Also, *max–min eigenvalue problems* over an affine parameter space fall into the present framework and can be handled with our approach.

The main contribution of the paper is the following: we will propose a primal-dual interior-point algorithm for problem (SDP). Moreover, we will discuss implementation

* Received by the editors June 1, 1994; accepted for publication (in revised form) November 9, 1994.

† Technische Universität Graz, Institut für Mathematik, Kopernikusgasse 24, A-8010 Graz, Austria. The research of these authors was supported by Christian Doppler Laboratorium für Diskrete Optimierung.

‡ School of Engineering and Applied Science, Princeton University, Princeton, NJ 08544. The research of this author was supported by Air Force Office of Scientific Research grant AFOSR-91-0359.

§ Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada. The research of this author was supported by the National Science and Engineering Research Council Canada.

details and present some computational experiments indicating that the approach is also highly efficient in practice.

We close this section by describing research related to our work. Alizadeh, Haeberly, Overton, and Jarre [2, 3, 4, 14, 15, 21, 22] consider a problem similar to ours. Algorithmically, these authors use mostly interior-point-based techniques to solve the problem. Alizadeh proposes a potential reduction method and shows a polynomial running time to find an ϵ-optimal solution. Jarre uses a barrier approach and works directly on the dual. Finally, Overton [21, 22] studies the problem under nonlinear equality constraints. The formulations in [14, 21, 22, 4] are not in the form above, but it is easy to transform them into our model. Vandenberghe and Boyd study primal-dual potential reduction algorithms for semidefinite programs; see [27]. In [16], the monotone linear complementarity problem for symmetric matrices is investigated. The authors present several interior-point approaches to this type of semidefinite program. Finally, a general framework for interior-point methods applied to convex programs can be found in the monograph [20].

1.1. Preliminaries. We first collect some preliminary results and notation. We work mainly in the space $\mathcal{M}_n$ of symmetric $n \times n$ matrices, endowed with *inner product*

$$\langle U, V \rangle := \operatorname{tr}(UV^T).$$

The curly inequality symbol refers to the *Löwner partial order* induced by the cone of positive semidefinite matrices, i.e., $M_1 \preceq M_2$ ($M_1 \prec M_2$) means that $M_2 - M_1$ is positive semidefinite (positive definite, respectively). In contrast, the usual inequality symbol, $v \leq w$, refers to the partial order induced by the cone of nonnegative vectors. The *maximum eigenvalue* of M is denoted by $\lambda_{\max}(M)$. For two matrices $U = (u_{ij})$ and $V = (v_{ij})$ of the same size, $U \circ V$ denotes the Hadamard product, i.e., $(U \circ V)_{ij} = u_{ij} \cdot v_{ij}$.

Associated with any linear operator $A : \mathcal{M}_n \mapsto \Re^k$ is another linear operator, which we denote by A^T and which is defined by the adjoint relation

$$\langle A(X), y \rangle = \langle X, A^T(y) \rangle \quad \text{for all } X \in \mathcal{M}_n, y \in \Re^k.$$

Here we have used the angle bracket both for the inner product in $\mathcal{M}_n$ and in $\Re^k$.

For any convex cone S in R^n, we let S^* denote the *dual cone*, i.e.,

$$S^* = \{y \in \Re^n : x^T y \geq 0 \quad \text{for all } x \in S\}.$$

In a slight abuse of notation, given a real function f defined on a real domain and a vector v in $\Re^n$, we will occasionally write $f(v)$ for $(f(v_1), \ldots, f(v_n))$. Here, f will usually be the logarithm or a power function.

Finally, for X in $\mathcal{M}_n$, we let $\operatorname{diag}(X)$ denote the vector in $\Re^n$ consisting of the diagonal elements of X. Analogously, for a vector x in $\Re^n$, we let $\operatorname{Diag}(x)$ denote the diagonal matrix in $\mathcal{M}_n$ whose diagonal elements are obtained from x.

2. Duality. The general duality theory for problems such as (SDP) has been thoroughly studied; see, e.g., [30].

We derive the dual to (SDP) directly using Lagrangian methods. Indeed, let ω^* denote the optimal objective value for (SDP). Introducing Lagrange multipliers $y \in \Re^k$ and $t \in \Re^m_+$ for the equality and inequality constraints, respectively, we see

that

$$\begin{aligned}\omega^* &= \max_{X\succeq 0}\min_{t\geq 0,y} \quad \operatorname{tr} CX + y^T(a - A(X)) + t^T(b - B(X)) \\ &\leq \min_{t\geq 0,y}\max_{X\succeq 0} \quad \operatorname{tr}(C - A^T(y) - B^T(t))X + a^T y + b^T t.\end{aligned}$$

Now note that the inner maximization over X is bounded from above only if

$$A^T(y) + B^T(t) - C \succeq 0.$$

In this case the maximum occurs if complementarity holds, i.e.,

$$\operatorname{tr}(C - A^T(y) - B^T(t))X = 0.$$

Thus we get the (weak) dual (DSDP).

$$\text{(DSDP)} \qquad \begin{array}{ll} \text{minimize} & a^T y + b^T t \\ \text{subject to} & A^T(y) + B^T(t) - C \succeq 0, \\ & y \in \Re^k, \quad t \in \Re^m_+. \end{array}$$

The duality gap from the interchange of max and min vanishes only under additional assumptions on the Lagrangian. We tacitly assume that both problems have feasible solutions. If a constraint qualification holds, then it can be shown that both problems form a pair of dual problems and strong duality holds, i.e., the minimum attained in (SDP) coincides with the maximum attained in (DSDP); see, e.g., [30]. A sufficient condition for strong duality to hold is the existence of strictly feasible interior points for both the primal and the dual problems. (Weak duality, in which the max is less than or equal to the min, holds by construction.)

In our applications, we will focus mostly on the special case where $A(X) = \operatorname{diag}(X)$, in which case $A^T(y) = \operatorname{Diag}(y)$.

Before developing an algorithm for (SDP), we first show in the next section that this type of problem provides a strong machinery for deriving tight bounds to several basic NP-hard optimization problems.

3. Applications.

3.1. Max–cut problem. The max–cut problem is the problem of partitioning the node set of an edge-weighted undirected graph into two parts to maximize the total weight of edges *cut* by the partition. We tacitly assume that the graph in question is complete (if not, nonexisting edges can be given weight 0 to complete the graph). Mathematically, the problem can be formulated as follows; see, e.g., [19]. Let the graph be given by its weighted adjacency matrix A. Define the matrix $L := \operatorname{Diag}(Ae) - A$, where e is the vector of all 1's. (The matrix L is called the *Laplacian matrix* associated with the graph.) If a cut S is represented by a vector x, where $x_i \in \{-1, 1\}$ depending on whether or not $i \in S$, we get the following formulation for the max–cut problem:

$$\text{(MC)} \qquad \begin{array}{ll} \text{maximize} & \frac{1}{4}x^T L x \\ \text{subject to} & x \in \{-1, 1\}^n. \end{array}$$

Using $X := \frac{1}{4}xx^T$, this is equivalent to

$$\begin{array}{ll} \text{maximize} & \operatorname{tr} LX \\ \text{subject to} & \operatorname{diag}(X) = \dfrac{1}{4}e, \\ & \operatorname{rank}(X) = 1, \\ & X \succeq 0. \end{array}$$

Dropping the rank condition, we obtain a problem of the form (SDP) with no inequalities, $a = \frac{1}{4}e$, and $A(X) = \operatorname{diag}(X)$.

This relaxation of max–cut is well known and is studied, e.g., in [9, 11, 24]. Goemans and Williamson [11] have recently shown that the optimal value of this relaxation is at most 14% above the value of the maximum cut, provided $A \geq 0$, i.e., no negative edge weights exist.

The variable X can be interpreted as being defined on the edge set of the (complete) graph. Therefore, we can add further linear constraints that are satisfied by all edge vectors representing cuts. One such class of constraints is obtained by the following trivial observation. Consider an arbitrary triangle with vertices $i < j < k$ in the graph G. Then any partition cuts either zero or two of its edges. Translated into our model this leads to

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_{ij} \\ x_{ik} \\ x_{jk} \end{pmatrix} + \frac{1}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 0.$$

(Note that in our model an edge is cut if $x_i x_j = -1$; thus $x_{ij} = -\frac{1}{4}$.) The first constraint states that at most two edges can be cut. The other three constraints state that if one edge is cut, then there must be at least one other edge which must lie in the cut. Let us now collect these constraints for all triangles in the operator B. Then this leads to

$$B(X) - b \geq 0,$$

where $\dim(B) = 4\binom{n}{3}$ and b is a vector. Thus we get the following stronger relaxation for max–cut, which was first proposed in [24]:

$$\begin{array}{ll} \text{maximize} & \operatorname{tr} LX \\ \text{subject to} & \operatorname{diag}(X) - a = 0, \\ & B(X) - b \geq 0, \\ & X \succeq 0. \end{array}$$

Removing the semidefiniteness constraint on X, we obtain an ordinary linear program (LP), which is again a relaxation for max–cut. This relaxation is usually called the *metric relaxation*, because the polyhedron

$$\{X : B(X) - b \geq 0, \operatorname{diag}(X) = a\}$$

is often referred to as the *metric polytope*; see, e.g., [17]. We point out that this LP has $\binom{n}{2}$ variables and roughly $\frac{2}{3}n^3$ (very sparse) constraints. This polyhedron is highly degenerate, so it is still considered a computational challenge to optimize an arbitrary linear function over this polytope for, say, $n \approx 40$. (If the graph is planar, then the metric relaxation already provides the max–cut; see, e.g., [5]. See also [12].)

3.2. Graph bisection. Graph bisection is similar to the max–cut problem, but here we seek a partition (S,T) of the node set V such that the two sets have prespecified cardinalities, say, $|S| = k$ and $|T| = n-k$ for some given k. An important special case occurs if $k = \frac{n}{2}$. In this case, one looks to partition the node set into two sets of equal size to minimize the weight of the cut. (This problem is usually formulated as a minimization rather than a maximization problem.) The additional cardinality constraint $|S| = \frac{n}{2}$ translates into $\sum_i x_i = 0$. Thus we have

$$\text{(BS)} \qquad \begin{array}{ll} \text{minimize} & \frac{1}{4}x^T L x \\ \text{subject to} & x_i \in \{-1,1\}, \\ & e^T x = 0. \end{array}$$

Analogous to max–cut, we obtain the following relaxation:

$$\begin{array}{ll} \text{minimize} & \text{tr } LX \\ \text{subject to} & \text{diag}(X) = a, \\ & \text{tr } XJ = 0, \\ & X \succeq 0. \end{array}$$

Here, $J = ee^T$ is the matrix of all 1's. Note that the constraint tr $JX = 0$ is obtained by squaring the cardinality constraint: $0 = (e^T x)^2 = \text{tr } JX$. This relaxation was also studied in [10], where the authors treated it as a min–max eigenvalue problem using nonsmooth optimization techniques. A more theoretical investigation of this bound is given by Boppana [6].

3.3. Maximum cliques in graphs. Semidefinite programs are also used in conjunction with stable set and clique problems in graphs; see [18]. Suppose a graph G on n vertices is given by its edge set E. Define $E_{ij} := e_i e_j^t + e_j e_i^t$, where e_j is column j of the identity matrix I_n of size n. As above, $J = ee^t$ is the matrix of all 1's of size n. The following semidefinite program was introduced in [18] and provides an upper bound on the largest clique in G. The optimal objective function of this program is usually denoted by $\vartheta(G)$.

$$\begin{array}{lll} \text{maximize} & \text{tr } JX & \\ \text{subject to} & \text{tr } E_{ij}X = 0 & \forall (ij) \notin E, \\ & \text{tr } X = 1, & \\ & X \succeq 0. & \end{array}$$

Note that the number of equality constraints is $\binom{n}{2} - |E| + 1$. Therefore, even if n is small, the dimension of the dual may be quite large.

3.4. Min–max eigenvalue problems. The following min–max eigenvalue problem is studied, e.g., in [14, 21, 22]:

$$\lambda^* := \min_{y \in \Re^k} \lambda_{\max}(C + A(y)).$$

Here, $C \in \mathcal{M}_n$ and $A : \Re^k \mapsto \mathcal{M}_n$ is a linear operator. Overton [21, 22] actually considers a more general case that allows A to be nonlinear. It is well known that this problem can be reformulated as an (SDP) problem:

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \lambda I - C - A(y) \succeq 0, \\ & y \in \Re^k, \quad \lambda \in \Re. \end{array}$$

The dual is

$$\begin{array}{ll} \text{maximize} & \operatorname{tr} CX \\ \text{subject to} & A^T(X) = 0, \\ & \operatorname{tr} X = 1, \\ & X \succeq 0. \end{array}$$

Note that complementary slackness at optimality implies that $ZX = 0$, with $Z := \lambda I - C - A(y)$. Dual feasibility then implies that the eigenvectors for the optimal eigenvalue are found as the columns of X. If strict complementary slackness holds, i.e., $\operatorname{rank}(X)+\operatorname{rank}(Z) = n$, then $\operatorname{rank}(X)$ equals the multiplicity of the optimal eigenvalue.

Applications to other combinatorial optimization problems are described in [3].

4. An Interior-point method for SDP. In this section we will develop a primal-dual interior-point method that solves (SDP) and (DSDP) simultaneously. The nature of this approach requires the existence of an X that strictly satisfies the inequalities of the primal problem, i.e.,

$$b - B(X) > 0 \quad \text{and} \quad X \succ 0.$$

Furthermore, we assume without loss of generality that the equality constraints on X are linearly independent, $\operatorname{rank}(A(\cdot)) = k$. Since we will have to apply operators A and B to nonsymmetric matrices as well, we extend their definition by mapping the skew-symmetric part to zero. This implies

$$A(M) = A(M^T) \quad \text{and} \quad B(M) = B(M^T). \tag{4.1}$$

We follow the usual derivation of primal-dual interior-point methods in linear programming and first introduce the associated *barrier problem* for (DSDP), which we call the *dual barrier problem*:

$$\text{(DBP)} \qquad \begin{array}{ll} \text{minimize} & a^T y + b^T t - \mu(\log\det Z + e^T \log t) \\ \text{subject to} & A^T(y) + B^T(t) - C = Z, \\ & t \geq 0, Z \succeq 0. \end{array} \tag{4.2}$$

Here μ is a positive real number called the *barrier parameter*.

For each $\mu > 0$, there is a corresponding Lagrangian:

$$\begin{aligned} L_\mu(X, y, t, Z) = {} & a^T y + b^T t - \mu(\log\det Z + e^T \log t) \\ & + \left\langle Z + C - A^T(y) - B^T(t), X \right\rangle. \end{aligned} \tag{4.3}$$

The first-order optimality conditions for the saddle point of this Lagrangian are obtained easily using the adjoint identity for A and B:

$$\nabla_X L_\mu = Z + C - A^T(y) - B^T(t) = 0, \tag{4.4}$$

$$\nabla_y L_\mu = a - A(X) = 0, \tag{4.5}$$

$$\nabla_t L_\mu = b - B(X) - \mu t^{-1} = 0, \tag{4.6}$$

$$\nabla_Z L_\mu = X - \mu Z^{-1} = 0. \tag{4.7}$$

The strict concavity of $\log\det Z$ and $\log t_i$ implies that there exists a unique solution $(X_\mu, y_\mu, t_\mu, Z_\mu)$ to these optimality conditions. The one-parameter family

$\{(X_\mu, y_\mu, t_\mu, Z_\mu) : 0 \leq \mu \leq \infty\}$ is called the *central trajectory*. Given a point (X, y, t, Z) on the central trajectory, it is easy to determine its associated μ value using (4.6) and/or (4.7):

$$\mu = \frac{\operatorname{tr}(ZX)}{n} = \frac{t^T(b - B(X))}{m} = \frac{\operatorname{tr}(ZX) + t^T(b - B(X))}{n + m}. \tag{4.8}$$

We note that if the point is a feasible solution of the primal and the dual problems, then $\operatorname{tr}(ZX) + t^T(b - B(X))$ is the gap between the primal and the dual objective values. We shall use (4.8) to associate μ values with quadruples (X, y, t, Z) even when these quadruples don't belong to the central trajectory.

Our interior-point algorithm is derived as follows. We start with a quadruple (X, y, t, Z) for which $X \succ 0$, $Z \succ 0$, $t > 0$, and $b - B(X) > 0$, but which is otherwise arbitrary. From this point we estimate the current μ value using (4.8) and divide it by 2:

$$\mu = \frac{\operatorname{tr}(ZX) + t^T(b - B(X))}{2(n + m)}. \tag{4.9}$$

(Experience from linear programming indicates that this simple heuristic performs very well, even though it does not guarantee monotonic decrease in μ; see [28].) We next attempt to find directions $(\Delta X, \Delta y, \Delta t, \Delta Z)$ such that the new point $(X + \Delta X, y + \Delta y, t + \Delta t, Z + \Delta Z)$ lies on the central trajectory at this value of μ. However, since not all the defining equations, (4.4)–(4.7), are linear, it is not possible to solve this system directly. In fact, only (4.6) and (4.7) are nonlinear. They can be written in several equivalent forms, each form giving rise to a different linearization. We will give a more detailed discussion of the possibilities in the next section, although we restrict the discussion to (4.7). For our algorithm we use the linearization of the form

$$ZX - \mu I = 0.$$

For simplicity of notation, we rewrite (4.4) to (4.7) as the function

$$F_\mu(s) = F_\mu(X, y, t, Z) := \begin{pmatrix} Z + C - A^T(y) - B^T(t) \\ a - A(X) \\ t \circ (b - B(X)) - \mu e \\ ZX - \mu I \end{pmatrix} =: \begin{pmatrix} F_d \\ F_p \\ F_{tB} \\ F_{ZX}. \end{pmatrix}. \tag{4.10}$$

The solution s^* to $F_\mu(s) = 0$ satisfies the Karush–Kuhn–Tucker conditions (4.4) to (4.7) and is the optimal solution to the barrier problem. To find a direction $\Delta s = (\Delta X, \Delta y, \Delta t, \Delta Z)$ toward s^*, we use Newton's method, which says that Δs must satisfy

$$F_\mu + \nabla F_\mu(\Delta s) = 0.$$

The direction Δs is therefore the solution of the system

$$\Delta Z - A^T(\Delta y) - B^T(\Delta t) = -F_d, \tag{4.11}$$

$$-A(\Delta X) = -F_p, \tag{4.12}$$

$$\Delta t \circ (b - B(X)) - t \circ B(\Delta X) = -F_{tB}, \tag{4.13}$$

$$Z\Delta X + \Delta Z X = -F_{ZX}. \tag{4.14}$$

This linear system can now be solved for $(\Delta X, \Delta y, \Delta t, \Delta Z)$. Indeed, we first solve (4.11) for an obviously symmetric ΔZ (in terms of Δy and Δt),

$$\Delta Z = -F_d + A^T(\Delta y) + B^T(\Delta t), \tag{4.15}$$

and then substitute this expression into (4.14) to get

$$\Delta\tilde{X} = \mu Z^{-1} - X + Z^{-1}F_dX - Z^{-1}(A^T(\Delta y) + B^T(\Delta t))X. \tag{4.16}$$

Evidently, $\Delta\tilde{X}$ is not generally symmetric. Substituting this expression for ΔX into (4.12), we get the first equation for Δy and Δt:

$$O_{11}(\Delta y) + O_{12}(\Delta t) = v_1, \tag{4.17}$$

where O_{11} and O_{12} are the linear operators defined by

$$O_{11}(\cdot) := A(Z^{-1}A^T(\cdot)X), \tag{4.18}$$

$$O_{12}(\cdot) := A(Z^{-1}B^T(\cdot)X) \tag{4.19}$$

and v_1 is the vector

$$v_1 := \mu A(Z^{-1}) - a + A(Z^{-1}F_dX). \tag{4.20}$$

Finally we substitute (4.16) into (4.13) and get

$$O_{21}(\Delta y) + O_{22}(\Delta t) = v_2, \tag{4.21}$$

where O_{21} and O_{22} are the linear operators defined by

$$O_{21}(\cdot) := B(Z^{-1}A^T(\cdot)X), \tag{4.22}$$

$$O_{22}(\cdot) := (b - B(X)) \circ t^{-1} \circ (\cdot) + B(Z^{-1}B^T(\cdot)X) \tag{4.23}$$

and v_2 is the vector

$$v_2 := \mu t^{-1} - b + \mu B(Z^{-1}) + B(Z^{-1}F_dX). \tag{4.24}$$

Because of (4.1), operators O_{11} and O_{22} are self adjoint and operator O_{12} is the adjoint operator to O_{21}. Equations (4.17) and (4.21) form a symmetric linear system for Δy and Δt. In fact, this system is even positive definite. To show this, we define a new operator O that maps $\mathcal{M}_n$ into $\Re^{k+m}$:

$$O(X) = \begin{pmatrix} A(X) \\ B(X) \end{pmatrix}.$$

The adjoint operator is given by the adjoint identity

$$\left\langle O(X), \begin{pmatrix} y \\ t \end{pmatrix} \right\rangle = \left\langle X, A^T(y) + B^T(t) \right\rangle.$$

The system can now be written as

$$O\left(Z^{-1}O^T\begin{pmatrix} \Delta y \\ \Delta t \end{pmatrix} X\right) + \begin{pmatrix} 0 \\ (b - B(X)) \circ t^{-1} \circ (\Delta t) \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{4.25}$$

Observe that the second summand adds positive coefficients to the main diagonal entries corresponding to the inequality constraints. In fact, this increment on the main diagonal is the only difference between inequality and equality constraints. The second summand clearly forms a positive semidefinite operator on $\binom{\Delta y}{\Delta t}$. It is positive definite for all vectors having at least one component $\Delta t_i \neq 0$. For the first summand we get

$$\langle O(Z^{-1}O^T(v)X), v\rangle \tag{4.26}$$

$$= \langle Z^{-1}O^T(v)X, O^T(v)\rangle \tag{4.27}$$

$$= \operatorname{tr}\left(Z^{-\frac{1}{2}}O^T(v)X^{\frac{1}{2}}X^{\frac{1}{2}}O^T(v)Z^{-\frac{1}{2}}\right) \tag{4.28}$$

$$= \left\langle Z^{-\frac{1}{2}}O^T(v)X^{\frac{1}{2}}, Z^{-\frac{1}{2}}O^T(v)X^{\frac{1}{2}}\right\rangle \geq 0. \tag{4.29}$$

Since X and Z^{-1} are positive definite and the equality constraints are linearly independent, equality is possible if and only if $v = 0$ or at least one $\Delta t_i \neq 0$. It follows that the system is indeed positive definite. It can be solved efficiently in $O((k+m)^3)$. Observe that an equivalent representation of operator $O(X)$ is given by $(\operatorname{tr}(A_1X), \ldots, \operatorname{tr}(A_{k+m}X))^T$ with the A_i being appropriately chosen symmetric matrices. Then the ijth element of the matrix describing $O(Z^{-1}O^T(\cdot)X)$ reads $\operatorname{tr}(A_iZ^{-1}A_jX)$.

The solution yields a quadruple with $\Delta\tilde{X}$ that is not necessarily symmetric. This is mended by using only the symmetric part of $\Delta\tilde{X}$:

$$\Delta X = \frac{\Delta\tilde{X}^T + \Delta\tilde{X}}{2}. \tag{4.30}$$

To summarize, we solve for the quadruple $\Delta s = (\Delta X, \Delta y, \Delta t, \Delta Z)$ by first solving (4.17) and (4.21) for Δy and Δt and then substituting this into (4.15) to solve for ΔZ and finally substituting that into (4.14) to solve for $\Delta\tilde{X}$, of which we take only the symmetric part. In §5 we will show that this indeed yields a descent direction.

Having determined the desired quadruple $(\Delta X, \Delta y, \Delta t, \Delta Z)$ of directions, we would step to the new quadruple $(X + \Delta X, y + \Delta y, t + \Delta t, Z + \Delta Z)$, except that it might violate the nonnegativity of t and $b - B(X)$ and the positive definiteness property required of the two matrices. Hence, we perform a line search to find constants α_p and α_d such that $t + \alpha_d\Delta t$ and $b - B(X + \alpha_p\Delta X)$ are strictly positive and $X + \alpha_p\Delta X$ and $Z + \alpha_d\Delta Z$ are positive definite. Given α_p and α_d, we step to the new point

$$\begin{aligned} X + \alpha_p\Delta X,\\ y + \alpha_d\Delta y,\\ t + \alpha_d\Delta t,\\ Z + \alpha_d\Delta Z. \end{aligned}$$

We update μ using (4.9) and repeat. The algorithm continues until the current quadruple (X, y, t, Z) satisfies primal feasibility and dual feasibility and the duality gap is sufficiently small. This completes the description of our interior-point algorithm.

5. Descent direction. We have found a modified Newton direction for the optimality conditions. To shed some light on the quality of this direction $(\Delta X, \Delta y, \Delta t, \Delta Z)$,

we will prove that it forms a descent direction with respect to an appropriately defined merit function. We measure the progress of the algorithm using the following merit function. (This type of merit function was also used in [1]. See also [29].)

$$\begin{aligned} f_\mu(X,y,t,Z) &= \langle Z, X\rangle - \mu \log\det(XZ) \\ &\quad + t^T(b - B(X)) - \mu e^T \log(t \circ (b - B(X))) \\ &\quad + \frac{1}{2}||F_p||^2 + \frac{1}{2}||F_d||^2. \end{aligned} \tag{5.1}$$

For feasible points, the merit function is the difference between the objective values of the dual and primal barrier functions. Therefore f_μ is convex over the set of feasible points. Because the minimum of $(x - \mu \log x)$ for $x > 0$ is attained at $x = \mu$, function f_μ is bounded below by $(n+m)\mu(1 - \log\mu)$. We note that $F_\mu(s) = 0 \iff f_\mu = (n+m)\mu(1-\log\mu)$. f_μ is continuously differentiable on the interior ($X \succ 0$, $t > 0$, $Z \succ 0$) and grows toward infinity on the boundary. In Lemma 1 we prove that Δs (as defined earlier) is a descent direction for f_μ.

LEMMA 1. *The directional derivative of f_μ in the direction Δs satisfies*

$$\langle \nabla_s f_\mu, \Delta s\rangle \le 0 \tag{5.2}$$

with equality holding if and only if $F_\mu(s) = 0$.

Proof. We must prove that

$$\langle \nabla_X f_\mu, \Delta X\rangle + \langle \nabla_y f_\mu, \Delta y\rangle + \langle \nabla_t f_\mu, \Delta t\rangle + \langle \nabla_Z f_\mu, \Delta Z\rangle \le 0.$$

The partial derivatives of f_μ are

$$\begin{aligned} \nabla_X f_\mu &= F_p^T \nabla_X F_p + F_d^T \nabla_X F_d + Z - \mu X^{-1} - B^T(t) - \mu B^T((b - B(X))^{-1}), \\ \nabla_y f_\mu &= F_p^T \nabla_y F_p + F_d^T \nabla_y F_d, \\ \nabla_t f_\mu &= F_p^T \nabla_t F_p + F_d^T \nabla_t F_d + b - B(X) - \mu t^{-1}, \\ \nabla_Z f_\mu &= F_p^T \nabla_Z F_p + F_d^T \nabla_Z F_d + X - \mu Z^{-1}. \end{aligned}$$

Summing the directional derivative over all partial derivatives of F_p, we get

$$\begin{aligned} &\langle F_p^T \nabla_X F_p, \Delta X\rangle + \langle F_p^T \nabla_y F_p, \Delta y\rangle + \langle F_p^T \nabla_t F_p, \Delta t\rangle + \langle F_p^T \nabla_Z F_p, \Delta Z\rangle \\ &\quad = \langle F_p, -A(\Delta X)\rangle \\ &\quad = \langle F_p, -A(\Delta \tilde{X})\rangle \\ &\quad = -||F_p||^2, \end{aligned}$$

where the third line follows from $A(M) = A(M^T)$ and the fourth from (4.12). Analogously, we get

$$\langle F_d^T \nabla_X F_d, \Delta X\rangle + \langle F_d^T \nabla_y F_d, \Delta y\rangle + \langle F_d^T \nabla_t F_d, \Delta t\rangle + \langle F_d^T \nabla_Z F_d, \Delta Z\rangle = -||F_d||^2.$$

For the next step we observe that for a symmetric matrix S, $\langle S, \Delta X\rangle = \langle S, \Delta \tilde{X}\rangle$ since the skew-symmetric part of $\Delta \tilde{X}$ is orthogonal to symmetric matrices.

$$\begin{aligned} &\langle Z - \mu X^{-1}, \Delta X\rangle + \langle X - \mu Z^{-1}, \Delta Z\rangle \\ &= \langle Z - \mu X^{-1}, \Delta \tilde{X}\rangle + \langle X - \mu Z^{-1}, \Delta Z\rangle \\ &= \operatorname{tr}(I - \mu X^{-1} Z^{-1})(Z \Delta X' + \Delta Z X) \end{aligned}$$

$$
\begin{aligned}
&= -\mathrm{tr}(I - \mu X^{-1}Z^{-1})(ZX - \mu I) \\
&= -\mathrm{tr}(ZX - 2\mu I + \mu^2 (ZX)^{-1}) \\
&= -\sum_{i=1}^{n} \lambda_i(ZX)\left(1 - \mu\lambda_i^{-1}(ZX)\right)^2 .
\end{aligned}
$$

Because Z and X are positive definite, all eigenvalues of ZX are strictly positive. The last expression is less than or equal to zero, and it equals zero if and only if $ZX = \mu I$.

In the same manner we get

$$
\begin{aligned}
&(b - B(X) - \mu t^{-1})^T \Delta t + (t - \mu(b - B(X))^{-1})^T B(\Delta X) \\
&= -(t \circ (b - B(X)))^T (e - \mu t^{-1} \circ (b - B(X))^{-1})^2 .
\end{aligned}
$$

Again, $t \circ (b - B(X))$ is strictly positive, so the expression is less than or equal to zero, and it equals zero if and only if $t \circ (b - B(X)) = \mu e$.

Summing up we have

$$
\begin{aligned}
\frac{\partial f_\mu}{\partial s}\Delta s = &-||F_p||^2 - ||F_d||^2 - \sum_{i=1}^{n} \lambda_i(ZX)\left(1 - \mu\lambda_i^{-1}(ZX)\right)^2 \\
&-(t \circ (b - B(X)))^T (e - \mu t^{-1} \circ (b - B(X))^{-1})^2 \le 0
\end{aligned} \tag{5.3}
$$

with equality holding if and only if $F_\mu(s) = 0$ holds. □

Inaccurate line search with respect to the merit function and a feasible starting point is sufficient to guarantee convergence; an elementary proof can be found in [13]. This need not be true for arbitrarily chosen infeasible starting points. Some computational evidence for this claim is contained in [4].

Strong results on the rate of convergence can be obtained by tightly restricting the choice of μ and the stepsize. In an independent paper, Kojima, Shindoh, and Hara [16] propose the very same algorithmical framework for the semidefinite complementarity problem. Within this framework they give several variants of algorithms leading to polynomial convergence. The first method is close to our setting and we will give some details. A complementarity problem asks for a point $(X, Z) \in \mathcal{M}_n^2$ which lies within a given $n(n+1)/2$ dimensional affine subspace $\mathcal{F}$ of $\mathcal{M}_n^2$ and satisfies the complementarity condition $\mathrm{tr}(ZX) = 0$, i.e., find a pair $(Z, X) \in \mathcal{M}_n^2$ such that

$$
\text{(LCP)} \qquad (X, Z) \in \mathcal{F},\ X \succeq 0,\ Z \succeq 0,\ \text{and } \mathrm{tr}(XZ) = 0.
$$

Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of XZ. By

$$
\mathcal{N}(\gamma) = \left\{ (X, Z) \in \mathcal{F}, X \succ 0, Z \succ 0 \,\middle|\, \left(\sum_{i=1}^{n} (\lambda_i - \mu)^2 \right)^{\frac{1}{2}} \le \gamma\mu \ \text{ with } \ \mu = \frac{\mathrm{tr}(ZX)}{n} \right\},
$$

a horn neighborhood of the central trajectory is introduced. The width of this neighborhood is controlled by γ. For $\gamma \in (0, 0.1]$ this neighborhood has a convenient property (Theorem 4.1 in [16]): by reducing μ by a factor of $(1 - \frac{\gamma}{\sqrt{n}})$ the "Newton step" $(X + \Delta X, Z + \Delta Z)$ is again in $\mathcal{N}(\gamma)$. Furthermore, the new μ-value of the new point is smaller than the old value of μ by at least a factor of $(1 - \frac{\gamma}{2\sqrt{n}})$. So if $(X^0, Z^0) \in \mathcal{N}(\gamma)$, the algorithm yields a sequence $(X^j, Z^j) \in \mathcal{N}(\gamma)$ such that for every $j = 0, 1, \ldots$

$$
\mathrm{tr}(X^j, Z^j) \le \left(1 - \frac{\gamma}{2\sqrt{n}}\right)^j \mathrm{tr}(X^0 Z^0).
$$

Thus, if we require $\operatorname{tr}(X^j Z^j) < \varepsilon$ for a given $\varepsilon > 0$, then the algorithm will stop in $O(\sqrt{n} \log \frac{\operatorname{tr}(X^0 Z^0)}{\varepsilon})$ iterations. In practical applications the strict requirements for the choice of μ must be violated for the benefit of efficiency.

We will now discuss the linearization stemming from equivalent formulations of the condition $ZX - \mu I = 0$. For ease of notation we will restrict our investigation to the equality constraints and leave out the inequality operator B. We have the following possibilities.

$$\mu I - Z^{\frac{1}{2}} X Z^{\frac{1}{2}} = 0, \tag{5.4}$$

$$\mu I - X^{\frac{1}{2}} Z X^{\frac{1}{2}} = 0, \tag{5.5}$$

$$\mu Z^{-1} - X = 0, \tag{5.6}$$

$$\mu X^{-1} - Z = 0, \tag{5.7}$$

$$ZX - \mu I = 0, \tag{5.8}$$

$$XZ - \mu I = 0, \tag{5.9}$$

$$ZX + XZ - 2\mu I = 0. \tag{5.10}$$

The first two linearizations are popular in linear programming, in which case X and Z are diagonal matrices. The two forms are then the same and do not really involve matrix square roots. However, in the semidefinite programming case they do involve matrix square roots, which makes them computationally less attractive.

The linearization of the third form (5.6) leads to Jarre's algorithm [14]. In this case the defining equation for Δy reads

$$A(Z^{-1} A^T(\Delta y) Z^{-1}) = A(Z^{-1} + Z^{-1} F_d Z^{-1}) - \frac{a}{\mu}.$$

Note that no information about the primal variable is contained in this equation. This is a pure dual approach. The resulting direction for the primal variable is poor. This was confirmed by practical experiments.

Analogously, the linearization of (5.7) leads to

$$A(X A^T(\Delta y) X) = A(\mu X + XCX - X A^T(y) X) - \mu F_p.$$

This formulation is used in [25]. This time the step is based mainly on the primal variable. We can consider this linearization as a pure primal approach.

The linearization of (5.8) is the choice of this paper. It is easy to see that (5.9) results in the same step. Indeed, $\Delta \tilde{X}$ of the latter linearization is just the transpose of $\Delta \tilde{X}$ of our algorithm. In both cases the direction found contains the information of the primal and dual variables equally. Both linearizations are especially well suited for Mehrotra's LP predictor–corrector method as described in [7].

The latter two statements also apply to the linearization of (5.10) as discussed in [4]. This linearization is advantageous because it preserves symmetry. Furthermore, Alizadeh, Haeberly, and Overton present some evidence of good numerical properties. However, because the system is more difficult to solve, computation time per iteration can be considerably higher.

6. Computational results.

6.1. Interior-point approach for max–cut. In this section we show how the max–cut relaxation given in §3 can be efficiently implemented within this framework. We first look at the relaxation without the triangle inequalities. The resulting program is simple and can be quickly solved for quite large n; see Table 6.1.

The cost matrix is given by the Laplacian L of the weighted adjacency matrix of the graph. The SDP forming the relaxation reads

$$\begin{array}{rl} \text{maximize} & \operatorname{tr} LX \\ \text{subject to} & \operatorname{diag}(X) = \dfrac{1}{4}e =: a, \\ & X \succeq 0. \end{array} \tag{6.1}$$

Since Diag is the adjoint of diag, the dual SDP reads

$$\begin{array}{rl} \text{minimize} & a^T y \\ \text{subject to} & \operatorname{Diag}(y) - L = Z, \\ & Z \succeq 0. \end{array} \tag{6.2}$$

The starting point

$$X := \operatorname{Diag}(a), \tag{6.3}$$

$$y := 1.1 \cdot \operatorname{abs}(L)e, \tag{6.4}$$

$$Z := \operatorname{Diag}(y) - L \tag{6.5}$$

is feasible and in the interior. If we look at the defining equation for Δy we get

$$O_{11}(\Delta y) = \operatorname{diag}(Z^{-1}\operatorname{Diag}(\Delta y)X) = (Z^{-1} \circ X)\Delta y, \tag{6.6}$$

$$v_1 = \mu \operatorname{diag}(Z^{-1}) - a. \tag{6.7}$$

This means that at each iteration we solve the system by computing

$$\Delta y := (Z^{-1} \circ X)^{-1}(\mu \operatorname{diag}(Z^{-1}) - a), \tag{6.8}$$

$$\Delta Z := \operatorname{Diag}(\Delta y), \tag{6.9}$$

$$\Delta \tilde{X} := \mu Z^{-1} - X - Z^{-1}\Delta Z X, \tag{6.10}$$

$$\Delta X := \frac{(\Delta \tilde{X}^T + \Delta \tilde{X})}{2}. \tag{6.11}$$

To further emphasize the simplicity and ease of implementation of this approach, we include in the next section a MATLAB function that solves this relaxation. This is a very efficient code for problems of sizes up to $n \approx 200$ and the reader is encouraged to test it.

The numerical results of Table 6.1 were computed on a 486, 66 Mhz PC using a C encoded version of this algorithm. We point out that the number of iterations does not seem to depend significantly on n. In all tables *hh* gives the number of hours, *mm* the number of minutes, and *ss* the seconds. The instances were generated as random unweighted graphs with edge probability $p = 0.5$. We set the stopping condition to six digits of accuracy, as in the MATLAB routine in §7.

As a second experiment, we looked at the max–cut relaxation formed by the intersection of all the $4\binom{n}{3}$ triangle inequalities with the semidefiniteness constraints;

TABLE 6.1
SDP relaxation for max-cut, computed on a 486, 66 Mhz PC.

size n	iterations	hh:mm:ss
100	14	48
150	12	2:30
200	12	6:30
250	13	14:24
300	14	30:24
400	14	1:24:36
500	14	2:13.18

see §3.1. When we included triangle constraints it was favorable to use Mehrotra's predictor–corrector approach, which is closely investigated in [7]. In this case, the system is solved twice with the same factorization but with different right-hand sides. First, the right-hand side is chosen to minimize the objective, i.e., $\mu = 0$ is used. The second step computes a good centering direction for the predicted point using an estimate of μ from the previous iteration. The triangle inequalities were included successively according to the amount of their violation; for a detailed description of the algorithm the reader is referred to [13]. Some results for random graphs are given in Table 6.2. Most of the time the best cut was already found by the relaxation without triangles and could not be improved later on. It was constructed from the rounded rows of X giving an initial cut. We then applied a local improvement strategy to this cut by considering swapping any single node from one side to the other side of the cut. If any such swap improved the cut, we chose the swap with highest gain. We stopped if no further improvement could be obtained this way.

We note that these semidefinite programs have n equality and $4\binom{n}{3}$ inequality constraints in the dual program. The final solution satisfies all these constraints. The number of iterations necessary to obtain these results is astonishingly small; see the last column of Table 6.2. To be explicit, the solution of problems with $n = 100$ satisfies $4\binom{100}{3} = 646,800$ inequality constraints. Our approach does not seem to depend on structural properties of the underlying graph. It significantly extends purely polyhedral techniques for max–cut. Further and more substantial computational results with this approach applied to max–cut can be found in [13].

6.2. The ϑ-function. In this section we give computational results for the ϑ-function, as defined in §3.3. For the convenience of the reader we give the corresponding primal-dual pair of semidefinite programs,

$$\begin{array}{lll} \text{maximize} & \operatorname{tr} JX & \\ \text{subject to} & \operatorname{tr} E_{ij}X = 0 & \forall (ij) \notin E, \\ & \operatorname{tr} X = 1, & \\ & X \succeq 0. & \end{array} \qquad \begin{array}{lll} \text{minimize} & \lambda & \\ \text{subject to} & Z = \lambda I - J + \sum_{(ij)\notin E} E_{ij} y_{ij}, & \\ & Z \succeq 0,\ y_{ij} \in \Re & \forall (ij) \notin E. \end{array}$$

As the starting point we used the feasible interior-point

$$\begin{aligned} X &:= \frac{1}{n} I, \\ \lambda &:= 1.1 \cdot n, \\ y_{ij} &:= 0 \quad \forall (ij) \notin E, \\ Z &:= \lambda I - J. \end{aligned}$$

TABLE 6.2

Solutions to max–cut relaxation with triangle inequalities. The number of nodes in the graph is n, cut refers to the best cut found, upper bnd is the value of the relaxation, and % is the percentage of the gap between relaxation and the best cut found. The next column gives the computation times on a DEC-station 5000/240. *Note that the number of iterations is almost independent of n if the gap is not closed.*

random graphs, edge-weights $\in \{-10, \ldots, 10\}$				
size n	cut	upper bnd (%)	hh:mm:ss	iterations
30	330	330.0001 (0.0)	54	59
40	619	619.0000 (0.0)	2:02	60
50	708	708.0001 (0.0)	1:04:22	166
70	1215	1248.7142 (2.8)	4:42:21	220
100	2440	2540.2528 (4.1)	37:56:05	266
random unweighted graphs, edge probability 0.5				
size n	cut	upper bnd (%)	hh:mm:ss	iterations
30	143	143.0000 (0.0)	56	62
40	245	245.6864 (0.3)	59:19	247
50	374	375.3689 (0.4)	1:54:18	250
70	723	726.0399 (0.4)	5:57:22	257

We tested the code for random graphs of the class $G_{n;p}$, i.e., graphs with n vertices and each edge (ij) being in the set E with probability p. We include the source code, written in C++, that we used to generate these pseudorandom graphs so that the reader can reproduce them. To generate the adjacency matrix A, we used the following random number generator with parameters n, *seed*, and *prob*.

```
double r=(4.*double(seed)+1.)/16384./16384.;
for(int i=0;i<n;i++){
    for(int j=i+1;j<n;j++) {
        r=fmod(r*41475557.,1.);
        if (r<1.-prob) A(i,j)=1.;
        else A(i,j)=0.;
    }
}
```

The results for several choices of *seed*, *prob*, and n are displayed in Table 6.3. The code employed the predictor–corrector approach using separate primal and dual stepsizes. The program was written in C++ and executed on a DEC-station 5000/240. The number of primal constraints or dual variables is the number of *nonedges* plus 1. Computing times strongly depend on the latter number but are almost independent of n as long as *nonedges* $\gg n$. Again, the number of iterations seems to be quite independent of both n and the number of constraints.

6.3. Min–max eigenvalue problems. Consider the min–max eigenvalue problem:

$$\begin{aligned} &\text{minimize} && \lambda_{\max}(C - \mathrm{Diag}(v)) \\ &\text{subject to} && e^T v = 0. \end{aligned} \tag{6.12}$$

The objective function in (6.12) is not differentiable when the multiplicity of the largest eigenvalue exceeds 1. In fact, a singleton eigenvalue characterizes differentiability. Since the largest eigenvalue is a convex function, subgradient approaches can

TABLE 6.3
The ϑ-function for random graphs.

seed	prob	size n	nonedges	ϑ	hh.mm.ss	iterations
1	0.5	50	593	7.9233	7:00	9
2	0.8	50	237	16.0012	42	14
3	0.9	50	124	21.0910	10	11
4	0.8	100	1018	21.9283	44:05	10
5	0.9	100	511	32.4967	5:29	10
6	0.9	150	1130	41.6814	1:04:23	10
7	0.95	150	574	56.4224	10:56	11
8	0.95	200	972	70.5405	44:46	10
9	0.97	200	585	85.0430	17:39	12
10	0.97	250	915	98.5259	51:10	11
11	0.98	250	604	114.6005	25:43	11
12	0.97	300	1310	112.4511	2:30:46	11

be used to solve (6.12) (see, e.g., [8]). More recently, it has been shown that Newton-based algorithms with local quadratic convergence exist (see, e.g., [23]), but the local convergence depends on the correct identification of the multiplicity of the largest eigenvalue. We present computational experiments which show that our interior-point method is indeed robust in the presence of high multiplicity.

By minimizing λ over the set $\{\lambda I \succeq C - \mathrm{Diag}(v), e^T v = 0\}$ and substituting $y := v + \lambda e$ we see that (6.12) is indeed a special case (with $a = \frac{e}{n}$) of the more general problem

$$\begin{array}{ll} \text{minimize} & a^T y \\ \text{subject to} & \mathrm{Diag}(y) - C \succeq 0. \end{array} \tag{6.13}$$

The MATLAB code given in §7 also applies to this problem. To test the code on problem instances that exhibit given multiplicity at the optimal solution, we developed a special generator which we now describe.

To generate positive semidefinite programs, we generate the elements of a uniformly on some interval of the nonnegative half-line (the primal problem is clearly infeasible if any component of a is negative). For the experiments described below, we used $a = \frac{e}{n}$.

Given a, we generate C as follows. First, we generate an $n \times m$ random matrix A and apply row scaling to make all squared row norms equal to the corresponding elements of a, that is,

$$\mathrm{diag}(AA^T) = a. \tag{6.14}$$

We denote the columns of A by $v_1, \ldots, v_m$. We then construct $n-m$ additional random n-vectors $v_{m+1}, \ldots, v_n$ and apply Gram–Schmidt orthonormalization to $v_1, \ldots, v_n$ to produce an $n \times n$ orthogonal matrix Q whose first m columns span the same space as $v_1, \ldots, v_m$. Finally, we set

$$C = Q \Lambda Q^T, \tag{6.15}$$

where Λ is a diagonal matrix whose first $k \geq m$ entries are all set to $\lambda_{\max}$ (which is a constant that can be chosen arbitrarily—we used 5) and the remaining diagonal entries are generated uniformly on some interval that is strictly smaller than $\lambda_{\max}$.

TABLE 6.4
Statistics for problems with multiplicity 1. *BT refers to the bundle trust method and IP refers to our interior-point method.*

			Time (hh:mm:ss.s)	
n	m	k	BT	IP
10	1	1	0.0	0.0
20	1	1	0.1	0.3
30	1	1	0.3	1.0
50	1	1	1.2	4.2
100	1	1	12.6	35.6
200	1	1	1:56.9	6:17.4

TABLE 6.5
Statistics for problems with higher built-in multiplicity. BT refers to the bundle trust method and IP refers to our interior-point method.

			Time (hh:mm:ss.s)		Comments
n	m	k	BT	IP	
20	3	3	0.4	0.4	
20	5	5	2.8	0.3	
20	5	12	2.8	0.3	
20	8	8	2.7	0.3	
20	12	12	3.6	0.3	
30	3	3	1.5	1.0	
30	3	6	1.5	0.9	
30	6	6	18.2	0.8	
30	10	10	4.0	1.0	Num. trouble in BT
50	5	5	>20:00.0	4.3	5 sig. fig. in BT
100	3	3	18.7	33.9	
100	6	6	>15:00.0	36.9	5 sig. fig. in BT
500	50	50	-	2:02:47.0	No attempt at BT

For such a matrix C, we claim that

$$X = AA^T,$$
$$y = \lambda_{\max} e,$$
$$Z = \mathrm{Diag}(y) - C$$

is optimal. Indeed, it follows from (6.14) that X is feasible for the primal and it is clear from (6.15) that (y, Z) is feasible for the dual. Finally, optimality follows from the absence of a duality gap:

$$\mathrm{tr}(ZX) = \mathrm{tr}\{(\lambda_{\max} I - C)AA^T\} = 0.$$

The last equality follows from the fact that the columns of A are eigenvectors of C associated with the maximal eigenvalue.

Table 6.4 shows the comparison between the bundle trust method (see [26]) and our interior-point method when the optimal eigenvalue is a singleton ($k = 1$). For these problems, the bundle trust method is three to four times faster (computing times are given for a Silicon Graphics Indigo workstation R 4000). However, this situation never arises in practice. Indeed, for $k = 1$ in our construction above, we see that we are requiring the vector of all 1's to be a maximal eigenvector of C. This is clearly an unlikely event in real applications.

Table 6.5 shows comparisons for higher multiplicities. Here the results look much better for the interior-point method. In fact, it is clear that the bundle trust method completely breaks down rather rapidly as the multiplicity increases.

7. Matlab implementation. The following MATLAB function solves the semidefinite programming problem described earlier with $A(X) = \mathrm{diag}(X)$ and no inequality constraints. We include it to further emphasize the simplicity of our interior-point approach. We assume that MATLAB's version 4 is available in the positive definiteness test and use MATLAB's built-in function `chol`. The program could also be run under older versions of MATLAB. We have somewhat arbitrarily set the stopping condition to six digits of accuracy. Thus, after successful termination, we return primal and dual feasible solutions, whose objective values agree to approximately six digits.

In all our practical experiments it proved unnecessary to check the decrease of the merit function. Therefore, this feature is not included in the present algorithm. We also mention that in case of "large steps," i.e., $\alpha_p + \alpha_d \geq 1.8$, we select the new value of μ to be $\mathrm{tr}XZ/(4n)$, and otherwise $\mathrm{tr}XZ/(2n)$.

```
function [phi, X, y] = psd_ip( L);
% solves: max trace(LX) s.t. X psd, diag(X) = b;  b = ones(n,1)/4
%         min b'y       s.t. Diag(y) - L psd, y unconstrained,
% input:  L ... symmetric matrix
% output: phi ... optimal value of primal, phi =trace(LX)
%         X   ... optimal primal matrix
%         y   ... optimal dual vector
% call:   [phi, X, y] = psd_ip( L);

digits = 6;                     % 6 significant digits of phi
[n, n1] = size( L);             % problem size
b = ones( n,1 ) / 4;            % any b>0 works just as well
X = diag( b);                   % initial primal matrix is pos. def.
y = sum( abs( L))' * 1.1;       % initial y is chosen so that
Z = diag( y) - L;               % initial dual slack Z is pos. def.
phi = b'*y;                     % initial dual
psi = L(:)' * X( :);            % and primal costs
mu = Z( :)' * X( :)/( 2*n);     % initial complementarity
iter=0;                         % iteration count

disp(['      iter    alphap    alphad      gap      lower      upper']);

while phi-psi > max([1,abs(phi)]) * 10^(-digits)

      iter = iter + 1;          % start a new iteration
      Zi = inv( Z);             % inv(Z) is needed explicitly
      Zi = (Zi + Zi')/2;
      dy =  (Zi.*X) \ (mu * diag(Zi) - b);      % solve for dy
      dX = - Zi * diag( dy) * X + mu * Zi - X;  % back substitute for dX
      dX = ( dX + dX')/2;       % symmetrize

% line search on primal
      alphap = 1;               % initial steplength
      [dummy,posdef] = chol( X + alphap * dX );  % test if pos.def
      while posdef > 0,
              alphap = alphap * .8;
              [dummy,posdef] = chol( X + alphap * dX );
              end;
      if alphap < 1, alphap = alphap * .95; end; % stay away from boundary
% line search on dual; dZ is handled implicitly: dZ = diag( dy);
      alphad = 1;
      [dummy,posdef] = chol( Z + alphad * diag(dy) );
      while posdef > 0;
              alphad = alphad * .8;
              [dummy,posdef] = chol( Z + alphad * diag(dy) );
              end;
      if alphad < 1, alphad = alphad * .95; end;
% update
      X = X + alphap * dX;
      y = y + alphad * dy;
```

```
      Z = Z + alphad * diag(dy);
      mu = X( :)' * Z( :) / (2*n);
      if alphap + alphad > 1.8, mu = mu/2; end; % speed up for long steps
      phi = b' * y; psi = L( :)' * X( :);
% display current iteration
        disp([ iter alphap alphad (phi-psi) psi  phi ]);

        end;            % end of main loop
```

Acknowledgments. We thank the anonymous referees and the associate editor for their constructive comments on earlier versions of this paper. Their critical remarks helped to improve this paper. This author also thanks the Department of Civil Engineering and Operations Research, Princeton University, for their support during his stay while on research leave.

REFERENCES

[1] K. M. ANSTREICHER AND J.-P. VIAL, *On the convergence of an infeasible primal-dual interior-point method for convex programming*, Optimization Methods and Software, 3 (1994), pp. 273–283.

[2] F. ALIZADEH, *Combinatorial Optimization with Interior Point Methods and Semidefinite Matrices*, Ph.D. Thesis, University of Minnesota, Minneapolis, MN, 1991.

[3] ———, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[4] F. ALIZADEH, J. A. HAEBERLY, AND M. L. OVERTON, *Primal-dual interior point methods for semidefinite programming*, Tech. report, Courant Institute of Mathematical Sciences, New York University, New York, 1994.

[5] F. BARAHONA, *The max–cut problem in graphs not contractible to K_5*, Oper. Res. Lett., 2 (1983), pp. 107–111.

[6] R. B. BOPPANA, *Eigenvalues and graph bisection: An average case analysis*, in Proc. 28th Annual Symposium on Foundations of Computer Science, Los Angeles, CA, IEEE, 1987, pp. 280–285.

[7] T. J. CARPENTER, I. J. LUSTIG, J. M. MULVEY, AND D. F. SHANNO, *Higher-order predictor–corrector interior point methods with application to quadratic objectives*, SIAM J. Optim., 3 (1993), pp. 696–725.

[8] J. CULLUM, W. E. DONATH, AND P. WOLFE, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Study, 3 (1975), pp. 35–55.

[9] C. DELORME AND S. POLJAK, *Laplacian eigenvalues and the maximum cut problem*, Math. Programming, 62 (1993), pp. 557–574.

[10] J. FALKNER, F. RENDL, AND H. WOLKOWICZ, *A computational study of graph partitioning*, Math. Programming, 66 (1994), pp. 211–240.

[11] M. X. GOEMANS AND D. P. WILLIAMSON, *.878-approximation algorithm for max–cut and max–2sat*, Tech. report, MIT, Cambridge, MA, 1993.

[12] F. O. HADLOCK, *Finding a maximum cut in a planar graph in polynomial time*, SIAM J. Comput., 4 (1975), pp. 221–225.

[13] C. HELMBERG, *An interior point method for semidefinite programming and max–cut bounds*, Ph.D. Thesis, Graz University of Technology, Graz, Austria, Oct. 1994.

[14] F. JARRE, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[15] ———*Interior-point methods via self-concordance or relative Lipschitz condition*, Habilitationsschrift (thesis), Würzburg University, March 1994.

[16] M. KOJIMA, S. SHINDOH, AND S. HARA, *Interior-point Methods for the Monotone Linear Complementarity Problem in Symmetric Matrices*, Tech. report B-282, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Apr. 1994.

[17] M. LAURENT AND S. POLJAK, *The metric polytope*, in Proc. IPCO 2, E. Balas, G. Cornuejols, and R. Kannan, eds., Carnegie–Mellon University, Pittsburgh, PA, 1992, pp. 274–286.

[18] L. LOVÁSZ, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[19] B. MOHAR AND S. POLJAK, *Eigenvalues in combinatorial optimization*, in Combinatorial and

graph-theoretic problems in linear algebra, R. Brualdi, S. Friedland, and V. Klee, eds., IMA Volumes in Mathematics and Its Applications, Vol. 50, Springer-Verlag, Berlin, New York, 1993.

[20] Y. E. NESTEROV AND A. S. NEMIROVSKII, *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1993.

[21] M. L. OVERTON, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.

[22] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[23] M. L. OVERTON AND R. S. WOMERSLEY, *Second derivatives for optimizing eigenvalues of symmetric matrices*, Tech. report 627, Computer Science Department, New York University, New York, 1993, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 697–718.

[24] S. POLJAK AND F. RENDL, *Nonpolyhedral relaxations of graph-bisection problems*, Tech. report, Graz University of Technology, 1992, SIAM J. Optim., 5 (1995), pp. 467–487.

[25] F. RENDL, R. J. VANDERBEI, AND H. WOLKOWICZ, *Max–min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems*, Optimization Methods and Software, 5 (1995), pp. 1–16.

[26] H. SCHRAMM AND J. ZOWE, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim., 2 (1993), pp. 121–152.

[27] L. VANDENBERGHE AND S. BOYD, *Primal-dual potential reduction method for problems involving matrix inequalities*, Math. Programming, 69 (1995), pp. 205–236.

[28] R. J. VANDERBEI AND T. J. CARPENTER, *Symmetric indefinite systems for interior-point methods*, Math. Programming, 58 (1993), pp. 1–32.

[29] J.-P. VIAL, *Computational experience with a primal-dual interior-point method for smooth convex programming*, Optimization Methods and Software, 3 (1994), pp. 285–316.

[30] H. WOLKOWICZ, *Some applications of optimization in matrix theory*, Linear Algebra Appl., 40 (1981), pp. 101–118.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 362–372, May 1996
© 1996 Society for Industrial and Applied Mathematics
007

COMPLETE CHARACTERIZATIONS OF GLOBAL OPTIMALITY FOR PROBLEMS INVOLVING THE POINTWISE MINIMUM OF SUBLINEAR FUNCTIONS*

B. M. GLOVER†, Y. ISHIZUKA‡, V. JEYAKUMAR§, AND H. D. TUAN¶

Abstract. Necessary and sufficient global optimality conditions are presented for certain nonconvex minimization problems subject to inequality constraints that are expressed as the pointwise minimum of sublinear (MSL) functions. A generalized Farkas lemma for inequality systems with MSL functions plays a crucial role in presenting the conditions in dual forms. Applications to certain multiplicative sublinear programming problems and fractional programming problems are also given.

Key words. complete characterization, global optimality, generalized Farkas lemma, ε-subdifferential

AMS subject classifications. 90C30, 49B27, 90C48

1. Introduction.

Global nonconvex optimization problems are intrinsically difficult to solve in general because of the occurrence of multiple local optima, and necessary optimality conditions are usually of limited value. Nevertheless, one class of problems which seems computationally tractable and has abundant applications (see Pardalos and Rosen [12] and Horst and Tuy [8, 9]) is the global maximization of convex or difference of convex functions subject to linear constraints. From the theoretical viewpoint Hiriart-Urruty [4] and Hiriart-Urruty and Lemarechal [6, 7] have provided a starting point for a detailed investigation of necessary and sufficient global optimality conditions for the maximization of convex functions utilizing the ε-subdifferential. On the other hand, global optimization problems with nonlinear constraints such as multiplicative programming problems also arise in many situations [11, 15]. The use of this ε-subdifferential set and the tools of nonsmooth analysis enables the study of global optimality conditions for a large class of nonconvex global optimization problems with nonlinear constraints. It appears however that very little work has been carried out on the development of necessary and sufficient global optimality conditions for the maximization of convex functions subject to nonlinear constraints.

The aim of this work is to establish complete characterizations of global optimality for optimization problems, including convex maximization problems, with inequality constraints that can be expressed as the pointwise minimum of sublinear (MSL) functions. The class of MSL functions includes the classes of sublinear functions and, more importantly, difference sublinear functions and products of certain sublinear functions. The optimality conditions are presented in dual forms, that is, in terms of subgradients and ε-subdifferentials. This is achieved by applying a lemma of Farkas type for inequality systems involving MSL functions. This together with the ε-subdifferential characterization of optimality (see [4, 15]) paves the way to deriving necessary and sufficient global optimality conditions for the general constrained optimization problems. Moreover, the method of employing the Farkas lemma provides an approach to the derivation of global optimality conditions that parallels the devel-

* Received by the editors September 17, 1993; accepted for publication (in revised form) October 26, 1994.

† School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria, Australia.

‡ Department of Mechanical Engineering, Sophia University, Tokyo, Japan.

§ School of Mathematics, University of New South Wales, Sydney 2052, Australia.

¶ Institute of Mathematics, Bo Ho, Hanoi, Vietnam.

opment of local optimality conditions for the traditional mathematical programming problems (cf. Jeyakumar and Glover [10]). We also apply this lemma to obtain necessary and sufficient optimality conditions for certain classes of nonlinear multiplicative programming problems and fractional programming problems.

2. Inequality systems of pointwise minimum of sublinear functions. In this section, based on a Farkas-type lemma for sublinear systems, we derive an extension of the Farkas lemma to systems involving functions expressible as the pointwise minimum of sublinear (MSL) functions (here called MSL systems). In what follows X is a Banach space. For a set $D \subset X$ we shall denote the *closure* and *convex hull* of D by cl D and co D, respectively. The *cone generated by* D is denoted cone $D = \cup_{\alpha \geq 0} \alpha D$. Throughout the course of this paper we shall refer to the *closed convex cone* generated by a set D, namely, cl cone co D.

A function $p : X \to \boldsymbol{R} \cup \{+\infty\}$ is said to be *sublinear* if it is proper, convex, and positively homogeneous of degree one (i.e., $p(\alpha x) = \alpha p(x)$ for all $\alpha \geq 0$ and $x \in X$ and $p(x+y) \leq p(x)+p(y)$ for all $x, y \in X$). If p is lower semicontinuous at 0 [16] then the convex subdifferential $\partial p(0) = \{v \in X' : (\forall x \in X) \;\; v(x) \leq p(x)\}$ is a nonempty weak* closed subset of X', the continuous dual space of X. Here p coincides with the support function of $\partial p(0)$, namely, for each $x \in X$, $p(x) = \sigma(x \mid \partial p(0))$, where $\sigma(\cdot \mid A)$ denotes the support function of a set $A \subset X'$, i.e., $\sigma(x \mid A) = \sup_{a \in A} \; a(x)$. In addition, if p is continuous, $\partial p(0)$ is weak* compact.

Recall that for $\varepsilon \geq 0$, the *ε-subdifferential* at $a \in X$ of a function $f : X \to \boldsymbol{R}$ is defined as

$$\partial_\varepsilon f(a) = \{v \in X' : (\forall x \in X) \;\; f(x) - f(a) \geq v(x-a) - \varepsilon\}.$$

If f is a convex function then $\partial_\varepsilon f(a)$ is a nonempty weak* closed convex set. See Hiriart-Urruty and Lemarechal [7] for a detailed discussion of this set and its properties and Rockafellar [13] for aspects of convex analysis. Note that $\cap_{\varepsilon > 0} \partial_\varepsilon f(a) = \partial f(a)$, the latter set denoting the usual convex subdifferential of f at a.

DEFINITION 2.1. *We shall say that a function $\ell : X \to \boldsymbol{R}$ is difference sublinear (DSL) if there exist nonempty weak* compact and convex sets $\underline{\partial}\ell(0)$ and $\overline{\partial}\ell(0)$ in X' such that for each $x \in X$,*

$$\begin{aligned} \ell(x) &= \max_{v \in \underline{\partial}\ell(0)} v(x) - \max_{w \in \overline{\partial}\ell(0)} w(x) \\ &= \sigma(x \mid \underline{\partial}\ell(0)) - \sigma(x \mid \overline{\partial}\ell(0)). \end{aligned}$$

The DSL function ℓ possesses a family of *upper sublinear approximations*, since for any $w \in \overline{\partial}\ell(0)$ we have $\ell(x) \leq \sigma(x \mid \underline{\partial}\ell(0)) - w(x)$ for each $x \in X$. The DSL function ℓ can be expressed as

$$\ell(x) = \min_{w \in \overline{\partial}\ell(0)} \; \max_{u \in \underline{\partial}\ell(0) - \{w\}} u(x). \tag{1}$$

It is worth observing that every DSL function is locally Lipschitz.

DEFINITION 2.2. *A function $k : X \to \boldsymbol{R}$ is said to be MSL (pointwise minimum of sublinear functions) if there exist an index set T and a family $\{\partial(k,t) : t \in T\}$ of nonempty and weak* closed sets in X' such that for each $x \in X$,*

$$\begin{aligned} k(x) &= \min_{t \in T} \; \sup_{v \in \partial(k,t)} v(x) \\ &= \min_{t \in T} \; \sigma(x \mid \partial(k,t)). \end{aligned} \tag{2}$$

From (1) and (2) it follows that every DSL function is an MSL function. Conversely, an MSL function may not be locally Lipschitz, and hence it may not, in general, be DSL. To see this, consider the following function $k : \boldsymbol{R}^2 \to \boldsymbol{R}$:

$$k(x,y) = \left(\sqrt{|x|} + \sqrt{|y|}\right)^2.$$

The function k is continuous and positively homogeneous. But k is not locally Lipschitz, and hence is not DSL. However k is MSL since

$$\begin{aligned} k(x,y) &= \min_{t\in[0,1]} \left\{ \frac{|x|}{t} + \frac{|y|}{1-t} \right\} \\ &= \min_{t\in[0,1]} \sup_{(u,v)\in\partial(k,t)} (ux+vy), \end{aligned}$$

where

$$\partial(k,t) = \begin{cases} \frac{1}{t}[-1,1] \times \frac{1}{1-t}[-1,1] & \text{if } t \in (0,1), \\ \boldsymbol{R} \times [-1,1] & \text{if } t = 0, \\ [-1,1] \times \boldsymbol{R} & \text{if } t = 1. \end{cases}$$

In what follows we shall assume that I is an arbitrary (possibly infinite) index set, $h_0,\ h_i : X \to \boldsymbol{R}, i \in I$ are DSL functions, and $g_0,\ g_i : X \to \boldsymbol{R}, i \in I$ are MSL functions such that there exist families

$$\{\partial(g_i,t_i) : t_i \in T_i\},\ i \in \{0\} \cup I$$

of nonempty and weak* closed sets in X', with

$$\begin{aligned} (\forall x \in X)\ \ g_i(x) &= \min_{t_i\in T_i} \sup_{v_i\in\partial(g_i,t_i)} v_i(x) \\ &= \min_{t_i\in T_i} \sigma(x \,|\, \partial(g_i,t_i)),\ i \in \{0\} \cup I. \end{aligned} \tag{3}$$

To derive a generalized Farkas lemma for MSL inequality systems, we introduce some notation. Let $\mathcal{T}$ denote the set of all selections (τ_i) such that $\tau_i \in T_i$ for $i \in I$, that is, a selection (τ_i) is a set consisting of elements $\tau_i, i \in I$, each of which is selected from $T_i, i \in I$. For the MSL inequality system

$$g_i(x) \le b_i,\ i \in I,$$

we define

$$\mathcal{T}_F = \{(\tau_i) \in \mathcal{T} : (\exists x \in X)\ \ \sigma(x \,|\, \partial(g_i,\tau_i)) \le b_i,\ i \in I\}. \tag{4}$$

Similarly, for the DSL inequality system

$$h_i(x) \le b_i,\ i \in I,$$

we define $\mathcal{W}$ to be the set of all selections (w_i) with $w_i \in \overline{\partial} h_i(0)$ for $i \in I$, and

$$\mathcal{W}_F = \{(w_i) \in \mathcal{W} : (\exists x \in X)\ \ \sigma(x \,|\, \underline{\partial} h_i(0)) - w_i(x) \le b_i,\ i \in I\}. \tag{5}$$

We now establish an extension of the Farkas lemma to MSL systems. To present a simplified proof, we first state the following nonhomogeneous version of the Farkas

lemma for *sublinear* inequality systems which can be deduced from the sublinear Farkas lemma for convex cones (see [2, 3]).

THEOREM 2.1 (generalized Farkas lemma for sublinear systems). *Let $\{0\} \cup I$ be an arbitrary index set. Assume that for each $i \in \{0\} \cup I$, $p_i : X \to \boldsymbol{R} \cup \{+\infty\}$ is a sublinear function which is lower semicontinuous at 0, $b_i \in \boldsymbol{R}, i \in \{0\} \cup I$ and that the system*

$$p_i(x) \leq b_i, \ i \in I, \tag{6}$$

is consistent. Then the following statements are equivalent:

(i) $p_i(x) \leq b_i, i \in I \ \Rightarrow \ p_0(x) \geq b_0$.

(ii) $$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \text{cl} \left\{ \begin{pmatrix} \partial p_0(0) \\ b_0 \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \geq 0}} \begin{pmatrix} \partial p_i(0) \\ b_i + r \end{pmatrix} \right\}.$$

It is worth noting that if, for instance, p_0 is continuous at 0 then $\partial p_0(0)$ is weak* compact, and the set in Theorem 2.1,

$$\begin{pmatrix} \partial p_0(0) \\ b_0 \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \geq 0}} \begin{pmatrix} \partial p_i(0) \\ b_i + r \end{pmatrix},$$

is weak* closed. In this case statement (ii) becomes

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial p_0(0) \\ b_0 \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \geq 0}} \begin{pmatrix} \partial p_i(0) \\ b_i + r \end{pmatrix}.$$

A generalized Farkas lemma for MSL systems can be derived from Theorem 2.1 as follows.

THEOREM 2.2 (generalized Farkas lemma for MSL systems). *Let $\{0\} \cup I$ be an arbitrary index set. Assume that for each $i \in \{0\} \cup I$, $g_i : X \to \boldsymbol{R}$ is an MSL function with* (3) *being satisfied, $b_i \in \boldsymbol{R}, i \in \{0\} \cup 0I$ and that the system*

$$g_i(x) \leq b_i, \ i \in I,$$

is consistent. Then the following statements are equivalent:

(i) $g_i(x) \leq b_i, i \in I \ \Rightarrow \ g_0(x) \geq b_0$.

(ii) *For each $\tau_0 \in T_0$ and for each selection $(\tau_i) \in \mathcal{T}_F$,*

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \text{cl} \left\{ \text{co} \begin{pmatrix} \partial(g_0, \tau_0) \\ b_0 \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \geq 0}} \begin{pmatrix} \partial(g_i, \tau_i) \\ b_i + r \end{pmatrix} \right\},$$

where $\mathcal{T}_F$ is defined by (4).

Proof. Statement (i) is clearly equivalent to the following statement.

For each $\tau_0 \in T_0$ and for each $(\tau_i) \in \mathcal{T}_F$,

$$\sigma(x \,|\, \partial(g_i, \tau_i)) \leq b_i, \ i \in I \ \Rightarrow \ \sigma(x \,|\, \partial(g_0, \tau_0)) \geq b_0. \tag{7}$$

Since $\sigma(\cdot \mid \partial(g_i, \tau_i))$ are sublinear and lower semicontinuous at 0 and

$$(\forall x \in X) \quad \sigma(x \mid \partial(g_0, \tau_0)) = \sigma(x \mid \operatorname{co} \partial(g_0, \tau_0)),$$

it follows from Theorem 2.1 that statement (7) is equivalent to condition (ii). □

As a direct consequence of Theorem 2.1, one can obtain the following generalized Farkas lemma for DSL functions extending the corresponding recent results in [3].

COROLLARY 2.1 (generalized Farkas lemma for DSL systems). *Let $\{0\} \cup I$ be an arbitrary index set. Assume that $h_i : X \to \boldsymbol{R}, i \in \{0\} \cup I$ are DSL functions, $b_i \in \boldsymbol{R}, i \in I$, and the system*

$$h_i(x) \le b_i, \ i \in I,$$

is consistent. Then the following statements are equivalent:

(i) $h_i(x) \le b_i, i \in I \ \Rightarrow \ h_0(x) \ge b_0$.

(ii) *For each selection $(w_i) \in \mathcal{W}_F$,*

$$\begin{pmatrix} \overline{\partial} h_0(0) \\ 0 \end{pmatrix} \subseteq \begin{pmatrix} \underline{\partial} h_0(0) \\ b_0 \end{pmatrix} + \operatorname{cl} \operatorname{cone} \operatorname{co} \bigcup_{\substack{i \in I \\ r \ge 0}} \begin{pmatrix} \underline{\partial} h_i(0) - w_i \\ b_i + r \end{pmatrix},$$

where $\mathcal{W}_F$ is defined by (5).

Proof. From the definition of DSL functions, we have

$$\begin{aligned} h_i(x) &= \max_{v_i \in \underline{\partial} h_i(0)} v_i(x) - \max_{w_i \in \overline{\partial} h_i(0)} w_i(x) \\ &= \min_{w_i \in \overline{\partial} h_i(0)} \max_{v_i \in \underline{\partial} h_i(0) - \{w_i\}} v_i(x) \\ &= \min_{t_i \in T_i} \max_{v_i \in \partial(h_i, t_i)} v_i(x), \quad i \in \{0\} \cup I, \end{aligned}$$

where

$$T_i = \overline{\partial} h_i(0), \quad \partial(h_i, t_i) = \underline{\partial} h_i(0) - \{t_i\}.$$

So, in this case,

$$\begin{aligned} \mathcal{T}_F &= \{(\tau_i) \in \mathcal{T} : (\exists x \in X) \ \ \sigma(x \mid \partial(h_i, \tau_i)) \le b_i, \ i \in I\} \\ &= \{(w_i) \in \mathcal{W} : (\exists x \in X) \ \ \sigma(x \mid \underline{\partial} h_i(0)) - w_i(x) \le b_i, \ i \in I\} \\ &= \mathcal{W}_F. \end{aligned}$$

Since $\overline{\partial} h_0(0)$ is a weak* compact and convex set, the conclusion follows from Theorem 2.2. □

A Farkas lemma for DSL systems similar to Corollary 2.1 was established in [3] under a "strong consistency" assumption which requires that

$$(\exists x^o \in X) \ (\forall i \in I) \ \ \sigma(x^o \mid \underline{\partial} h_i(0)) + \sigma(-x^o \mid \overline{\partial} h_i(0)) \le b_i,$$

or equivalently,

$$\mathcal{W}_F = \mathcal{W}.$$

So, Corollary 2.1, requiring only the consistency of the given inequality system (i.e., $\mathcal{W}_F \neq \emptyset$), provides a stronger result than in [3].

3. Complete characterizations of global optimality for a class of difference convex problems with MSL constraints. In this section, we apply the generalized Farkas lemma of the previous section with the use of ε-subdifferentials to establish complete characterizations of global optimality for a class of difference convex and fractional programming problems subject to MSL inequality constraints. First we consider the following difference of convex (D.C.) minimization model problem,

$$(P_1) \qquad \begin{array}{ll} \text{global minimize} & p(x) - f(x) \\ \text{subject to} & g_i(x) \le b_i,\ i \in I, \end{array}$$

where $p : X \to \boldsymbol{R}$ is a continuous sublinear function, $f : X \to \boldsymbol{R}$ is a continuous convex function, and $g_i : X \to \boldsymbol{R}, i \in I$ are MSL functions. It is worth observing that this model problem (P_1) covers convex maximization problems [10, 1] and DSL minimization problems [3]. The feasible set of (P_1) is given by

$$C = \{x \in X : g_i(x) \le b_i,\ i \in I\}.$$

THEOREM 3.1 (global minimization with MSL constraints). *Let $p : X \to \boldsymbol{R}$ be a continuous sublinear function, $f : X \to \boldsymbol{R}$ be a continuous convex function, I be an arbitrary index set, $g_i : X \to \boldsymbol{R}, i \in I$ be MSL functions with* (3) *being satisfied, and let $b_i \in \boldsymbol{R}, i \in I$. Then $a \in C$ is a global minimum of (P_1) if and only if for each selection $(\tau_i) \in \mathcal{T}_F$, $\varepsilon \ge 0$ and $u \in \partial_\varepsilon f(a)$,*

$$\begin{pmatrix} u \\ u(a) + \varepsilon \end{pmatrix} \in \begin{pmatrix} \partial p(0) \\ p(a) \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \ge 0}} \begin{pmatrix} \partial(g_i, \tau_i) \\ b_i + r \end{pmatrix}.$$

Proof. The point $a \in C$ solves (P_1) if and only if a solves the following problem:

$$\text{minimize}\ \ p(x) - f(x) + \delta(x \mid C).$$

From Hiriart-Urruty [4] and Tuy [15], the necessary and sufficient condition for $a \in C$ to be a global minimum of this problem is

$$(\forall \varepsilon \ge 0)\ \ \partial_\varepsilon f(a) \subseteq \partial_\varepsilon F(a), \tag{8}$$

where $F(x) = p(x) + \delta(x \mid C)$. We now show a characterization of $\partial_\varepsilon F(a)$ by using the generalized Farkas lemma. Noting that $a \in C$, we have $u \in \partial_\varepsilon F(a)$ if and only if

$$(\forall x \in X)\ \ p(x) + \delta(x \mid C) \ge p(a) + u(x - a) - \varepsilon.$$

Clearly this is equivalent to the statement

$$x \in C \ \Rightarrow\ (p - u)(x) \ge p(a) - u(a) - \varepsilon.$$

Thus, from Theorem 2.2, this is equivalent to the following.

For every $(\tau_i) \in \mathcal{T}_F$,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial p(0) - u \\ p(a) - u(a) - \varepsilon \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \ge 0}} \begin{pmatrix} \partial(g_i, \tau_i) \\ b_i + r \end{pmatrix}.$$

Thus $u \in \partial_\varepsilon F(a)$ if and only if for every $(\tau_i) \in \mathcal{T}_F$,

$$\begin{pmatrix} u \\ u(a) + \varepsilon \end{pmatrix} \in \begin{pmatrix} \partial p(0) \\ p(a) \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i \in I \\ r \ge 0}} \begin{pmatrix} \partial(g_i, \tau_i) \\ b_i + r \end{pmatrix}.$$

Thus from (8) the conclusion follows. □

As usual, under some constraint qualifications, one can obtain "nonasymptotic" optimality conditions. For example, if $X = \boldsymbol{R}^n$ and I is a finite set, and if the following constraint qualification is satisfied, then the closure operation in the condition of Theorem 3.1 is superfluous.

For every $(\tau_i) \in \mathcal{T}_F$, $\partial(g_i, \tau_i)$, $i \in I$ are nonempty compact sets of $\boldsymbol{R}^n$, and

$$0 \notin \operatorname{co} \bigcup_{i \in I} \partial(g_i, \tau_i),$$

or equivalently,

$$(\exists x^o \in X) \;\; \sigma(x^o \mid \partial(g_i, \tau_i)) < 0, \;\; i \in I.$$

Next we consider the following MSL optimization problem:

$$(P_2) \qquad \begin{array}{ll} \text{global minimize} & g_0(x) \\ \text{subject to} & g_i(x) \leq b_i, \;\; i \in I, \end{array}$$

where $g_i : X \to \boldsymbol{R}, i \in \{0\} \cup I$ are MSL functions. The feasible set is again

$$C = \{x \in X : g_i(x) \leq b_i, \;\; i \in I\}.$$

THEOREM 3.2 (MSL optimization problem). *Let $\{0\} \cup I$ be an arbitrary index set, $g_i : X \to \boldsymbol{R}, i \in \{0\} \cup I$ be MSL functions with (3) being satisfied, and let $b_i \in \boldsymbol{R}, i \in I$. Then $a \in C$ is a global minimum of (P_2) if and only if for each $\tau_0 \in T_0$ and for each selection $(\tau_i) \in \mathcal{T}_F$,*

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \operatorname{cl} \left\{ \operatorname{co} \begin{pmatrix} \partial(g_0, \tau_0) \\ g_0(a) \end{pmatrix} + \operatorname{cl} \operatorname{cone} \operatorname{co} \bigcup_{\substack{i \in I \\ r \geq 0}} \begin{pmatrix} \partial(g_i, \tau_i) \\ b_i + r \end{pmatrix} \right\}.$$

Proof. The point $a \in C$ solves (P_2) if and only if

$$x \in C \Rightarrow \;\; g_0(x) \geq g_0(a).$$

Thus the conclusion follows from Theorem 2.2. □

As an application of Theorem 3.1, we consider a complete characterization of optimality for a fractional programming problem with MSL inequality constraints.

Example 3.1 (fractional programming with MSL constraints). Consider the fractional programming problem

$$(P_3) \qquad \begin{array}{ll} \text{global minimize} & \dfrac{p(x)}{f(x)} \\ \text{subject to} & g_i(x) \leq b_i, \;\; i \in I, \end{array}$$

where $p : X \to \boldsymbol{R}$ is a continuous sublinear function, $f : X \to \boldsymbol{R}$ is a continuous convex function, I is an arbitrary index set, $g_i : X \to \boldsymbol{R}, i \in I$ are MSL functions with (3) being satisfied and $b_i \in \boldsymbol{R}, i \in I$. Let C denote the feasible region of (P_3). We shall assume that $f(x) > 0$ and $p(x) \geq 0$ for all $x \in C$. Hence, $a \in C$ is a global minimum of (P_3) if and only if

$$x \in C \;\; \Rightarrow \;\; \frac{p(a)}{f(a)} \leq \frac{p(x)}{f(x)},$$

that is,

$$x \in C \;\Rightarrow\; f(a)p(x) - p(a)f(x) \geq 0.$$

This is equivalent to $a \in C$ minimizing $f(a)p(x) - p(a)f(x)$ over C. Thus we can apply Theorem 3.1 assuming the consistency of the constraint system to obtain the following condition characterizing optimality for (P_3): For every selection $(\tau_i) \in \mathcal{T}_F$, $\varepsilon \geq 0$ and $u \in \partial_\varepsilon(\beta f)(a)$,

$$\begin{pmatrix} u \\ u(a)+\varepsilon \end{pmatrix} \in \alpha \begin{pmatrix} \partial p(0) \\ \beta \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i\in I \\ r\geq 0}} \begin{pmatrix} \partial(g_i,\tau_i) \\ b_i + r \end{pmatrix}, \tag{9}$$

where $\alpha = f(a)$, $\beta = p(a)$. When $p(a) = \beta > 0$, this condition becomes the following: For every $(\tau_i) \in \mathcal{T}_F$, $\varepsilon \geq 0$ and $u \in \partial_\varepsilon f(a)$,

$$\begin{pmatrix} u \\ u(a)+\varepsilon \end{pmatrix} \in \frac{f(a)}{p(a)} \begin{pmatrix} \partial p(0) \\ p(a) \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i\in I \\ r\geq 0}} \begin{pmatrix} \partial(g_i,\tau_i) \\ b_i + r \end{pmatrix}.$$

If $p(a) = \beta = 0$ then (9) reduces to the following: For every $(\tau_i) \in \mathcal{T}_F$,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial p(0) \\ 0 \end{pmatrix} + \text{cl cone co} \bigcup_{\substack{i\in I \\ r\geq 0}} \begin{pmatrix} \partial(g_i,\tau_i) \\ b_i + r \end{pmatrix}.$$

4. Applications to multiplicative programming. In this section, as consequences of the Farkas lemma (Theorem 2.2) and Theorem 3.2, we present a complete characterization of global optimality for a class of multiplicative convex programming problems and a solvability theorem for an MSL inequality system.

In some practical optimization problems, the product of two convex (or linear) functions appears as an objective and/or as a constraint function. As mentioned in the introduction, we shall consider a special case of such problems—using the product of two sublinear functions. In general, the product of two convex functions $f_1(x)f_2(x)$ has a D.C. representation—$2f_1f_2 = (f_1+f_2)^2 - (f_1^2+f_2^2)$—and it plays an important role in solving multiplicative convex programs. However, thanks to the Farkas lemma for MSL functions and the problem transformation technique, developed in [14, 11], we are able to obtain necessary and sufficient optimality conditions without such a representation. We consider the following problem:

$$(P_4) \qquad \begin{array}{ll} \text{global minimize} & p(x)q(x) \\ \text{subject to} & Ax \leq b, \end{array}$$

where $p, q : \boldsymbol{R}^n \to \boldsymbol{R}$ are continuous sublinear functions and A is an $m \times n$ matrix. The constraint set is given by

$$C = \{x \in \boldsymbol{R}^n : \; Ax \leq b\}.$$

THEOREM 4.1 (multiplicative sublinear programming). *Let $p, q : \boldsymbol{R}^n \to \boldsymbol{R}$ be continuous sublinear functions, A be an $m \times n$ matrix, and let $b \in \boldsymbol{R}^m$. Assume that $p(x) \geq 0, q(x) \geq 0$ for all $x \in \boldsymbol{R}^n$ and $p(x)q(x) = 0$ only if $p(x) = q(x) = 0$. Then $a \in C$ is a global minimum of (P_4) if and only if for each $t > 0$, there exists a $\lambda \geq 0$ such that*

$$0 \in t\partial p(0) + \frac{1}{t}\partial q(0) + A^T\lambda,$$

$$0 \geq \sqrt{p(a)q(a)} + \lambda^T b.$$

Proof. Clearly, $a \in C$ solves (P_4) if and only if $a \in C$ solves the following problem:

$$(P_4') \qquad \begin{array}{ll} \text{minimize} & 2\sqrt{p(x)q(x)} \\ \text{subject to} & Ax \leq b. \end{array}$$

Noting that for any $\alpha, \beta > 0$,

$$2\sqrt{\alpha\beta} = \min_{t>0} \left(t\alpha + \frac{\beta}{t} \right)$$

we can rewrite (P_4') as

$$(P_4'') \qquad \begin{array}{ll} \text{minimize} & \min\limits_{t>0} \left\{ tp(x) + \frac{1}{t} q(x) \right\} \\ \text{subject to} & Ax \leq b. \end{array}$$

Note that

$$f(x) := \min_{t>0} \left\{ tp(x) + \frac{1}{t} q(x) \right\} = \min_{t>0} \max_{v \in \partial(f,t)} v^T x,$$

where

$$\partial(f,t) = \left\{ v \in \boldsymbol{R}^m : v \epsilon t \partial p(0) + \frac{1}{t} \partial q(0) \right\}.$$

Thus from Theorem 3.2, $a \in C$ solves (P_4'') if and only if for every $t > 0$,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} t\partial p(0) + \frac{1}{t}\partial q(0) \\ 2\sqrt{p(a)q(a)} \end{pmatrix} + \text{cone co} \bigcup_{\substack{i \in I \\ r \geq 0}} \begin{pmatrix} a_i \\ b_i + r \end{pmatrix},$$

where $a_i, i = 1, \ldots, m$ are the row vectors of A and $I = \{1, 2, \ldots, m\}$. Thus the conclusion follows. □

Using the same approach as that above, one can obtain complete characterizations of global optimality for several nonconvex problems. We provide some examples below.

$$(P_5) \qquad \begin{array}{ll} \text{global minimize} & p(x)q(x) \\ \text{subject to} & g_i(x) \leq b_i, \ i \in I, \end{array}$$

where p and q are as above and g_i are MSL functions.

$$(P_6) \qquad \begin{array}{ll} \text{global maximize} & f(x) \\ \text{subject to} & p_i(x)q_i(x) \leq b_i, \ i \in I, \end{array}$$

where $f : X \to \boldsymbol{R}$ is a continuous convex function, $p_i, q_i : X \to \boldsymbol{R}$, $i \in I$ are continuous sublinear functions satisfying that $p_i(x) \geq 0$, $q_i(x) \geq 0$ for all $x \in X$ and $p_i(x)q_i(x) = 0$ only if $p_i(x) = q_i(x) = 0$, and $b_i \geq 0$, $i \in I$.

$$(P_7) \qquad \begin{array}{ll} \text{global minimize} & \sqrt{p_0(x)q_0(x)} + r_0(x) \\ \text{subject to} & \sqrt{p_i(x)q_i(x)} + r_i(x) \leq b_i, \ i \in I, \end{array}$$

where $p_i, q_i, r_i : X \to \boldsymbol{R}$, $i \in \{0\} \cup I$ are continuous sublinear functions satisfying that $p_i(x) \geq 0$, $q_i(x) \geq 0$ for all $x \in X$ and $p_i(x)q_i(x) = 0$ only if $p_i(x) = q_i(x) = 0$.

We conclude by giving a solvability theorem for an MSL inequality system.

THEOREM 4.2 (solvability for MSL inequality system). *Let I be an arbitrary index set, $g_i : X \to \boldsymbol{R}, i \in I$, be MSL functions with* (3) *being satisfied and $b_i \in \boldsymbol{R}, i \in I$. Then exactly one of the following statements holds.*

(i)

$$(\exists x \in X) \quad g_i(x) \leq b_i, \ i \in I. \tag{10}$$

(ii) *For every selection* (τ_i) *with* $\tau_i \in T_i$ *for* $i \in I$,

$$\begin{pmatrix} 0 \\ -1 \end{pmatrix} \in \text{cl cone co} \bigcup_{i \in I} \begin{pmatrix} \partial(g_i, t_i) \\ b_i \end{pmatrix}.$$

Proof. Define $F, G_i : X \times \boldsymbol{R} \to \boldsymbol{R}$ by

$$F(x,t) = -t,$$
$$G_i(x,t) = g_i(x) - tb_i, \ i \in I.$$

Then, recalling that g_i are positively homogeneous, we note that the negation of (10) is equivalent to

$$(\forall i \in I) \quad G_i(x,t) \leq 0 \ \Rightarrow \ F(x,t) \geq 0.$$

Clearly F and G_i are MSL functions with representations:

$$F(x,t) = \min_{i \in \{0\}} \sup \{v(x) + \gamma t : (v,\gamma) \in \{0\} \times \{-1\}\},$$
$$G_i(x,t) = \min_{t_i \in T_i} \sup \{v_i(x) + \gamma_i t : (v_i, \gamma_i) \in \partial(g_i, t_i) \times \{-b_i\}\},$$

and for every $(\tau_i) \in \mathcal{T}$, $(x^o, t^o) = (0,0)$ one has

$$\sup \{v_i(x^o) + \gamma_i t^o : (v_i, \gamma_i) \in \partial(g_i, t_i) \times \{-b_i\}\} \leq 0.$$

Thus the conclusion follows from Theorem 2.2. □

5. Discussion. To solve constrained optimization problems using mathematical programming techniques it is important to develop necessary optimality conditions in dual (computationally verifiable) forms, that is, in terms of gradients or subdifferentials. In global nonconvex optimization, necessary optimality conditions are usually of limited value because of the occurrence of multiple local optima. In this paper, we presented dual characterizations of global optimality for various classes of nonconvex optimization problems where the constraints are pointwise minimum of sublinear functions. These dual conditions were given in terms of ϵ-subdifferentials. The computer calculation of ϵ-subdifferentials is critical for developing a global optimization method based on the necessary and sufficient optimality conditions. Recently, Hiriart-Urruty and Lemarechal [7] have provided a starting point for a detailed investigation of dual necessary and sufficient global optimality conditions for the maximization of convex functions utilizing the ϵ-subdifferential. More recently, Bomze and Danninger [1] have given a global optimization algorithm for concave quadratic programming problems based on dual global optimality conditions. The results of this paper show how the study of necessary and sufficient global optimality conditions can be extended for large classes of nonconvex optimization problems. Moreover, the results demonstrate the use of the ϵ-subdifferential set and the tools of nonsmooth analysis for the study of global optimality conditions of nonconvex optimization problems. However, significant computational problems arise in the calculation of the ϵ-subdifferential of the functions involved in the problem. The difficulties from a theoretical standpoint in even the simple case of convex quadratic functions are outlined in Hiriart-Urruty and Lemarechal [6].

Acknowledgments. The authors wish to thank the associate editor and anonymous referees for their helpful comments which have improved the presentation of this paper.

REFERENCES

[1] I. M. BOMZE AND G. DANNINGER, *A global optimization algorithm for concave quadratic programming problems*, SIAM J. Optim., 3 (1993), pp. 826–842.

[2] B. M. GLOVER, *A generalized Farkas lemma with applications to quasidifferentiable programming*, Z. Oper. Res., 26 (1982), pp. 125–142.

[3] B. M. GLOVER, V. JEYAKUMAR, AND W. OETTLI, *A Farkas lemma for difference sublinear systems and quasidifferentiable programming*, Math. Programming, 63 (1994), pp. 109–125.

[4] J.-B. HIRIART-URRUTY, *From convex minimization to nonconvex minimization: Necessary and sufficient conditions for global optimality*, in Nonsmooth Optimization and Related Topics, F. H. Clarke, V. F. Demyanov, and F. Giannessi, eds., Plenum, New York, 1989, pp. 219–240.

[5] ———, *ϵ-subdifferential calculus*, in Convex Analysis and Optimization, J.-P. Aubin and R. B. Vinter, eds., Pitman Advanced Publishers, New York, 1982, pp. 43–92.

[6] J.-B. HIRIART-URRUTY AND C. LEMARECHAL, *Testing Necessary and Sufficient Conditions for Global Optimality in the Problem of Maximizing a Convex Quadratic Function Over a Convex Polyhedran*, Preliminary report, Seminar of Numerical Analysis, University Paul Sabatier, Toulouse, 1990.

[7] ———, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[8] R. HORST, *Deterministic methods in constrained global optimization: Some recent advances and new fields of application*, Naval Res. Logist., 37 (1990), pp. 433–471.

[9] R. HORST AND H. TUY, *Global Optimization*, Springer-Verlag, Berlin, 1990.

[10] V. JEYAKUMAR AND B. M. GLOVER, *A new version of Farkas lemma and global convex maximization*, Appl. Math. Lett., 6(5) (1993), pp. 39–43.

[11] H. KONNO AND T. KUNO, *Linear multiplicative programming*, Math. Programming, 56 (1992), pp. 51–64.

[12] P. M. PARDALOS AND J. B. ROSEN, *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science 268, Springer-Verlag, Berlin, New York, 1987.

[13] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[14] T. TANAKA, P. T. THACH, AND S. SUZUKI, *Two nonconvex minimization approaches for the problem of determining an economic ordering policy for jointly replenished items*, J. Oper. Res. Soc. Japan, 34 (1991), pp. 109–124.

[15] H. TUY, *D.C. optimization: Theory, methods and algorithms*, in Handbook of Global Optimization, R. Horst and P. M. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1995, pp. 149–216.

[16] C. ZALINESCU, *Solvability results for sublinear functions and operators*, Z. Oper. Res., 31 (1987), pp. 79–101.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 373–417, May 1996
© 1996 Society for Industrial and Applied Mathematics
008

PROJECTED SEQUENTIAL QUADRATIC PROGRAMMING METHODS*

MATTHIAS HEINKENSCHLOSS†

Abstract. In this paper we introduce and analyze a class of optimization methods, called projected sequential quadratic programming (SQP) methods, for the solution of optimization problems with nonlinear equality constraints and simple bound constraints on parts of the variables. Such problems frequently arise in the numerical solution of optimal control problems. Projected SQP methods combine the ideas of projected Newton methods and SQP methods. They use the simple projection onto the set defined by the bound constraints and maintain feasibility with respect to these constraints. The iterates are computed using an extension of SQP methods and require only the solution of the linearized equality constraint. Global convergence of these methods is enforced using a constrained merit function and an Armijo-like line search. We discuss global and local convergence properties of these methods, the identification of active indices, and we present numerical examples for an optimal control problem governed by a nonlinear heat equation.

Key words. sequential quadratic programming methods, projected Newton methods, merit function, bound constraints, optimal control

AMS subject classifications. 49M37, 90C30

1. Introduction.

In this paper we analyze a class of methods for solving optimization problems of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0, \\ & a \le u \le b, \end{aligned} \tag{1.1}$$

where the variable x is partitioned into two groups y and u:

$$x = (y, u) \in \mathbb{R}^{m+n}.$$

The functions

$$f : \mathbb{R}^{m+n} \to \mathbb{R}, \qquad c : \mathbb{R}^{m+n} \to \mathbb{R}^m$$

are assumed to be sufficiently smooth. Detailed assumptions are stated below. The inequalities $a \le u \le b$ are understood component wise.

Problems of the form (1.7) occur in the numerical solution of optimal control problems. In this case y and u are the state variables and control variables, respectively, and the equality constraint $c(x) = 0$ represents the state equation. The applications we have in mind are optimal control problems governed by nonlinear partial differential equations. In this case the equality constraints are obtained from a discretization of the nonlinear differential equation and form a very large system with a particular structure.

A typical example for such an optimal control problem is the control of a heating process. The following simplified model for the heating of a probe in a kiln is discussed

* Received by the editors June 6, 1994; accepted for publication (in revised form) November 1, 1994. This work was supported in part by National Science Foundation grant DMS-9403699 and by Air Force Office of Scientific Research grant F49620-93-1-0280.

† Virginia Polytechnic Institute and State University, Interdisciplinary Center for Applied Mathematics and Department of Mathematics, Blacksburg, VA 24061-0531 (na.heinken@na-net.ornl.gov).

in [6]. The temperature $y(x,t)$ inside the probe to be heated obeys the following differential equation:

$$
\begin{aligned}
\tau(y(x,t))\tfrac{\partial y}{\partial t}(x,t) - \partial_x(\kappa(y(x,t))\partial_x y(x,t)) &= q(x,t), && (x,t) \in Q,\\
\kappa(y(0,t))\partial_x y(0,t) &= g[y(0,t) - u(t)], && t \in (0,T),\\
\kappa(y(1,t))\partial_x y(1,t) &= 0, && t \in (0,T),\\
y(x,0) &= y_0(x), && x \in \Omega\,.
\end{aligned} \tag{1.2}
$$

The spatial domain is given by $\Omega = (0,1)$. We set $Q = \Omega \times (0,T)$. The boundary $x = 1$ is the inside of the probe and $x = 0$ is the boundary of the probe. The functions $\tau : I\!R \to I\!R$ and $\kappa : I\!R \to I\!R$ denote the specific heat capacity and the heat conduction, respectively, y_0 is the initial temperature distribution, q is the source term, and g is a given scalar.

If the goal is to control the heating process in such a way that the temperature inside the probe follows a certain desired temperature profile $y_d(t)$, then we obtain the following control problem:

$$
\text{Minimize } \frac{1}{2}\int_0^T [(y(1,t) - y_d(t))^2 + \gamma u^2(t)]dt \tag{1.3}
$$

over all $x = (y,u)$ satisfying the state equation (1.2) and the control constraints

$$
u_{\text{low}} \le u \le u_{\text{upp}}. \tag{1.4}
$$

Here $u_{\text{low}}, u_{\text{upp}} \in L^\infty(0,T)$ are given functions. If the differential equation and the integral are discretized we obtain an optimization problem of the form (1.1). This example is investigated further in §6.

As mentioned before, the applications we have in mind are optimal control problems governed by nonlinear partial differential equations. In this case the equality constraint is a very large nonlinear system with a particular structure obtained from a discretization of the nonlinear differential equation. As indicated by the formulation (1.1), we treat state and control as independent variables. One of the advantages of this approach is that we do not need to solve $c(y,u) = 0$ in y for given u, i.e., we do not need to solve the nonlinear state equation for given control u. For the problems we have in mind this would be expensive. The solution of the linearized state equation, however, is usually less time consuming. In fact, if the nonlinear state equation is solved by Newton's method, then the computation of y for given u requires the formation and solution of a sequence of linearized state equations. Moreover, because of the decoupling of state and control, the problem usually becomes less nonlinear and the optimization algorithm needs fewer iterations. For a more detailed exposition of this all-at-once approach and its comparison with other formulations of optimization problems we refer to [11].

The structure of optimal control problems will be used in the design of projected sequential quadratic programming (SQP) methods. In particular, we will show that for the application of these methods to optimal control problems of the form (1.2), (1.3), (1.4) the user must provide a subroutine for the solution of the linearized state equation, a subroutine for the solution of the adjoint equation, and some first order derivative information. The algorithms that are used to solve these two linear problems do not enter the formulation and analysis of projected SQP methods. Therefore, existing efficient algorithms can be used and new algorithms for the solution of these linear problems can be incorporated rather easily.

The algorithms for (1.1) which are introduced and analyzed in this paper are obtained as combinations and extensions of projected Newton methods and SQP methods. They make use of the simple projection onto the set $\mathcal{U} = \{a \leq u \leq b\}$ and generate iterates which are feasible with respect to the inequality constraints. As in SQP methods, the steps are computed using a quadratic model of the Lagrangian and the linearization of the equality constraints. Because of the bound constraints this is coupled with the projection onto the set $\mathcal{U}$. For the modeling of the quadratic subproblem and for the computation of the steps we use the so-called separability approach recently introduced for SQP methods. This formulation allows us to use the structure of optimization problems arising in optimal control. We formulate our algorithms using the reduced SQP approach, which is suitable if the dimension of the control space n is much smaller than the dimension of the state space m. This is the case for boundary control problems where u acts only on a portion of the boundary, such as (1.2), (1.3), (1.4).

To ensure global convergence we introduce a merit function and an Armijo-like line search procedure. The merit function is similar to Fletcher's exact merit function, but we use the particular structure of the problem and the algorithm. Since the algorithm generates iterates which are feasible with respect to the box constraints, our merit function is constrained.

For the computation of the steps and the Lagrange multiplier estimates we only need to solve linear problems. As mentioned before, if the optimization problem arises from a discretization of an optimal control problem, we only need to solve the linearized state equation and the adjoint equation. Since we use the separability approach, the methods used for the solution of the linearized state equation and the adjoint equation are not part of our formulation of the projected SQP methods. This modular formulation allows a rather easy and flexible combination of the optimization algorithms with efficient, often application-dependent solvers for the linearized state and the adjoint equations.

If the equality constraints are linear, i.e., if $c(x) = C_y y + C_u u + c$, where C_y, C_u are constant matrices and c is a constant vector, then our methods are equivalent to projected Newton methods and if $a = -\infty, b = \infty$, then our methods are equivalent to SQP methods. We will show that, under suitable assumptions, the active set is identified after finitely many iterations. If the active set is identified, then the projected SQP method is equivalent to an SQP method on the inactive indices. Therefore, the local convergence behavior of our method is determined by the underlying SQP method, i.e., by the way in which second order information is approximated. If exact second order information is used, then the local convergence rate is q-quadratic. If quasi-Newton methods are used to approximate second order terms, the algorithm shows some kind of q-superlinear convergence depending on the update used.

Since we treat the inequality constraints using projections onto the set $\mathcal{U}$, the reduced problems are of constant size n. Therefore, quasi-Newton updates for the reduced Hessian can be used and analyzed in a rather straightforward way.

The problem (1.1) is a special case of the general equality, inequality constrained optimization problem

$$\begin{aligned} &\min && f(x) \\ &\text{s.t.} && c(x) = 0\,, \\ &&& a(x) \geq 0\,. \end{aligned} \tag{1.5}$$

A common approach for the solution of the large scale problem (1.5) is to linearize

the problem at a current iterate x_c and determine a step s as the solution of

$$\begin{aligned} \min \quad & \nabla_x f(x)^T s + \tfrac{1}{2} s^T H_c s \\ \text{s.t.} \quad & C(x_c)^T s + c(x_c) = 0\,, \\ & A(x_c)^T s + a(x_c) \geq 0\,, \end{aligned} \tag{1.6}$$

where $C(x_c)^T$, $A(x_c)^T$ denote the Jacobians of c and a, respectively, and where H_c is the (full) Hessian of the Lagrange function or an approximation thereof. See, e.g., [1], [3], [4], [12], [22], [29], [33]. Although (1.6) is a quadratic problem, its solution is not trivial and may be very time consuming, even if the constraints are given by simple bound constraints; cf. [9], [19], [21], [37]. Our approach is different in that we compute the steps using an SQP-like approach which is combined with the projection onto the set described by the bound constraints. The presence of inequality constraints in (1.6) usually requires an iterative procedure for the solution of (1.6). This is not the case in our approach. Steps are computed directly and not by iterative schemes for the solution of subproblems. In all of the previously mentioned papers Newton-SQP or full SQP methods are used. We are able to use reduced SQP methods, which are favorable for optimal control problems with boundary controls as described above.

Another approach for the solution of (1.1) with bounds on all variables, i.e., with $a \leq u \leq b$ replaced by $a \leq x \leq b$, is presented in [10]. In this approach the equality constraint is included into the objective function using an augmented Lagrangian function. This approach requires the solution of a minimization problem with nonlinear objective function and simple bound constraints in every iteration. This algorithm can be shown to converge globally and with a local r-linear convergence rate. As outlined previously, the algorithms presented here require only the solution of linear problems in every iteration, they converge globally, and locally they have at least some kind of q-superlinear convergence rate.

Interior point methods for optimization problems with nonlinear objective function, linear constraints, and bound constraints are studied, e.g., in [5], [28], [35], [36]. Recently, interior point methods have also been studied for general nonlinear programming problems. See [15], [30], [34].

In [23] a pointwise projected Newton method was introduced and analyzed for the solution of optimal control problems with bound constraints on the control. The starting point for the derivation of this method is the optimality system for the optimal control problem, which is viewed as a nonlinear system of equations. The presence of a projection, however, makes this nonlinear system nondifferentiable. This difficulty is overcome in [23] by using ideas from projected Newton methods; cf. §3. This approach is similar to ours in the sense that both approaches are motivated by optimal control problems, both methods require only the solution of the linearized state equation, and both methods make use of the simple projection. However, both methods are different. The pointwise projected Newton method in [23] uses exact second order information and the convergence theory is local. The convergence analysis in [23] is done in an infinite dimensional framework for optimal control problems governed by ordinary differential equations. Some of the features of this method are related to this function space framework. The algorithms in this paper are derived in a different way by combining SQP and projected Newton methods. As mentioned above, our algorithms use reduced SQP formulation, they allow the use of quasi-Newton updates, and a globalization of the convergence is achieved by introducing a merit function in combination with a step size rule. Our convergence analysis is done for the finite dimensional problem.

To simplify the presentation, we consider the following problem:

$$\begin{aligned} &\min && f(x) \\ &\text{s.t.} && c(x) = 0\,, \\ &&& u \geq 0\,. \end{aligned} \tag{1.7}$$

The extension necessary to solve problems of the form (1.1) is straightforward.

Notation. We write gradients as column vectors and by $C(x) = (\nabla c_1(x), \ldots, \nabla c_m(x)) \in \mathbb{R}^{(m+n)\times m}$ we denote the transpose of the Jacobian of c.

Because of the partitioning of $x = (y, u) \in \mathbb{R}^{m+n}$ we use similar partitions for steps and derivatives. For example, the matrix $C(x)$ is partitioned into

$$C(x) = \begin{pmatrix} C_y(x) \\ C_u(x) \end{pmatrix}, \quad C_y(x) \in \mathbb{R}^{m\times m}, C_u(x) \in \mathbb{R}^{n\times m}$$

and the step s_x and the gradient of f are written in the form $s_x = (s_y, s_u)$ and $\nabla f(x)^T = (\nabla_y f(x)^T, \nabla_u f(x)^T)$, respectively.

If we want to emphasize the partitioning of x we often replace the argument x by (y, u). If functions are evaluated at $x_k = (y_k, u_k)$, we sometimes omit the argument x_k and use a subscript k instead. For example, we may write f_k, c_k, L_k, λ_k instead of $f(x_k), c(x_k), L(x_k, \lambda_k), \lambda(x_k)$.

If not stated otherwise, we use the ℓ_2-norm: $\|\cdot\| = \|\cdot\|_2$.

Throughout this paper we make the following assumptions.

(A1) The functions f and c are twice differentiable with Lipschitz continuous second derivatives.

(A2) The matrix $C_y(x)$ is invertible for all $x = (y, u)$ with $u \geq 0$.

If (1.7) is obtained from an optimal control problem the matrix $C_y(x)$ is obtained from a discretization of the linearized state equation. Often, the linearized state equation is uniquely solvable such that the assumption (A2) is no restriction.

We consider problems in which the number n of control variables u is small compared to the number m of state variables. This does not influence the formulation and the analysis of the projected SQP methods, but is used only to motivate the use of reduced SQP methods. In theory, reduced SQP methods can be used independent of the relation between n and m. But if $m + n$ is very large and $n \ll m$, then there are some obvious advantages if reduced methods are used.

In §§2 and 3 we briefly review reduced SQP methods and projected Newton methods, respectively. The purpose of these sections is to introduce some notation and the framework for projected SQP methods. Projected SQP methods are introduced in §4. In this section we also formulate the merit function and the line search procedures and discuss some basic properties and the well posedness of our projected SQP methods. The convergence analysis, including the proof of the identification of active indices, will be given in §5. A numerical example using the boundary control problem (1.2), (1.3), (1.4) is studied in §6.

2. Solution of the equality constrained problem. In this section we consider the numerical solution of the equality constrained problem corresponding to (1.7) given by

$$\begin{aligned} &\min && f(x) \\ &\text{s.t.} && c(x) = 0\,. \end{aligned} \tag{2.1}$$

The purpose of this section is to introduce some basic results and notation and to provide a motivation for projected SQP methods for solving (1.7).

As outlined in the Introduction, we consider the case that n is small compared to m. This is satisfied, e.g., for boundary control problems and is introduced only to motivate the use of reduced SQP methods. Theoretically this relation between m and n is not needed.

The presentation here uses the so-called separability framework introduced and used for the convergence analysis of reduced SQP methods in [25], [26], [27].

The Lagrange function for (2.1) is given by

$$L(x,\lambda) = f(x) + \lambda^T c(x). \tag{2.2}$$

The first order necessary optimality conditions for (2.1) are given by $\nabla_x L(x,\lambda) = 0$, $\nabla_\lambda L(x,\lambda) = 0$. Because of the partitioning of x this is equivalent to the following system:

$$\begin{aligned} \nabla_y f(x) + C_y(x)\lambda &= 0, \\ \nabla_u f(x) + C_u(x)\lambda &= 0, \\ c(x) &= 0. \end{aligned} \tag{2.3}$$

For optimal control problems the third equation in (2.3) represents the state equation, the first equation corresponds to the adjoint equation which determines the co-state λ, and the left-hand side of the second equation corresponds to the gradient.

If Newton's method is applied for the solution of the optimality system (2.3), then in each iteration the following system must be solved:

$$\begin{pmatrix} \nabla_{xx} L(x,\lambda) & C(x) \\ C(x)^T & 0 \end{pmatrix} \begin{pmatrix} s_x \\ s_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x) + C(x)\lambda \\ c(x) \end{pmatrix}. \tag{2.4}$$

Using $s_x = (s_y, s_u)$, the equation $C(x)^T s_x = -c(x)$ can be written as $C_y(x)^T s_y + C_u(x)^T s_u = -c(x)$ or, equivalently, as

$$s_y = -C_y(x)^{-T} C_u(x)^T s_u - C_y(x)^{-T} c(x). \tag{2.5}$$

Thus, the set of solutions for the linearized constraint is given by

$$s_x = r(x) + T(x) s_u, \qquad s_u \in \mathbb{R}^n, \tag{2.6}$$

where

$$r(x) = \begin{pmatrix} -C_y(x)^{-T} c(x) \\ 0 \end{pmatrix}, \quad T(x) = \begin{pmatrix} -C_y(x)^{-T} C_u(x)^T \\ I \end{pmatrix}. \tag{2.7}$$

The matrix $T(x)$ characterizes the null space of $C(x)^T$. It holds that

$$\mathcal{N}(C(x)^T) = \{ s \mid s = T(x) s_u,\ s_u \in \mathbb{R}^n \}.$$

Using (2.6), (2.7), and the equalities $C(x)^T T(x) = 0$, $C(x)^T r(x) = -c(x)$ we can rewrite (2.4) in the reduced form

$$\big(\nabla_{xx} L(x,\lambda) T(x),\ C(x) \big) \begin{pmatrix} s_u \\ s_\lambda \end{pmatrix} = -\big(\nabla_x f(x) + C(x)\lambda + \nabla_{xx} L(x,\lambda) r(x) \big). \tag{2.8}$$

The systems (2.4), (2.8) are solvable if the reduced Hessian

$$\begin{aligned} H(x,\lambda) &= T(x)^T \nabla_{xx} L(x,\lambda)\, T(x) \\ &= C_u(x) C_y(x)^{-1} \nabla_{yy} L(x,\lambda)\, C_y(x)^{-T} C_u(x)^T - C_u(x) C_y(x)^{-1} \nabla_{yu} L(x,\lambda) \\ &\quad -\nabla_{uy} L(x,\lambda)\, C_y(x)^{-T} C_u(x)^T + \nabla_{uu} L(x,\lambda) \end{aligned} \tag{2.9}$$

is nonsingular. In fact, multiplying (2.8) with $T(x)^T$ from the left and using $T(x)^T C(x) = 0$ yields

$$s_u = -H(x,\lambda)^{-1} T(x)^T (\nabla_x f(x) + \nabla_{xx} L(x,\lambda) r(x)). \tag{2.10}$$

Note that the reduced Hessian is positive definite at points near local minimizers for which the second order sufficient conditions are satisfied.

If second order information is available, then the new iterate is given by $(y_+, u_+) = (y + \alpha s_y, u + \alpha s_u)$, where s_u, s_y are computed using (2.10), (2.5) and $\alpha \geq 0$ is a step size. If the new iterate $x_+ = (y_+, u_+)$ is determined, then a new estimate for the Lagrange multiplier is computed. One way to define a new Lagrange multiplier estimate is to use the least squares solution of $C(x_+)\lambda = -\nabla_x f(x_+)$, i.e.,

$$\begin{aligned} \lambda(x_+) = -\Big(C_y(x_+)^T C_y(x_+) + C_u(x_+)^T C_u(x_+) \Big)^{-1} \\ \cdot \Big(C_y(x_+)^T \nabla_y f(x_+) + C_u(x_+)^T \nabla_u f(x_+) \Big). \end{aligned}$$

However, other Lagrange multiplier estimates are possible. In fact, for optimal control problems it is more efficient to use the co-state as the Lagrange multiplier estimate:

$$\lambda(x_+) = -C_y(x_+)^{-1} \nabla_y f(x_+).$$

Since the reduced gradient $T(x)^T \nabla_x f(x)$ can be written as

$$T(x)^T \nabla_x f(x) = -C_u(x) C_y(x)^{-1} \nabla_y f(x) + \nabla_u f(x) = C_u(x)\lambda(x) + \nabla_u f(x),$$

the term $-C_y(x_+)^{-1} \nabla_y f(x_+)$ must be computed anyway and its use as a Lagrange multiplier estimate requires no extra cost.

If second order information is too expensive to compute, the reduced Hessian $H(x,\lambda)$ is replaced by a matrix H which is updated using some quasi-Newton formula. The vector $T(x)^T \nabla_{xx} L(x,\lambda)\, r(x)$ in (2.10) is also replaced by some vector $d(x)$ using only lower order derivatives. Usually the choice of $d(x)$ depends on the quasi-Newton update used for $H(x,\lambda)$. It is beyond the scope of the paper to give an overview of the possible choices of H and d and their local convergence analysis. We refer to the literature, e.g., [7], [8], [20], [31], [25], [27], for a detailed discussion. The formulation and analysis of the algorithms studied in this paper do not depend on a particular choice of H and d. We assume only that H is uniformly positive definite and that the size of d is related to the size of c. In particular we require that $d(x) = 0$ is zero if $c(x) = 0$. This is a natural assumption since $d(x) \approx T(x)^T \nabla_{xx} L(x,\lambda)\, r(x)$ and $r(x) = 0$ if $c(x) = 0$.

For the purpose of presentation and a numerical comparison with existing results, we mention the following reduced SQP method which uses a BFGS update and $d(x_k) = 0$. This method has been used in [27] for the numerical solution of the control problem (1.2), (1.3). A local convergence analysis of this and other reduced SQP methods in Hilbert spaces is given in [25].

ALGORITHM 2.1 (REDUCED BFGS METHOD).

Initialization:

Given $x_0 \in \mathbb{R}^{m+n}$ and $H_0 \in \mathbb{R}^{n\times n}$, H_0 positive definite.

Set $k = 0$.

Step k:

Compute the reduced gradient:

Solve $C_y(x_k)\,\lambda = -\nabla_y f(x_k)$.

Compute $T(x_k)^T \nabla_x f(x_k) = C_u(x_k)\lambda(x_k) + \nabla_u f(x_k)$.

Solve $H_k s_u = -T(x_k)^T \nabla f(x_k)$.

Solve $C_y(x_k)^T s_y = -c(x_k) - C_u(x_k)^T s_u$.

Set $y_{k+1} = y_k + s_y$, $u_{k+1} = u_k + s_u$.

Update H_{k+1}:

Compute $v = T(x_k + T(x_k)s_u)^T \nabla f(x_k + T(x_k)s_u) - T(x_k)^T \nabla f(x_k)$.

If $v^T s_u > 0$, set

$$H_{k+1} = H_k + \frac{vv^T}{v^T s_u} - \frac{(H_k s_u)(H_k s_u)^T}{s_u^T H_k s_u}.$$

Else set $H_{k+1} = H_k$.

An important observation is that reduced SQP methods using the separability framework can be combined in a natural way with tools usually available for the solution of optimal control problems. If Algorithm 2.1 is applied to an optimal control problem, then the user mainly has to provide the solution λ of the adjoint equation $C_y(x_k)\lambda = -\nabla_y f(x_k)$ and the solution s_y of the linearized state equation $C_y(x_k)^T s_y = -c(x_k) - C_u(x_k)^T s_u$. Note that the derivative $C_u(x_k)^T$ is not needed explicitly, but only in the form $C_u(x_k)^T s_u$, a term which may be approximated by finite differences. Since n is assumed to be small compared to m, the solution of $H_k s_u = -T(x_k)^T \nabla f(x_k)$ can be done using standard techniques; cf. [13].

Algorithm 2.1 is a local version. In every iteration full steps are taken. To guarantee global convergence, one must add an appropriate line search or trust region strategy [22], [14], [29], [32]. For the determination of the step size or trust region radius one needs a so-called merit function. One possible choice is Fletcher's exact merit function [17], [18] given by

$$\Psi(x, \lambda, \rho) = f(x) + \lambda(x)^T c(x) + \rho ||c(x)||^2. \tag{2.11}$$

3. Projected Newton methods. In this section we show how projected Newton methods introduced in [2] can be applied to solve (1.7). We will also exploit some relations between these methods and the reduced SQP method investigated in §2. In §4 these relations will be used to derive projected SQP methods.

Projected Newton methods solve nonlinear optimization problems with simple constraints. To apply these methods to (1.7) we first must eliminate the equality constraint $c(x) = 0$. Suppose that instead of (A2) the following assumption is satisfied.

(A2′) For all $u \geq 0$ there exists a solution y of $c(y, u) = 0$. The matrix $C_y(x)$ is invertible for all $x = (y, u)$ with $u \geq 0$ and $c(y, u) = 0$.

Then the implicit function theorem guarantees the existence of a differentiable function

$$y : \mathbb{R}^n_+ \to \mathbb{R}^m$$

defined by

$$c(y(u), u) = 0.$$

The derivative of y is given by

$$y'(u) = -C_y(y(u), u)^{-T} C_u(y(u), u)^T. \tag{3.1}$$

Using the function $y(\cdot)$, the problem (1.7) can be written as

$$\begin{aligned} &\min \quad \phi(u) = f(y(u), u) \\ &\text{s.t.} \quad u \geq 0\,. \end{aligned} \tag{3.2}$$

Since $y(\cdot)$ is differentiable the function ϕ is differentiable and its gradient is given by

$$\begin{aligned} \nabla\phi(u) &= y'(u)^T \nabla_y f(y(u), u) + \nabla_u f(y(u), u) \\ &= -C_u(y(u), u) C_y(y(u), u)^{-1} \nabla_y f(y(u), u) + \nabla_u f(y(u), u). \end{aligned}$$

If $\lambda(u) \in \mathbb{R}^m$ is the solution of

$$C_y(y(u), u)\, \lambda(u) = -\nabla_y f(y(u), u), \tag{3.3}$$

then the gradient can be written as

$$\nabla\phi(u) = \nabla_u f(y(u), u) + C_u(y(u), u)\, \lambda(u). \tag{3.4}$$

The system (3.3) corresponds to the adjoint equation in optimal control problems. From (2.3) one can see that the adjoint equation (3.3) is equivalent to

$$\nabla_y L(y(u), u, \lambda(u)) = 0. \tag{3.5}$$

Comparing (3.4) with the gradient (2.3) of the Lagrange function (2.2) we see that

$$\nabla\phi(u) = \nabla_u L(y(u), u, \lambda(u)). \tag{3.6}$$

Moreover, using the definition (2.7) of T and equation (3.3) the gradient can also be written as

$$\nabla\phi(u) = T(y(u), u)^T \nabla_x f(y(u), u). \tag{3.7}$$

Equations (3.7) and (3.5) show that $(y(u_*), u_*, \lambda(u_*))$ solve the necessary optimality conditions (2.3) if and only if $\nabla\phi(u_*) = 0$.

For the discussion of the projected Newton method we also need the second derivative of ϕ. Under the assumption (A2′) the Hessian of ϕ exists and is given by

$$\begin{aligned} \nabla^2\phi(u) = {} & \nabla_{uy} L(y(u), u, \lambda(u))\, y'(u) + \nabla_{uu} L(y(u), u, \lambda(u)) \\ & + \nabla_{u\lambda} L(y(u), u, \lambda(u))\, \lambda'(u), \end{aligned} \tag{3.8}$$

where $y'(u)$ is defined by (3.1). The derivative $\lambda'(u)$ is obtained by differentiating (3.5):

$$\nabla_{yy} L(y(u), u, \lambda(u))\, y'(u) + \nabla_{yu} L(y(u), u, \lambda(u)) + \nabla_{y\lambda} L(y(u), u, \lambda(u))\, \lambda'(u) = 0.$$

Using $\nabla_{y\lambda} L(y(u), u, \lambda(u)) = C_y(y(u), u)$ and (3.1) we find that

$$\lambda'(u) = C_y(y(u), u)^{-1} \big[\nabla_{yy} L(y(u), u, \lambda(u))\, C_y(y(u), u)^{-T} C_u(y(u), u)^T - \nabla_{yu} L(y(u), u, \lambda(u)) \big]. \tag{3.9}$$

Inserting (3.9) and (3.1) into (3.8) and observing that $\nabla_{u\lambda} L(y(u), u, \lambda(u)) = C_u(y(u), u)$ the Hessian of ϕ can be written as

$$\nabla^2 \phi(u) = H(y(u), u, \lambda(u)), \tag{3.10}$$

where $H(y(u), u, \lambda(u))$ is given as in (2.9) with $x = (y(u), u)$ and $\lambda = \lambda(u)$.

With (3.7) and (3.10) the Newton system for the unconstrained problem $\min \phi(u)$ is given by

$$H(y(u), u, \lambda(u))\, s_u = -T(y(u), u)^T \nabla_x f(y(u), u). \tag{3.11}$$

This equation corresponds to equation (2.10) since $x = (y(u), u)$ satisfies $c(x) = 0$ and, therefore, $r(x) = 0$.

If $u_* \geq 0$ solves (3.2), then

$$P(u_* - \alpha \nabla \phi(u_*)) = u_* \qquad \forall \alpha \geq 0,$$

where P is the projection onto the feasible set; see Theorem 4.3. For the inequality constraints in (3.2) the projection P is given by

$$(Pu)_i = \begin{cases} 0 & \text{if } u_i < 0, \\ u_i & \text{else.} \end{cases}$$

A classical method for the solution of the constrained problem (3.2) is the gradient projection method, which uses an iteration of the form

$$u_{k+1} = P(u_k - \alpha_k \nabla \phi(u_k)),$$

where $\alpha \geq 0$ is determined by some step size rule. Since the gradient projection method uses only first order derivative information, its local convergence is rather slow. It is desirable to use second order information to obtain better performance. However, naive use of the second order information in the form

$$u_{k+1} = P(u_k - \alpha_k (\nabla^2 \phi(u_k))^{-1} \nabla \phi(u_k))$$

does not work since it may be impossible to find step sizes α_k such that $\phi(u_{k+1}) < \phi(u_k)$. For an illustration see [2]. In [2], Bertsekas introduced projected Newton methods which use Hessian information and the simple projection. The idea of the projected Newton method is to omit parts of the Hessian corresponding to indices i in the active set for which the negative gradient points out of the feasible set, i.e., for which $u_i = 0$ and $\partial \phi(u)/\partial u_i > 0$. To give a precise description of this method we need the following definition.

DEFINITION 3.1. *Let $I \subset \{1, \ldots, n\}$ be an index set. A matrix $D \in \mathbb{R}^{n \times n}$ is called diagonal with respect to I if $D_{ij} = 0$ for all i, j with $i \neq j$ and $i \in I$ or $j \in I$.*

If $I = \{r+1, \ldots, n\}$, then a matrix D which is diagonal with respect to I has the form

$$D = \left(\begin{array}{c|ccc} \tilde{D} & 0 & \cdots & 0 \\ \hline 0 & d_{r+1} & & \\ \vdots & & \ddots & \\ 0 & & & d_n \end{array} \right),$$

with $\tilde{D} \in \mathbb{R}^{r \times r}$ and $d_i \in \mathbb{R}$. For an index set $A \subset \{1, \ldots, n\}$ we define its complement by

$$A^c = \{1, \ldots, n\} \setminus A$$

and the indicator function

$$I_A : \mathbb{R}^n \to \mathbb{R}^n, \quad (I_A u)_i = \begin{cases} u_i, & i \in A, \\ 0, & i \notin A. \end{cases}$$

If $H \in \mathbb{R}^{n \times n}$, then the matrix

$$D = I_A \operatorname{diag}(H) I_A + I_{A^c} H I_{A^c},$$

where $\operatorname{diag}(H)$ denotes the diagonal matrix formed with the diagonal elements of H, is diagonal with respect to A.

To define a projected Newton method we also need the following subset of the active indices:

$$A^+(u) = \left\{ i \;\middle|\; u_i = 0, \quad \frac{\partial \phi(u)}{\partial u_i} > 0 \right\}.$$

A projected Newton method takes the form

$$u_{k+1} = P(u_k - \alpha_k D_k^{-1} \nabla \phi(u_k)), \tag{3.12}$$

where D_k is a positive definite matrix which is diagonal with respect to $A^+(u_k)$ and $\alpha_k > 0$ is an appropriate step size.

It is shown in [2, Prop. 1] that u_k is a critical point for (3.2) if and only if $u_k = P(u_k - \alpha D_k^{-1} \nabla \phi(u_k))$ for all $\alpha \geq 0$. If u_k is not a critical point for (3.2), then there exists $\bar{\alpha}_k > 0$ such that $\phi(u_{k+1}) < \phi(u_k)$ for all $\alpha_k \in (0, \bar{\alpha}_k)$.

Moreover, it is proven in [2] that, under appropriate assumptions, the active indices are identified after finitely many iterations and the projected Newton method essentially becomes an unconstrained method with local convergence behavior determined by the choice of D_k. For example, if the projected Hessian

$$D_k = I_{A_k^+} \operatorname{diag}(\nabla^2 \phi(u_k))\, I_{A_k^+} + I_{A_k^{+c}} \nabla^2 \phi(u_k) I_{A_k^{+c}},$$

with $A_k^+ = A^+(u_k)$, is used in (3.12) and if the Hessian $\nabla^2 \phi(u)$ is Lipschitz continuous with eigenvalues contained in some interval $[c_1, c_2]$, $c_1 > 0$, for all $u \in \{u \mid \phi(u) \leq \phi(u_0)\}$, then the projected Newton method converges toward a local minimizer u_* of (3.2). Furthermore, if for this minimizer the active set $A(u_*) = \{i \mid u_i = 0\}$ is equal to $A^+(u_*)$, then the active indices are identified after finitely many iterations, i.e., $A(u_k) = A(u_*)$ for all $k \geq k_*$. For iterations $k \geq k_*$ the projected Newton method is equivalent to Newton's method for the unconstrained problem in the set of inactive variables and, thus, is locally q-quadratic convergent.

The previous presentation leads to the following algorithm.

ALGORITHM 3.2 (PROJECTED NEWTON METHOD).
Initialization:
Given $u_0 \in \mathbb{R}^n$, $u_{\text{low}}, u_{\text{upp}} \in \mathbb{R}^n$.
Set $k = 0$.

Step k:

Solve the nonlinear equation $c(y, u_k) = 0$ to get y_k.

Compute the gradient:

Solve $C_y(x_k)\lambda = -\nabla_y f(x_k)$.

Set $\nabla\phi(u_k) = T(x_k)^T \nabla f(x_k) = \nabla_u f(x_k) + C_u(x_k)\lambda$.

Compute the Hessian $\nabla^2\phi$ or an approximation thereof.

Determine the active set $A^+(u_k)$ and compute the reduced Hessian D_k.

Solve $D_k s_u = -T(x_k)^T \nabla f(x_k)$.

Compute the new iterate $u_{k+1} = P(u_k + \alpha s_u)$ using some step size rule.

The efficient use of the simple projection and the fast local convergence behavior makes projected Newton methods attractive. However, the obvious disadvantage is that for problems like (1.7) the nonlinear equation $c(y, u) = 0$ must be solved for y in every iteration. This is expensive for large dimensional problems with nonlinear c, such as, for example, optimal control problems with nonlinear state equation. Moreover, viewing the state y as a function of the control usually makes the problem more nonlinear and more iterations are needed.

4. Derivation of projected SQP methods. In the previous sections we have seen that SQP methods and projected Newton methods have certain advantages and disadvantages when applied to problems of the form (1.7).

SQP methods have the advantage that only linearized constraints need to be solved and that these methods have a good local convergence behavior. Moreover, the structure of optimal control problems can be used nicely. However, it is not obvious how the inequality constraints can be handled in a way which is efficient and which preserves the ability to use structure.

On the other hand, projected Newton methods use the simple form of the projection in an efficient way. Moreover, they generate iterates which are feasible, a property which is important for some applications. The disadvantage, however, is that projected Newton methods require the solution of the nonlinear equation $c(y(u), u) = 0$ in every iteration. This is necessary even if the projected gradient method is used.

In this section we show how the two methods can be combined to construct methods which have the strength of SQP and projected Newton methods, but avoid the disadvantages. We use the notation established in the previous sections.

4.1. Construction of the iterates. The Lagrange function for (1.7) is given by

$$\bar{L}(x, \lambda, \mu) = f(x) + \lambda^T c(x) - \mu^T u.$$

First order necessary optimality conditions for (1.7) are stated in the following theorem.

THEOREM 4.1. *Let the assumptions* (A1), (A2) *be valid. If* $x_* = (y_*, u_*)$ *is a solution of* (1.7), *then*

$$u_* \geq 0, \quad c(x_*) = 0, \quad P(u_* - \nabla_u L(x_*, \lambda(x_*))) = u_*, \tag{4.1}$$

where $\lambda(x_*) = -C_y(x_*)^{-1}\nabla_y f(x_*)$.

Proof. Under the assumptions (A1), (A2) the matrix

$$\begin{pmatrix} C_y(x_*)^T & C_y(x_*)^T \\ 0 & I_* \end{pmatrix} \tag{4.2}$$

has full rank. Here I_* is the matrix with columns e_i^T for i in the active set

$$A(x_*) = \{i \mid (u_*)_i = 0\}. \tag{4.3}$$

Therefore the first order constraint qualifications [16, p. 19] are satisfied. See [16, Cor. 3, p. 22].

Using [16, Cor. 1, p. 20] we can establish the existence of Lagrange multipliers $\lambda_* \in \mathbb{R}^m$, $\mu_* \in \mathbb{R}^n$ such that $(y_*, u_*, \lambda_*, \mu_*)$ satisfy the Karush–Kuhn–Tucker conditions

$$\begin{aligned} \nabla_y f(x_*) + C_y(x_*)\lambda_* &= 0, \\ \nabla_u f(x_*) + C_u(x_*)\lambda_* - \mu_* &= 0, \\ c(x_*) &= 0, \\ u_*, \mu_* &\geq 0, \\ \mu_*^T u_* &= 0. \end{aligned} \tag{4.4}$$

Using the active set we can reformulate (4.4) as

$$\nabla_y f(x_*) + C_y(x_*)\lambda_* = 0, \tag{4.5a}$$

$$\nabla_u f(x_*) + C_u(x_*)\lambda_* \geq 0, \tag{4.5b}$$

$$\left(\nabla_u f(x_*) + C_u(x_*)\lambda_*\right)_i = 0 \qquad \forall i \notin A(x_*), \tag{4.5c}$$

$$c(x_*) = 0, \tag{4.5d}$$

$$u_* \geq 0. \tag{4.5e}$$

If (4.5a), (4.5d) are satisfied, then (4.5b), (4.5c) can be written as

$$T(x_*)^T \nabla_x f(x_*) \geq 0, \tag{4.6b}$$

$$\left(T(x_*)^T \nabla_x f(x_*)\right)_i = 0 \qquad \forall i \notin A(x_*), \tag{4.6c}$$

or as $P(u_* - \alpha \nabla_u L(x_*, \lambda_*)) = u_*$ for all $\alpha \geq 0$; cf. (3.4), (3.6), (3.7). □

Our goal is to construct an algorithm for the solution of (1.7) which takes advantage of the simple projection and uses an iteration of the form (3.12) for the u-component of the iterates and requires the solution of the linearized constraints only. If we compare the presentations in §2 with those in §3, then (2.10) and (3.11) suggest replacing $\nabla\phi(u)$ by $T(x)^T(\nabla_x f(x) + \nabla_{xx}^2 L(x,\lambda)r(x))$ and $\nabla^2\phi(u)$ by the reduced Hessian $H(x,\lambda)$ to obtain an iteration in the u-component.

We will use a slightly more general formulation which is motivated by the desire to allow the approximation of second order derivatives. We introduce a function $d(x) \in \mathbb{R}^n$ which is assumed to satisfy

$$c(x) = 0 \implies d(x) = 0. \tag{4.7}$$

Obviously, (4.7) is satisfied for $d(x) = T(x)^T \nabla_{xx}^2 L(x,\lambda) r(x)$, or $d(x) = 0$. However, other choices for $d(x)$ are possible. Instead of the reduced Hessian $H(x,\lambda)$ we use a general symmetric matrix with certain positive definiteness properties which will be

specified later. For the definition of the subset $A_k^+(u)$ of active indices we replace $\nabla\phi(u)$ by $T(x)^T\nabla_x f(x) + d(x)$ and we define

$$A_k^+(x) = \Big\{ i \;\Big|\; u_i = 0, \quad \Big(T(x)^T\nabla_x f(x) + d(x)\Big)_i > 0 \Big\}. \tag{4.8}$$

For $x = x_k$ we set $A_k^+(x_k) = A_k^+$ and by $A_k^{+^c}$ we denote the complement of A_k^+. If $H_k \in \mathbb{R}^{n\times n}$ is the matrix replacing the reduced Hessian in iteration k, then we define the projected matrix by

$$D_k = I_{A_k^+}\mathrm{diag}(H_k)I_{A_k^+} + I_{A_k^{+^c}} H_k I_{A_k^{+^c}}.$$

For the definition of the step s_u in the u-direction we replace $\nabla\phi(u)$ by $T(x)^T\nabla_x f(x) + d(x)$. Since y_k is not required to satisfy the state equation, we also need a correction in y_k. In a first approach the new y-component of the iterate is computed so that the step in the u-component, $P(u_k + \alpha s_u) - u_k$, and the step in the y-component, $y_k(\alpha) - y_k$, solve the linearized constraints. Thus, if D_k is invertible, the new iterates are defined by

$$\begin{aligned} s_u &= -D_k^{-1}\big(T(x_k)^T\nabla_x f(x_k) + d(x_k)\big), \\ u_k(\alpha) &= P(u_k + \alpha s_u), \\ y_k(\alpha) &= y_k - \alpha C_y(x_k)^{-T}c(x_k) - C_y(x_k)^{-T}C_u(x_k)^T\big(P(u_k + \alpha s_u) - u_k\big), \end{aligned} \tag{4.9}$$

where $\alpha_k > 0$ is a step size. Such a step size will be determined iteratively. To evaluate $y_k(\alpha)$ for a trial step size we must solve a linear system with matrix $C_y(x_k)^T$. If $C_y(x_k)^T$ is not available in factored form this may be very expensive and (4.9) may be practically unacceptable. Therefore one must replace the definition of $y_k(\alpha)$ by a more efficient formula. One can think of the following alternatives to (4.9):

$$y_k(\alpha) = y_k - \alpha C_y(x_k)^{-T}\big(c(x_k) + C_u(x_k)^T\big(P(u_k + s_u) - u_k\big)\big) \tag{4.10}$$

or

$$y_k(\alpha) = y_k - \alpha C_y(x_k)^{-T}\big(c(x_k) + C_u(x_k)^T I_{A_k^{+^c}} s_u\big). \tag{4.11}$$

For the derivation of (4.10) we simply replaced $P(u_k+\alpha s_u)-u_k$ by $\alpha\big(P(u_k+s_u)-u_k\big)$, i.e., instead of projecting for each step size α, we project once and trace back along this direction. The second alternative (4.11) is motivated by the following consideration. If $(s_u)_i \le 0$ for all $i \in A_k^+$ and $-(s_u)_i \le (u_k)_i$ for all $i \notin A_k^+$, then $P(u_k + \alpha s_u) - u_k = \alpha I_{A_k^{+^c}} s_u$. We expect this to be the case if u_k is close to the solution u_* and the estimate of the active set A_k^+ is equal to the active set $A(u_*)$. The bold lines in Figure 4.1 illustrate the vectors $P(u_k + \alpha s_u) - u_k$, $\alpha\big(P(u_k + s_u) - u_k\big)$, and $\alpha I_{A_k^{+^c}} s_u$ for $\alpha \in [0, 1]$.

The steps $y_k(\alpha) - y_k, u_k(\alpha) - u_k$, where $y_k(\alpha), u_k(\alpha)$ are defined by (4.9), solve the linearized constraints for all α. This is generally not the case for $y_k(\alpha)$ defined by (4.10) or (4.11). The fact that the steps corresponding to (4.10) and (4.11) do not satisfy the linearized constraints causes some theoretical and practical difficulties.

If we consider the definition (4.11), then, away from the solution, it may happen that $i \notin A_k^+(x)$, but $u_i \approx 0$, or even $u_i = 0$, and $(s_u)_i < 0$. In this case the ith component of $P(u + \alpha s_u)$ and $\alpha I_{A_k^{+^c}} s_u$ coincide only for very small α or do not coincide at all. To avoid this effect we allow more flexibility in the choice of the

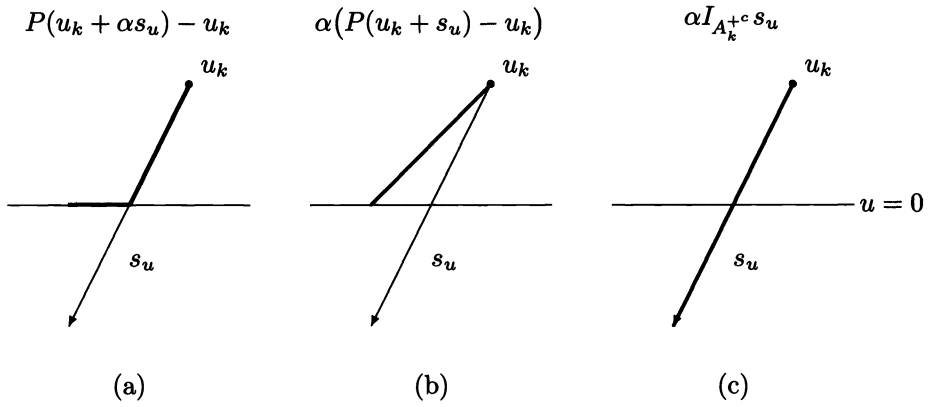


FIG. 4.1.

active set used for the computation of $y_k(\alpha)$ and introduce a subset $B(x)$ of $A^{+^c}(x)$. Motivated by the previous discussion, one may think of

$$B(x) = \{i \notin A^+(x) \mid u_i = 0 \text{ and } (s_u)_i \geq 0, \quad \text{or} \quad u_i > 0\}.$$

However, we do not specify $B(x)$ at this point. We use the notation $B_k = B(x_k)$. With this definition we replace (4.11) by

$$y_k(\alpha) = y_k - \alpha C_y(x_k)^{-T}\big(c(x_k) + C_u(x_k)^T I_{B_k} s_u\big). \tag{4.12}$$

Figure 4.1 suggests a third alternative to (4.9) which combines (4.10) and (4.12). As long as $u_k + \alpha I_{B_k} s_u$ is feasible, i.e., for $\alpha \in (0, \tilde{\alpha}_k]$ with

$$\tilde{\alpha}_k = \max\{\alpha \in [0,1] \mid u_k + \alpha I_{B_k} s_u \geq 0\}, \tag{4.13}$$

we use the direction $I_{B_k} s_u$. For step sizes $\alpha \in (\tilde{\alpha}_k, 1]$ we use a convex combination of $\tilde{\alpha}_k I_{B_k} s_u$ and $P(u_k + s_u) - u_k$. More precisely, we define the iterate in y direction as follows:

(4.14)

$$y_k(\alpha) = \begin{cases} y_k - \alpha C_y(x_k)^{-T}\big(c(x_k) + C_u(x_k)^T I_{B_k} s_u\big) & \text{if } \alpha \in (0, \tilde{\alpha}_k], \\ y_k - \alpha C_y(x_k)^{-T} c(x_k) - \frac{\alpha - \tilde{\alpha}_k}{1 - \tilde{\alpha}_k} C_y(x_k)^{-T} C_u(x_k)^T & \big(P(u_k + s_u) - u_k\big) \\ \quad -\big(1 - \frac{\alpha - \tilde{\alpha}_k}{1 - \tilde{\alpha}_k}\big) C_y(x_k)^{-T} C_u(x_k)^T \tilde{\alpha}_k I_{B_k} s_u & \text{if } \alpha \in (\tilde{\alpha}_k, 1]. \end{cases}$$

In the situation shown in Figure 4.1 the illustration for this choice is identical to Figure 4.1(a). For $\alpha \in (\tilde{\alpha}_k, 1]$ the iterate $y(\alpha)$ defined in (4.14) can be written as

$$\begin{aligned} y_k(\alpha) = y_k &- \tfrac{\alpha - \tilde{\alpha}_k}{1 - \tilde{\alpha}_k} C_y(x_k)^{-T}\Big(c(x_k) + C_u(x_k)^T\big(P(u_k + s_u) - u_k\big)\Big) \\ &-\big(1 - \tfrac{\alpha - \tilde{\alpha}_k}{1 - \tilde{\alpha}_k}\big)\tilde{\alpha}_k C_y(x_k)^{-T}\Big(c(x_k) + C_u(x_k)^T I_{B_k} s_u\Big). \end{aligned} \tag{4.15}$$

This shows that we can use the information needed for the computation of $y(\alpha)$ in (4.10) and (4.12).

We also need an update formula for the Lagrange multiplier estimate. Since our approach is related to (4.5a)–(4.5d) we need only an estimate for the Lagrange

multiplier λ. We use (4.5a) for the definition of our Lagrange multiplier estimate and set

$$\lambda(x) = -C_y(x)^{-1}\nabla_y f(x). \tag{4.16}$$

As noted already in §2 the reduced gradient can be written as

$$T(x)^T\nabla_x f(x) = \nabla_u f(x) + C_u(x)\lambda(x) = \nabla_u L(x,\lambda(x)) \tag{4.17}$$

and therefore the Lagrange multiplier estimate (4.16) is available at no extra cost. Since the Lagrange multiplier estimate λ is determined by (y,u) we say that (y,u) is a *critical point* for (1.7) if $(y,u,\lambda(y,u))$ satisfies (4.1).

As mentioned before, the algorithm for the solution of (1.7) generates iterates

$$(y_{k+1},u_{k+1}) = (y_k(\alpha_k),u_k(\alpha_k)), \tag{4.18}$$

where $(y_k(\alpha),u_k(\alpha))$ is given by (4.9), (4.10), (4.12), or (4.14), and where $\alpha_k > 0$ is some step size, and a new Lagrange multiplier estimate $\lambda_{k+1} = \lambda(y_{k+1},u_{k+1})$.

Before we discuss the choice of the step size we show that if (y_k,u_k) is a critical point, the algorithm will not move away from this point.

THEOREM 4.2. *Let* (A1) *and* (A2) *be valid. Moreover, let* $B_k \subset A_k^{+^c}$ *be arbitrary and suppose that* $u_k \geq 0$, *that* D_k *is a positive definite matrix which is diagonal with respect to* A_k^+, *and that* $d(x_k)$ *satisfies* (4.7). *Then* (y_k,u_k) *is a critical point if and only if*

$$u_k(\alpha) = u_k, \quad y_k(\alpha) = y_k$$

for all $\alpha \geq 0$, *where* $(y_k(\alpha),u_k(\alpha))$ *is given by* (4.9), (4.10), (4.12), *or* (4.14).

Proof. (i) Suppose that (y_k,u_k) is a critical point. Then

$$c(x_k) = 0, \quad d(x_k) = 0, \quad r(x_k) = 0$$

holds true; cf. (2.7), (4.7). Moreover, (4.6b), (4.6c), and the definition of A_k^+ imply that

$$\begin{aligned}
\Big(T(x_k)^T\nabla_x f(x_k) + d(x_k)\Big)_i &= \Big(T(x_k)^T\nabla_x f(x_k)\Big)_i \geq 0 \qquad \forall\, i \in A_k^+,\\
\Big(T(x_k)^T\nabla_x f(x_k) + d(x_k)\Big)_i &= \Big(T(x_k)^T\nabla_x f(x_k)\Big)_i = 0 \qquad \forall\, i \in A_k^{+^c}.
\end{aligned}$$

Since D_k is positive definite and diagonal with respect to A_k^+ this yields

$$(s_u)_i \leq 0 \quad \forall\, i \in A_k^+ \qquad \text{and} \qquad (s_u)_i = 0 \quad \forall\, i \in A_k^{+^c}.$$

Hence, $u_k(\alpha) = P(u_k + \alpha s_u) = u_k$ and $y_k(\alpha) = y_k$ for all $\alpha \geq 0$.

(ii) Suppose that $u_k(\alpha) = u_k, y_k(\alpha) = y_k$ for all $\alpha \geq 0$.

(iia) First we consider the case that $(y_k(\alpha),u_k(\alpha))$ is given by (4.9). Since $C_y(x_k)$ is nonsingular, we can deduce from the definition of $y_k(\alpha)$ that $c(x_k) = 0$. Hence, $d(x_k) = 0$. The definition of $u_k(\alpha)$ yields that

$$\begin{aligned}
(s_u)_i &= 0 \qquad \text{for all } i \text{ with } (u_k)_i > 0,\\
(s_u)_i &\leq 0 \qquad \text{for all } i \text{ with } (u_k)_i = 0.
\end{aligned} \tag{4.19}$$

From $d(x_k) = 0$ and the definition of A_k^+ we find that $(u_k)_i = 0$ and $i \notin A_k^+$ imply $\big(T(x_k)^T \nabla_x f(x_k)\big)_i \leq 0$. This relation, (4.19), and the fact that D_k is positive and diagonal with respect to A_k^+ yield that

$$\begin{aligned} 0 \leq & \sum_{i \in A_k^{+c} \cap A(x_k)} (s_u)_i \big(T(x_k)^T \nabla_x f(x_k)\big)_i = \sum_{i \in A_k^{+c}} (s_u)_i \big(T(x_k)^T \nabla_x f(x_k)\big)_i \\ & = -\big(I_{A_k^{+c}} T(x_k)^T \nabla_x f(x_k)\big)^T \big(I_{A_k^{+c}} H_k I_{A_k^{+c}}\big)^{-1} \big(I_{A_k^{+c}} T(x_k)^T \nabla_x f(x_k)\big) \leq 0. \end{aligned}$$

Hence,

$$(s_u)_i \leq 0 \quad \forall i \in A_k^+ \qquad \text{and} \qquad (s_u)_i = 0 \quad \forall i \in A_k^{+c}.$$

Using the fact that D_k is positive definite and diagonal with respect to A_k^+ and that $d(x_k) = 0$ shows that (4.6b), (4.6c) are valid. Since the corresponding Lagrange multiplier $\lambda(u_k, y_k)$ satisfies (4.5a), the pair (y_k, u_k) is a critical point.

(iib) Now we consider the case that $(y_k(\alpha), u_k(\alpha))$ is given by (4.12). We define the sets $A_1 = \{i \notin A_k^+ \mid (u_k)_i = 0, (s_u)_i < 0\}$ and $A_2 = \{i \notin A_k^+ \mid u_i = 0 \text{ and } (s_u)_i \geq 0, \text{ or } u_i > 0\}$.

For $i \notin A_2$ it holds that $(u_k)_i = 0$ and $(s_u)_i \leq 0$. Hence,

$$P(u_k + \alpha s_u) - u_k = \alpha I_{A_2} s_u$$

for all $\alpha \in (0, \tilde{\alpha})$, where

$$\tilde{\alpha} = \sup\{\alpha \mid (u_k + \alpha s_u)_i \geq 0 \quad \forall i \in A_2\} \geq \frac{1}{||s_u||} \min\{(u_k)_i \mid (u_k)_i > 0\} > 0.$$

The assumption $u_k(\alpha) = u_k$ implies that

$$0 = P(u_k + \alpha s_u) - u_k = \alpha I_{A_2} s_u.$$

Next we show that $I_{A_k^{+c}} s_u = 0$. From the definition of A_k^+ and A_1 we find that

$$s_u^T I_{A_1} \Big(T(x_k)^T \nabla f(x_k) + d(x_k)\Big) = \sum_{i \in A_1} (s_u)_i \Big(T(x_k)^T \nabla f(x_k) + d(x_k)\Big)_i \geq 0.$$

Since D_k is positive definite and since $A_2 = A_k^{+c} \setminus A_1$ the relations

$$\begin{aligned} 0 &= \big(P(u_k + \alpha s_u) - u_k\big)^T \Big(T(x_k)^T \nabla f(x_k) + d(x_k)\Big) \\ &= \alpha s_u^T I_{A_2} \big(T(x_k)^T \nabla f(x_k) + d(x_k)\big) \\ &= \alpha s_u^T I_{A_k^{+c}} \big(T(x_k)^T \nabla f(x_k) + d(x_k)\big) - \alpha s_u^T I_{A_1} \big(T(x_k)^T \nabla f(x_k) + d(x_k)\big) \\ &\leq \alpha s_u^T I_{A_k^{+c}} \big(T(x_k)^T \nabla f(x_k) + d(x_k)\big) = -\alpha s_u^T I_{A_k^{+c}} D_k I_{A_k^{+c}} s_u \\ (4.20) \quad &\leq -\alpha \lambda_{\min} ||I_{A_k^{+c}} s_u||^2 \\ &\leq 0 \end{aligned}$$

hold for all $\alpha \in [0, \tilde{\alpha}]$, where $\lambda_{\min} > 0$ is the smallest eigenvalue of $I_{A_k^{+c}} D_k I_{A_k^{+c}}$. Thus,

$$I_{A_k^{+c}} s_u = 0.$$

Since $C_y(x_k)$ is nonsingular this equality implies that $c(x_k) = 0$. With the arguments used to prove the assertion in the case that $(y_k(\alpha), u_k(\alpha))$ is given by (4.9) we can deduce that (y_k, u_k) is a critical point.

(iic) The proof for the cases that $(y_k(\alpha), u_k(\alpha))$ is given by (4.10) or (4.14) follows immediately from (iia,b) and the definition of $(y_k(\alpha), u_k(\alpha))$. □

4.2. A merit function and a step size rule. For the definition of the step size we need a merit function which guarantees that the iterates x_k become feasible and minimize the objective over the feasible set. In our problem we have two sets of constraints. The first set consists of the equality constraints $c(x) = 0$ and the second set is given by the bound constraints $u \geq 0$. Because of the projection the u-component of the iterate is always nonnegative and therefore the second set of constraints is always satisfied. In this situation the merit function only has to guarantee that the equality constraint is satisfied in the limit and that the objective function is minimized. We use a merit function which corresponds to Fletcher's exact merit function (2.11) and which is given by

$$\Phi(x, \rho) = f(x) + \lambda(x)^T c(x) + \frac{\rho}{2}||c(x)||^2. \tag{4.21}$$

The vector $\lambda(x)$ is the Lagrange multiplier estimate defined by (4.16). The fact that the feasible points and the iterates satisfy $u \geq 0$ must be considered when the merit function (4.21) is used. This leads to a constrained merit function. The original optimization problem (1.7) corresponds to

$$\begin{aligned} &\min && \Phi(x, \rho) \\ &\text{s.t.} && u \geq 0. \end{aligned} \tag{4.22}$$

For the analysis of (4.22) we need the derivatives of Φ. Differentiation yields

$$\begin{aligned} \nabla_y(f(x) + \lambda(x)^T c(x)) &= \nabla_y f(x) + C_y(x)\lambda(x) + \lambda_y(x)^T c(x) \;=\; \lambda_y(x)^T c(x), \\ \nabla_u(f(x) + \lambda(x)^T c(x)) &= \nabla_u f(x) + C_u(x)\lambda(x) + \lambda_u(x)^T c(x), \end{aligned}$$

where $\lambda_y(x)$, $\lambda_u(x)$ are obtained from (4.16) by the implicit function theorem and are given by

$$\lambda_y(x) = -C_y(x)^{-1}\nabla_{yy}L(x, \lambda(x)), \quad \lambda_u(x) = -C_y(x)^{-1}\nabla_{yu}L(x, \lambda(x)).$$

Hence,

$$\begin{aligned} \nabla_y(f(x) + \lambda(x)^T c(x)) &= -\nabla_{yy}L(x, \lambda(x))\; C_y(x)^{-T} c(x), \\ \nabla_u(f(x) + \lambda(x)^T c(x)) &= \nabla_u f(x) + C_u(x)\lambda(x) \\ &\quad -\nabla_{yu}L(x, \lambda(x))^T C_y(x)^{-T} c(x), \end{aligned} \tag{4.23}$$

and the gradient of Φ is given by

$$\begin{aligned} \nabla_y \Phi(x, \rho) &= \big(-\nabla_{yy}L(x, \lambda(x)) + \rho\, C_y(x) C_y(x)^T \big) C_y(x)^{-T} c(x), \\ \nabla_u \Phi(x, \rho) &= \nabla_u f(x) + C_u(x)\lambda(x) \\ &\quad + \big(-\nabla_{yu}L(x, \lambda(x))^T C_y(x)^{-T} + \rho\, C_u(x) \big) c(x). \end{aligned} \tag{4.24}$$

We recall the following necessary optimality conditions for (4.22).

THEOREM 4.3. *Let the assumptions* (A1), (A2) *be valid. If* $x_* = (y_*, u_*)$ *is a solution of* (4.22), *then*

$$u_* \geq 0, \quad \nabla_y \Phi(x_*, \rho) = 0, \quad P(u_* - \nabla_u \Phi(x_*, \rho)) = u_*. \tag{4.25}$$

Proof. The proof uses arguments analogous to the ones in the proof of Theorem 4.1. □

The following result exploits the relation between the two problems (1.7) and (4.22).

THEOREM 4.4. *Let* (A1), (A2) *be valid. If* $x_* = (y_*, u_*)$ *is a critical point of* (1.7), *then* x_* *is a critical point of* (4.22). *Moreover, if*

$$\rho > \rho_* = \max\left\{0, \max_{z \neq 0} \frac{z^T \nabla_{yy} L(x_*, \lambda(x_*))z}{||C_y(x_*)^T z||^2}\right\}, \tag{4.26}$$

then a critical point x_* *of* (4.22) *is a critical point of* (1.7).

Proof. Suppose that $x_* = (y_*, u_*), u_* \geq 0$ is a critical point of (1.7). Then $c(x_*) = 0$, (4.24), and Theorem 4.1 show that $\nabla_y \Phi(x_*, \rho) = 0$ and $P(u_* - \alpha \nabla_u \Phi(x_*, \rho)) = P(u_* - \alpha \nabla_u L(x, \lambda(x))) = u_*$ for all $\alpha \geq 0$ and $\rho \geq 0$.

Now suppose that $x_* = (y_*, u_*)$ is a critical point for (4.22). Then it holds that

$$0 = \nabla_y \Phi(x_*, \rho) = \big(- \nabla_{yy} L(x_*, \lambda(x_*)) + \rho C_y(x_*) C_y(x_*)^T \big) C_y(x_*)^{-T} c(x_*).$$

If $\rho > \rho_*$, then $-\nabla_{yy} L(x_*, \lambda(x_*)) + \rho C_y(x_*) C_y(x_*)^T$ is positive definite and the latter equality implies that $c(x_*) = 0$. With this result we find that

$$P(u_* - \nabla_u L(x_*, \lambda(x_*))) = P(u_* - \nabla_u \Phi(x_*, \rho)) = u_*.$$

This proves the assertion. □

The previous theorem focuses on stationary points. In §5 we need a result relating minimizers of (1.7) and minimizers of (4.22). For the statement of second order conditions we need the matrix $\nabla_{xx} \Phi(x_*, \rho)$. If $f, c \in C^3(\mathbb{R}^{m+n})$, then (4.24) and $c(x_*) = 0$ can be used to show that

$$\begin{aligned}
\nabla_{yy} \Phi(x_*, \rho) &= -\nabla_{yy} L_* + \rho C_{y,*} C_{y,*}^T, \\
\nabla_{yu} \Phi(x_*, \rho) &= \big(- \nabla_{yy} L_* + \rho C_{y,*} C_{y,*}^T \big) C_{y,*}^{-T} C_{u,*}^T, \\
\nabla_{uy} \Phi(x_*, \rho) &= \nabla_{yu} \Phi(x_*, \rho)^T, \\
\nabla_{uu} \Phi(x_*, \rho) &= \nabla_{uu} L_* + C_{u,*} \lambda_u(x_*) + \big(- \nabla_{yu} L_*^T C_{y,*}^{-T} + \rho C_{u,*} \big) C_{u,*}^T \\
&= \nabla_{uu} L_* - C_{u,*} C_{y,*}^{-1} \nabla_{yu} L_* - \nabla_{uy} L_* C_{y,*}^{-T} C_{u,*}^T + \rho C_{u,*} C_{u,*}^T,
\end{aligned} \tag{4.27}$$

where $C_{y,*} = C_y(x_*)$, $C_{u,*} = C_u(x_*)$, and $L_* = L(x_*, \lambda(x_*))$.

The following theorem states the necessary and sufficient second order optimality conditions for (1.7) and (4.22).

THEOREM 4.5. *Let the assumptions* (A1), (A2) *be valid.*

(i) *If* $x_* = (y_*, u_*)$ *solves* (1.7), *then the first order conditions* (4.1) *are satisfied and* $s_u^T H(y_*, u_*, \lambda(x_*)) s_u \geq 0$ *for all* s_u *with* $(s_u)_i = 0$, $i \in A(x_*)$.

(ii) *If* $x_* = (y_*, u_*)$ *satisfies* (4.1) *and if* $s_u^T H(y_*, u_*, \lambda(x_*)) s_u > 0$ *for all* s_u *with* $(s_u)_i = 0$, $i \in A(x_*)^+$, $(s_u)_i \geq 0$, $i \in A(x_*) \setminus A(x_*)^+$, *then* x_* *is an isolated local minimizer of* (1.7).

Let (A1), (A2) *be valid, let* $f, c \in C^3(\mathbb{R}^{m+n})$, *and suppose that* $\rho > \rho_*$, *where* ρ_* *is given by* (4.26).

(iii) *If* $x_* = (y_*, u_*)$ *is a local minimum of* (4.22), *then the first order conditions* (4.25) *are satisfied and* $s^T \nabla_{xx} \Phi(x_*, \rho)\, s \geq 0$ *for all* $s^T = (s_y^T, s_u^T)$ *with* $(s_u)_i = 0$, $i \in A(x_*)$.

(iv) *If* $x_* = (y_*, u_*)$ *satisfies the first order necessary optimality conditions* (4.25) *and if* $s^T \nabla_{xx} \Phi(x_*, \rho)\, s > 0$ *for all* $s^T = (s_y^T, s_u^T)$ *with* $(s_u)_i = 0$, $i \in A(x_*)^+$, $(s_u)_i \geq 0$, $i \in A(x_*) \setminus A(x_*)^+$, *then* x_* *is an isolated local minimizer of* (4.22).

(v) *If the conditions in* (iv) *are satisfied at* $x_* = (y_*, u_*)$, *then there exist* $\delta, \sigma > 0$ *such that*

$$\Phi(x,\rho) - \Phi(x_*,\rho) \geq \sigma ||x - x_*||^2 \qquad \forall\, x \in \{(y,u) \mid u \geq 0,\ ||x - x_*|| < \delta\}.$$

Proof. (i) Under the assumptions (A1), (A2) the matrix (4.2) has full rank. Therefore the second order constraint qualifications [16, p. 25] are satisfied. See [16, Thm. 3, p. 26].

By [16, Thm. 2, p. 25] the first order conditions (4.4) are valid and it holds that

$$s^T \nabla_{xx} \bar{L}(x_*, \lambda_*, \mu_*) s \geq 0$$

for all $s = (s_y^T, s_u^T) \in \mathcal{N}(C(x_*)^T)$ with $(s_u)_i = 0,\ i \in A(x_*)$. Using $\mathcal{N}(C(x_*)^T) = \mathcal{R}(T(x_*))$ and (2.9) implies the assertion.

(ii) Using $\mathcal{N}(C(x_*)^T) = \mathcal{R}(T(x_*))$ and (2.9) we can conclude that

$$s^T \nabla_{xx} \bar{L}(x_*, \lambda_*, \mu_*) s > 0$$

for all $s = (s_y^T, s_u^T) \in \mathcal{N}(C(x_*)^T)$ and $(s_u)_i = 0,\ i \in A(x_*)^+$, $(s_u)_i \geq 0,\ i \in A(x_*) \setminus A(x_*)^+$. Moreover,

$$\begin{aligned}\{i \mid (u_*)_i = 0,\ (\mu_*)_i > 0\} &= \{i \mid (u_*)_i = 0,\ (\nabla_u (f(x_*) + \lambda_*^T c(x_*)))_i > 0\} \\ &= \left\{i \mid (u_*)_i = 0,\ \left(T(x_*)^T \nabla_x f(x_*)\right)_i > 0\right\} \ = \ A(x_*)^+;\end{aligned}$$

cf. (4.4), (4.7), (4.17). Therefore the assertion follows from [16, Thm. 4, p. 30].

(iii), (iv) From Theorem 4.4 it follows that $c(x_*) = 0$. Hence,

$$\begin{aligned}\{i \mid (u_*)_i = 0,\ (\nabla_u \Phi(x_*))_i > 0\} &= \{i \mid (u_*)_i = 0,\ (\nabla_u (f(x_*) + \lambda(x_*)^T c(x_*)))_i > 0\} \\ &= \left\{i \mid (u_*)_i = 0,\ \left(T(x_*)^T \nabla_x f(x_*)\right)_i > 0\right\} \ = \ A(x_*)^+;\end{aligned}$$

cf. (4.7), (4.17), (4.24). With this equality, the assertions follow immediately from [16, Thm. 2, p. 25] and [16, Thm. 4, p. 30].

(v) Define $\sigma_1 = \min_{i \in A(x_*)^+} \left(\nabla_u \Phi(x_*, \rho)\right)_i$, $\sigma_2 = ||\nabla_{xx} \Phi(x_*, \rho)||$, and let $\sigma_3 > 0$ satisfy $s^T \nabla_{xx} \Phi(x_*, \rho)\, s > \sigma_3 ||s||^2$ for all $s^T = (s_y^T, s_u^T)$ with $(s_u)_i = 0,\ i \in A(x_*)^+$, $(s_u)_i \geq 0,\ i \in A(x_*) \setminus A(x_*)^+$. Note that $\left(\nabla_u \Phi(x_*, \rho)\right)_i \geq 0$ for all $i \in A(x_*)$ and $\left(\nabla_u \Phi(x_*, \rho)\right)_i > 0$ for all $i \in A(x_*)^+$.

Consider $x = (y, u)$ with $u \geq 0$. For $s = (s_y, s_u) = x - x_*$ it holds that $s = s_1 + s_2$ with $s_1 = (s_y, s_u - I_{A(x_*)^+} s_u)$, $s_2 = (0, I_{A(x_*)^+} s_u)$. Moreover, $(s_u - I_{A(x_*)^+} s_u)_i = 0$ for all $i \in A(x_*)^+$, $(s_u - I_{A(x_*)^+} s_u)_i \geq 0$ for all $i \in A(x_*) \setminus A(x_*)^+$, and $I_{A(x_*)} s_u \geq 0$.

Hence,

$$\begin{aligned}&\Phi(x,\rho) - \Phi(x_*,\rho) \\ &= \nabla_x \Phi(x_*,\rho)^T s + \frac{1}{2} s^T \nabla_{xx} \Phi(x_*,\rho)^T s + o(||s||^2) \\ &= \nabla_u \Phi(x_*,\rho)^T I_{A(x_*)} s_u + \frac{1}{2} s_1^T \nabla_{xx} \Phi(x_*,\rho) s_1 + s_1^T \nabla_{xx} \Phi(x_*,\rho) s_2 \\ &\quad + \frac{1}{2} s_2^T \nabla_{xx} \Phi(x_*,\rho) s_2 + o(||s||^2) \\ &\geq \sigma_1 ||I_{A(x_*)^+} s_u|| + \frac{\sigma_3}{2} ||s_1||^2 - \sigma_2 ||I_{A(x_*)^+} s_u||\ ||s_1|| - \frac{\sigma_2}{2} ||I_{A(x_*)^+} s_u||^2 + o(||s||^2) \\ &\geq \frac{\sigma_3}{2} ||s_1||^2 + \frac{\sigma_3}{2} ||s_2||^2 + o(||s||^2)\end{aligned}$$

for all s with $\sigma_1 \geq \sigma_2 ||s_1|| + (\sigma_2 + \sigma_3)/2 \, ||s_2||$. This implies the desired result. □

The following theorem shows that under certain conditions isolated local minimizers of (1.7) are isolated local minimizers of (4.22) and vice versa.

THEOREM 4.6. *Let* (A1), (A2) *be valid, let* $f, c \in C^3(\mathbb{R}^{m+n})$, *and suppose that* $\rho > \rho_*$, *where* ρ_* *is given by* (4.26).

(i) *If* $x_* = (y_*, u_*)$ *is an isolated local minimum of* (1.7) *for which the conditions in Theorem* 4.5(ii) *are satisfied, then* $x_* = (y_*, u_*)$ *is an isolated local minimum of* (4.22)

(ii) *If* $x_* = (y_*, u_*)$ *is an isolated local minimum of* (4.22) *for which the conditions in Theorem* 4.5(iv) *are satisfied, then* $x_* = (y_*, u_*)$ *is an isolated local minimum of* (1.7).

Proof. From Theorem 4.4 we know that $x_* = (y_*, u_*)$ is a critical point of (1.7) if and only if it is a critical point of (4.22).

If we use (4.27), the definition (2.7) of $T(y,u)$, the definition (2.9) of $H(y,u,\lambda)$, and the partitioning $s = (s_y, s_u)$, then we can show that

$$\begin{aligned} s^T \nabla_{xx} \Phi(x_*, \rho) s = {} & \big(s_y + C_{y,*}^{-T} C_{u,*}^T s_u\big)^T \big(- \nabla_{yy} L_* + \rho \, C_{y,*} C_{y,*}^T\big) \big(s_y + C_{y,*}^{-T} C_{u,*}^T s_u\big) \\ & + s_u^T H(y_*, u_*, \lambda(u_*)) \, s_u. \end{aligned} \tag{4.28}$$

If the conditions of Theorem 4.5(ii) are satisfied, then the positive definiteness of $-\nabla_{yy} L_* + \rho \, C_{y,*} C_{y,*}^T$ for $\rho \geq \rho_*$ and (4.28) imply the validity of the conditions of Theorem 4.5(iv).

On the other hand, if the conditions of Theorem 4.5(iv) are satisfied, then (4.28) with the choice $s_y = -C_{y,*}^{-T} C_{u,*}^T I_{A_*^+ c} \, s_u$ implies that the conditions of Theorem 4.5(ii) are valid. □

As mentioned before, we will use the merit function Φ to define the step size α_k. If (y_k, u_k) is not a critical point we want to construct a new iterate which reduces the value of the merit function. It is well known (cf. [13, §6.3]) that a mere reduction in the merit function does not guarantee convergence. Instead we must achieve a sufficient decrease. In unconstrained optimization such a sufficient decrease condition means that the reduction in the function value is at least as large as a fraction of the reduction achieved in the linearization of the function. If the iterates $(u(\alpha), y(\alpha))$ are defined by (4.9), then, using (4.24), we find that the reduction in the linearization of Φ is given by

$$\begin{aligned} & (y(\alpha) - y)^T \nabla_y \Phi(y,u,\rho) + (u(\alpha) - u)^T \nabla_u \Phi(y,u,\rho) \\ & = -\alpha c(x)^T C_y(x)^{-1} \big(- \nabla_{yy} L(x, \lambda(x)) + \rho \, C_y(x) C_y(x)^T\big) C_y(x)^{-T} c(x) \\ & \quad + \big(P(u + \alpha s_u) - u\big)^T \Big(C_u(x) C_y(x)^{-1} \nabla_{yy} L(x, \lambda(x)) C_y(x)^{-T} c(x) - \rho \, C_u(x) c(x) \\ & \qquad + \nabla_u f(x) + C_u(x) \lambda(x) - \nabla_{yu} L(x, \lambda(x))^T C_y(x)^{-T} c(x) \\ & \qquad + \rho \, C_u(x) c(x) \Big) \\ & = -\alpha c(x)^T C_y(x)^{-1} \big(- \nabla_{yy} L(x, \lambda(x)) + \rho \, C_y(x) C_y(x)^T\big) C_y(x)^{-T} c(x) \\ & \quad + \big(P(u + \alpha s_u) - u\big)^T \Big(T(x)^T \big(\nabla f(x) + \nabla_{xx} L(x, \lambda(x)) r(x)\big)\Big). \end{aligned} \tag{4.29}$$

Since we want to avoid second order information we omit the term $\nabla_{yy} L(x, \lambda(x))$ and, as in the derivation of the iterates, replace $T(x)^T \nabla_{xx} L(x, \lambda(x)) r(x)$ by $d(x)$. As

mentioned before, we expect that eventually the active set A_k^+ will equal the active set at the optimal point u_*. In this case components of u_k corresponding to the active set are zero and it holds that $P(u_k+\alpha s_u)-u_k = \alpha I_{A_k^{+c}} s_u$. Therefore, we split $P(u_k+\alpha s_u)-u_k$ in (4.29) into two vectors containing components corresponding to active and inactive indices, respectively, and replace $I_{A_k^{+c}}\big(P(u_k+\alpha s_u)-u_k\big)$ by $\alpha I_{A_k^{+c}} s_u$. This leads to the following sufficient decrease condition:

$$
\begin{aligned}
&\Phi(y_k,u_k,\rho)-\Phi(y_k(\alpha),u_k(\alpha),\rho)\\
&\geq \sigma\Big(\rho\alpha||c(x_k)||^2-\alpha\big(I_{A^{+c}}s_u\big)^T\big(T(x_k)^T\nabla f(x_k)+d(x_k)\big)\\
&\qquad +\big(u_k-P(u_k+\alpha s_u)\big)^T I_{A_k^+}\big(T(x_k)^T\nabla f(x_k)+d(x_k)\big)\Big),
\end{aligned}
\tag{4.30}
$$

where $\sigma\in(0,1)$. Of course, we also must avoid the step size becoming too small. This can be done by choosing $\beta\in(0,1)$ and computing a step size $\alpha_k=\beta^{m_k}$, where m_k is the smallest integer such that (4.30) holds.

4.3. Formulation of the algorithm and proof of its well posedness. To obtain a more efficient method we need to relax the definition (4.8) of $A^+(x)$. Instead of requiring that the component of the iterate is zero we require only that the component of the iterate is close to zero. This relaxation avoids a zigzagging behavior of the algorithm near the boundary. We set

$$
A_\epsilon^+(x)=\Big\{i\;\Big|\;u_i\in[0,\epsilon],\quad \Big(T(x)^T\nabla_x f(x)+d(x)\Big)_i>0\Big\},
\tag{4.31}
$$

where $\epsilon=\epsilon(x)\geq 0$ satisfies the following condition:

$$
\text{If } x=(y,u) \text{ is a critical point, then } \epsilon=\epsilon(x)=0.
\tag{4.32}
$$

In practical implementations the condition (4.32) can be guaranteed, e.g., if

$$
\epsilon(x)=O\Big(||c(x)||^p+||u-P\big(u+M(x)(T(x)^T\nabla f(x)+d(x))\big)||^p\Big),
$$

where $p>0$ and $M(x)$ is a positive definite matrix which is diagonal with respect to $A(x)$; cf. Theorem 4.2. To prove the well posedness of projected SQP methods it is sufficient to require (4.32). However, to prove convergence results we need to specify $\epsilon(x)$ further. This will be done in §5.

Because of the relaxation of the active set $A_\epsilon^+(x)$ we also must replace the index set $B(x)\subset A^{+^c}(x)$ used in (4.12) and (4.14) by $B_\epsilon(x)\subset A_\epsilon^{+^c}(x)$. However, to guarantee the well posedness of the algorithm we must restrict the choices for $B_\epsilon(x)$. We assume that

$$
B_\epsilon(x)\subset\Big\{i\notin A_\epsilon^+(x)\;\Big|\;u_i=0\ \text{ and }\ (s_u)_i\geq 0,\quad \text{or}\quad u_i>0\Big\}.
\tag{4.33}
$$

For $u=u_k$ we set $A_\epsilon^+(x_k)=A_{\epsilon,k}^+$, $B_\epsilon(x_k)=B_{\epsilon,k}$ and by $A_{\epsilon,k}^{+^c}$ we denote the complement of $A_{\epsilon,k}^+$. In the following the index sets $A_k^+, A_k^{+^c}$ in (4.9) and (4.30) are replaced by $A_{\epsilon,k}^+, A_{\epsilon,k}^{+^c}$ and the index set B_k in (4.12) and (4.14) is replaced by $B_{\epsilon,k}$. This leads to the algorithm stated below. Here we have used an Armijo-like step size rule to guarantee that the step size is chosen as large as possible. However, this step size rule can be replaced by any other one as long as α_k is not chosen unnecessarily small.

ALGORITHM 4.7 (PROJECTED SQP).

Initialization:

Given vectors $x_0 = (y_0, u_0) \in \mathbb{R}^{m+n}$, and $d_0 \in \mathbb{R}^n$,
a positive definite matrix $H_0 \in \mathbb{R}^{n \times n}$,
a scalar $\epsilon_0 \geq 0$, a penalty parameter $\rho > 0$, and $\sigma, \beta \in (0,1)$.
Set $k = 0$.

Step k:

Set $A^+_{\epsilon,k} = A^+_\epsilon(x_k) = \{i \mid u_i \in [0, \epsilon_k], \quad \big(T(x_k)^T \nabla_x f(x_k) + d(x_k)\big)_i > 0\}$.

Compute the Lagrange multiplier estimate λ_k by solving $C_y(x_k)\lambda = -\nabla_y f(x_k)$.
Compute $T(x_k)^T \nabla_x f(x_k) = \nabla_u f(x_k) + C_u(x_k)\lambda_k$.

Compute $D_k = I_{A^+_{\epsilon,k}} \operatorname{diag}(H_k) I_{A^+_{\epsilon,k}} + I_{A^{+c}_{\epsilon,k}} H_k I_{A^{+c}_{\epsilon,k}}$.
Solve $D_k s_u = -\big(T(x_k)^T \nabla_x f(x_k) + d(x_k)\big)$.

Determine the step size $\alpha_k = \beta^{m_k}$, where m_k is the smallest integer satisfying (4.30) with $u_k(\alpha) = P(u_k + \alpha s_u)$, and $y_k(\alpha)$ defined by (4.9), (4.10), (4.12), or (4.14).

Update H_k to obtain a new positive definite matrix H_{k+1}.

Compute $d(x_{k+1})$ and ϵ_{k+1}.

The following theorem shows that the condition (4.30) implies a decrease in the merit function if x_k is not a critical point and that it is actually possible to find step sizes α satisfying (4.30).

THEOREM 4.8. *Suppose that* (A1), (A2) *are valid, that D_k is a positive definite matrix which is diagonal with respect to $A^+_{\epsilon,k}$, that $d(x_k)$ satisfies* (4.7), *and that $\epsilon_k = \epsilon(x_k)$ satisfies* (4.32). *Then the following assertions are true.*

(i) *All three terms in the right-hand side of* (4.30) *are nonnegative. If $\rho > 0$, the right-hand side in* (4.30) *is zero for all $\alpha > 0$ if and only if (y_k, u_k) is a critical point of* (1.7).

(ii) *Suppose that x_k is not a critical point and that $c(x_k) \neq 0$. If $(y_k(\alpha), u_k(\alpha))$ is defined by* (4.9), *then for all ρ_k satisfying*

$$\rho_k > (1-\sigma)^{-1}(\rho_* + K), \tag{4.34}$$

where $K > 0$ is a constant depending on $d(x_k)$, the functions f, c, and their derivatives, there exists $\bar{\alpha}_k$ such that (4.30) *holds for all $\alpha \in (0, \bar{\alpha}_k)$.*

(iii) *Suppose that x_k is not a critical point and that $c(x_k) \neq 0$. If $(y_k(\alpha), u_k(\alpha))$ is defined by* (4.12) *or* (4.14) *and if $\big(P(u + \alpha s_u) - u\big)_i = 0$ for all $i \notin B_{\epsilon,k}$, then for all ρ_k satisfying* (4.34) *there exists $\hat{\alpha}_k$ such that* (4.30) *holds for all $\alpha \in (0, \hat{\alpha}_k)$.*

(iv) *If the vector $d(x_k)$ is given by $d(x_k) = T(x_k)^T \nabla_{xx} L(x_k, \lambda(x_k)) r(x_k)$, then K in* (ii), (iii) *is given by $K = 0$.*

(v) *Suppose that x_k is not a critical point and that $c(x_k) = 0$. If $(y_k(\alpha), u_k(\alpha))$ is defined by* (4.9), (4.10), (4.12), *or* (4.14), *then there exists $\bar{\alpha}_k$ such that* (4.30) *holds for all $\alpha \in (0, \bar{\alpha}_k)$.*

Proof. In this proof we omit the subscript k and simply write (y, u) instead of (y_k, u_k), etc.

(i) Suppose that (y, u) is a critical point. Then it holds that $c(x) = 0$, $P(u + \alpha s_u) = u$, and $\epsilon = 0$. Hence, $I_{A^+_\epsilon} = I_{A^+}$. With (4.20) this implies that $I_{A^+} s_u = 0$. Hence, the right-hand side in (4.30) is zero for all $\alpha > 0$.

Now suppose that (y,u) is not a critical point. From the definition of s_u and the positive definiteness of D_k we obtain that

$$-\alpha s_u^T I_{A_\epsilon^{+c}} \big(T(x)^T \nabla f(x) + d(x)\big) = \alpha s_u^T I_{A_\epsilon^{+c}} D_k I_{A_\epsilon^{+c}} s_u \geq \alpha \lambda_{\min} ||I_{A_\epsilon^{+c}} s_u||^2, \tag{4.35}$$

where $\lambda_{\min} > 0$ is the smallest eigenvalue of D_k. Moreover, since D_k is positive definite and diagonal with respect to A_ϵ^+ it holds that

$$\big(T(x)^T \nabla f(x) + d(x)\big)_i > 0, \quad \big(s_u\big)_i < 0, \quad 0 \leq \big(u - P(u + \alpha s_u)\big)_i \leq -\alpha (s_u)_i \tag{4.36}$$

for all $i \in A_\epsilon^+$ and for all $\alpha > 0$.

This shows that the right-hand side in (4.30) is always nonnegative.

Next we show that the right-hand side is positive. If $c(x) \neq 0$ this is obviously true. If $c(x) = 0$, then it must hold that $P(u + \alpha s_u) \neq u$. To see this note that $P(u + \alpha s_u) = u$ implies

$$0 = \big(P(u + \alpha s_u) - u\big)^T \Big(T(x)^T \nabla f(x) + d(x)\Big) \leq -\alpha \lambda_{\min} ||I_{A_\epsilon^{+c}} s_u||^2 \leq 0;$$

cf. (4.20). Hence, $I_{A_\epsilon^{+c}} s_u = 0$. With $d(x) = 0$ this implies that

$$(T(x)^T \nabla f(x))_i = 0 \qquad \forall\, i \in A_\epsilon^{+^c}.$$

Moreover, the definition of A_ϵ^+, $P(u + \alpha s_u) = u$, and the structure of D_k guarantee that

$$(T(x)^T \nabla f(x))_i = 0 \qquad \forall\, i \in A_\epsilon^+ \cap \{i \mid u_i > 0\}.$$

This shows that if $c(x) = 0$ and $P(u + \alpha s_u) = u$, the conditions (4.6b), (4.6c) are valid, i.e., that (y_k, u_k) is a critical point. Since by assumption (y_k, u_k) is not a critical point this is a contradiction and we must have that $P(u + \alpha s_u) \neq u$.

If $\big(P(u + \alpha s_u) - u\big)_i \neq 0$ for some $i \in A_\epsilon^+$, then

$$\big(u - P(u + \alpha s_u)\big)^T I_{A_\epsilon^+} \big(T(x)^T \nabla f(x) + d(x)\big) > 0;$$

cf. (4.36). Hence, the right-hand side in (4.30) is positive.

If $\big(P(u + \alpha s_u) - u\big)_i = 0$ for all $i \in A_\epsilon^+$, then

$$0 \neq P(u + \alpha s_u) - u = \alpha I_{A_2} s_u \tag{4.37}$$

for all $\alpha \in (0, \tilde{\alpha})$, where

$$\begin{aligned} A_1 &= \Big\{ i \notin A_\epsilon^+ \;\Big|\; (u_k)_i = 0, (s_u)_i \leq 0 \Big\}, \\ A_2 &= \Big\{ i \notin A_\epsilon^+ \;\Big|\; (u_k)_i = 0 \text{ and } (s_u)_i > 0, \text{ or } (u_k)_i > 0 \Big\}, \end{aligned} \tag{4.38}$$

and

$$\tilde{\alpha} = \sup\{\alpha \mid (u_k + \alpha s_u)_i \geq 0 \quad \forall\, i \in A_2\} \geq \frac{1}{||s_u||} \min\{(u_k)_i \mid (u_k)_i > 0\} > 0.$$

Since $A_2 \subset A_\epsilon^{+^c}$ the equation (4.37) implies

$$-\alpha s_u^T I_{A_\epsilon^{+c}} \big(T(x)^T \nabla f(x) + d(x)\big) \geq \alpha \lambda_{\min} ||I_{A_\epsilon^{+c}} s_u||^2 \geq \alpha \lambda_{\min} ||I_{A_2^{+c}} s_u||^2 > 0;$$

cf. (4.35). This proves that the right-hand side in (4.30) is positive.

(ii) Suppose that $c(x) \neq 0$ and that $(u(\alpha), y(\alpha))$ is defined by (4.9). Using (4.29) and the positive definiteness of $\rho_* I - C_y(x)^{-1}\nabla_{yy}L(x,\lambda(x))C_y(x)^{-T}$, where ρ_* is defined in (4.26), we find that

$$
\begin{aligned}
&\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho)\\
&= -(y(\alpha)-y)^T\nabla_y\Phi(y,u,\rho) - (u(\alpha)-u)^T\nabla_u\Phi(y,u,\rho) + R(y(\alpha),u(\alpha),\rho)\\
&= (\rho-\rho_*)\alpha||c(x)||^2 + \big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla f(x) + d(x)\big)\\
&\quad +\alpha\Big(\rho_*||c(x)||^2 - c(x)^T C_y(x)^{-1}\nabla_{yy}L(x,\lambda(x))C_y(x)^{-T}c(x)\Big)\\
&\quad +\big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla_{xx}L(x,\lambda(x))r(x) - d(x)\big) + R(y(\alpha),u(\alpha),\rho)\\
&\geq (\rho-\rho_*)\alpha||c(x)||^2 + \big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla f(x) + d(x)\big)\\
&\quad +\big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla_{xx}L(x,\lambda(x))r(x) - d(x)\big) + R(y(\alpha),u(\alpha),\rho),
\end{aligned}
\tag{4.39}
$$

where $R(y(\alpha),u(\alpha),\rho)$ is a remainder term. Note that the remainder depends on ρ. Because of the assumption that the functions f and c and their derivatives up to second order are Lipschitz continuous the remainder $R(y(\alpha),u(\alpha),\rho)$ obeys

$$
|R(y(\alpha),u(\alpha),\rho)| \leq L\rho\big(||y-y(\alpha)||^2 + ||u-u(\alpha)||^2\big), \tag{4.40}
$$

with some constant $L > 0$. From the definition (4.9) of $y(\alpha), u(\alpha)$ and (4.40) we obtain that

$$
|R(y(\alpha),u(\alpha),\rho)| \leq K_1 L\rho\big(\alpha^2||c(x)||^2 + ||u-u(\alpha)||^2\big), \tag{4.41}
$$

where $K_1 > 0$ is a constant depending on c and its derivatives.

There exists $\gamma > 0$ such that[1]

$$
||d(x)|| \leq \gamma||c(x)||, \tag{4.42}
$$

and from the definition (2.7) of $r(x)$ we obtain the existence of $K_2 > 0$ with

$$
||T(x)^T\nabla_{xx}L(x,\lambda(x))r(x)|| \leq K_2||c(x)||. \tag{4.43}
$$

Combining (4.42) and (4.43) we obtain that

$$
\begin{aligned}
&\big|\big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla_{xx}L(x,\lambda(x))r(x) - d(x)\big)\big|\\
&\leq (K_2+\gamma)\,||u - P(u+\alpha s_u)||\;||c(x)||\\
&\leq \frac{\lambda_{\min}(1-\sigma)}{2\alpha}||u - P(u+\alpha s_u)||^2 + K_3\,\alpha\,||c(x)||^2,
\end{aligned}
\tag{4.44}
$$

where $K_3 = (K_2+\gamma)^2/(2\lambda_{\min}(1-\sigma))$ and $\lambda_{\min}$ is the smallest eigenvalue of D_k.

We introduce the sets

$$
\begin{aligned}
A_3 &= \Big\{i \notin A_\epsilon^+ \,\Big|\, \big(T(x)^T\nabla f(x) + d(x)\big)_i > 0\Big\},\\
A_4 &= \Big\{i \notin A_\epsilon^+ \,\Big|\, \big(T(x)^T\nabla f(x) + d(x)\big)_i \leq 0\Big\},
\end{aligned}
\tag{4.45}
$$

[1] Since $c(x) \neq 0$ equation (4.42) trivially holds with $\gamma = ||d(x)||/||c(x)||$. However, for the convergence analysis carried out in §5 it will be important to choose γ independent of x, cf. Remark 4.11, which is why we introduce the parameter γ. In §5 we will assume that $d(x)$ is chosen such that there exists a constant γ satisfying (4.42) for all x.

and we define

$$\eta = \min\Big\{ -\frac{u_i}{(s_u)_i} \mid i \in A_3,\ (s_u)_i < 0 \Big\}. \tag{4.46}$$

From the definition of A_ϵ^+ and A_3 we find that $u_i > \epsilon_k \geq 0$ for all $i \in A_3$. Hence

$$\eta \geq \frac{1}{||s_u||} \min\{u_i \mid i \in A_3\} > \frac{\epsilon_k}{||s_u||} \geq 0. \tag{4.47}$$

For all $\alpha \leq \eta$ the equality $I_{A_3} P(u + \alpha s_u) = I_{A_3}(u + \alpha s_u)$ is valid. Thus,

$$\big(u - P(u+\alpha s_u)\big)^T I_{A_3}\big(T(x)^T \nabla f(x) + d(x)\big) = -\alpha s_u^T I_{A_3}\big(T(x)^T \nabla f(x) + d(x)\big) \quad \forall\ \alpha \leq \eta. \tag{4.48}$$

Moreover, for all $\alpha \geq 0$ it holds that $u - P(u + \alpha s_u) \leq -\alpha s_u$. Together with the definition of A_4 this implies the inequality

$$\big(u - P(u+\alpha s_u)\big)^T I_{A_4}\big(T(x)^T \nabla f(x) + d(x)\big) \geq -\alpha s_u^T I_{A_4}\big(T(x)^T \nabla f(x) + d(x)\big). \tag{4.49}$$

Combining (4.48) and (4.49) we find that

$$\big(u - P(u+\alpha s_u)\big)^T I_{A_\epsilon^{+c}}\big(T(x)^T \nabla f(x) + d(x)\big) \geq -\alpha s_u^T I_{A_\epsilon^{+c}}\big(T(x)^T \nabla f(x) + d(x)\big) \quad \forall\ \alpha \leq \eta. \tag{4.50}$$

From (4.36), the definition of s_u, and the fact that D_k is diagonal with respect to A_ϵ^+ we find that

$$\begin{aligned} ||I_{A_\epsilon^+}(u - u(\alpha))||^2 &\leq -\alpha (u - u(\alpha))^T I_{A_\epsilon^+} s_u \\ &= \alpha (u - u(\alpha))^T I_{A_\epsilon^+} D_k^{-1} I_{A_\epsilon^+}\big(T(x)^T \nabla f(x) + d(x)\big) \\ &\leq \frac{\alpha}{\lambda_{\min}} (u - u(\alpha))^T I_{A_\epsilon^+}\big(T(x)^T \nabla f(x) + d(x)\big), \end{aligned} \tag{4.51}$$

where $\lambda_{\min} > 0$ is the smallest eigenvalue of D_k.

For all i it holds that $|(u - u(\alpha))_i| \leq \alpha |(s_u)_i|$. Hence

$$||I_{A_\epsilon^{+c}}\big(u - u(\alpha)\big)|| \leq \alpha ||I_{A_\epsilon^{+c}} s_u|| \qquad \forall\ \alpha \in [0,1]. \tag{4.52}$$

Using the definition of s_u we find that

$$\begin{aligned} ||I_{A_\epsilon^{+c}} s_u||^2 &= \big(T(x)^T \nabla f(x) + d(x)\big)^T I_{A_\epsilon^{+c}} D_k^{-2} I_{A_\epsilon^{+c}}\big(T(x)^T \nabla f(x) + d(x)\big) \\ &\leq \lambda_{\min}^{-1}\big(T(x)^T \nabla f(x) + d(x)\big)^T I_{A_\epsilon^{+c}} D_k^{-1} I_{A_\epsilon^{+c}}\big(T(x)^T \nabla f(x) + d(x)\big) \\ &= -\lambda_{\min}^{-1} s_u^T I_{A_\epsilon^{+c}}\big(T(x)^T \nabla f(x) + d(x)\big). \end{aligned} \tag{4.53}$$

Inserting (4.41), (4.44), and (4.50) into (4.39) we find that

$$\begin{aligned} &\Phi(y,u,\rho) - \Phi(y(\alpha), u(\alpha), \rho) \\ &\geq (\rho - \rho_*)\alpha ||c(x)||^2 + \big(u - P(u+\alpha s_u)\big)^T I_{A_\epsilon^+}\big(T(x)^T \nabla f(x) + d(x)\big) \\ &\quad -\alpha s_u^T I_{A_\epsilon^{+c}}\big(T(x)^T \nabla f(x) + d(x)\big) \\ &\quad -K_1 L \rho \alpha^2 ||c(x)||^2 - K_1 L \rho ||u - P(u+\alpha s_u)||^2 \\ &\quad -\frac{\lambda_{\min}(1-\sigma)}{2\alpha} ||u - P(u+\alpha s_u)||^2 - K_3\, \alpha\, ||c(x)||^2 \qquad \forall\ \alpha \leq \eta\,. \end{aligned} \tag{4.54}$$

Finally, inserting (4.51), (4.52), and (4.53) into (4.54) we obtain that

$$
\begin{aligned}
&\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho) \\
&\geq (\rho - \rho_* - K_3 - K_1 L\rho\alpha)\alpha||c(x)||^2 \\
&\quad + \Big(\frac{1+\sigma}{2} - \frac{K_1 L\rho\alpha}{\lambda_{\min}}\Big)\big(u - P(u+\alpha s_u)\big)^T I_{A_\epsilon^+}\big(T(x)^T\nabla f(x) + d(x)\big) \\
&\quad - \Big(\frac{1+\sigma}{2} - \frac{K_1 L\rho\alpha}{\lambda_{\min}}\Big)\alpha s_u^T I_{A_\epsilon^{+c}}\big(T(x)^T\nabla f(x) + d(x)\big) \qquad \forall\, \alpha \leq \eta\,. \qquad (4.55)
\end{aligned}
$$

From this equation it can be seen that (4.30) is satisfied for all

$$
\alpha < \min\left\{\eta,\ \frac{(1-\sigma)\rho - \rho_* - K_3}{K_1 L\rho},\ \frac{(1-\sigma)\lambda_{\min}}{2K_1 L\rho}\right\}, \qquad (4.56)
$$

provided ρ satisfies (4.34) with $K \geq K_3$.

(iiia) Suppose that $c(x) \neq 0$, that $\big(P(u+\alpha s_u) - u\big)_i = 0$ for all $i \notin B_\epsilon$, and that $(u(\alpha), y(\alpha))$ is defined by (4.12).

Using the formulas (4.24) for the gradient of Φ we find that

$$
\begin{aligned}
&(y(\alpha)-y)^T\nabla_y\Phi(y,u,\rho) + (u(\alpha)-u)^T\nabla_u\Phi(y,u,\rho) \\
&= -\alpha c(x)^T C_y(x)^{-1}\big(-\nabla_{yy}L(x,\lambda(x)) + \rho C_y(x)C_y(x)^T\big)C_y(x)^{-T}c(x) \\
&\quad + \big(\alpha I_{B_\epsilon}s_u\big)^T\Big(C_u(x)C_y(x)^{-1}\nabla_{yy}L(x,\lambda(x))C_y(x)^{-T}c(x) - \rho C_u(x)c(x)\Big) \\
&\quad + \big(P(u+\alpha s_u)-u\big)^T\Big(\nabla_u f(x) + C_u(x)\lambda(x) - \nabla_{yu}L(x,\lambda(x))^T C_y(x)^{-T}c(x) \\
&\qquad\qquad + \rho C_u(x)c(x)\Big) \\
&= -\alpha c(x)^T C_y(x)^{-1}\big(-\nabla_{yy}L(x,\lambda(x)) + \rho C_y(x)C_y(x)^T\big)C_y(x)^{-T}c(x) \\
&\quad + \big(P(u+\alpha s_u)-u\big)^T\Big(T(x)^T\big(\nabla f(x) + \nabla_{xx}L(x,\lambda(x))r(x)\big)\Big) \\
&\quad + \Big(\alpha I_{B_\epsilon}s_u - \big(P(u+\alpha s_u)-u\big)\Big)^T C_u(x)\Big(C_y(x)^{-1}\nabla_{yy}L(x,\lambda(x))C_y(x)^{-T} - \rho I\Big)c(x). \qquad (4.57)
\end{aligned}
$$

Analogous to the derivation of (4.39) we can use (4.57) to prove that

$$
\begin{aligned}
&\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho) \\
&\geq (\rho-\rho_*)\alpha||c(x)||^2 + \big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla f(x) + d(x)\big) \\
&\quad + \big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla_{xx}L(x,\lambda(x))r(x) - d(x)\big) \\
&\quad + \Big(\alpha I_{B_\epsilon}s_u - \big(P(u+\alpha s_u)-u\big)\Big)^T C_u(x)\Big(\rho I - C_y(x)^{-1}\nabla_{yy}L(x,\lambda(x))C_y(x)^{-T}\Big)c(x) \\
&\quad + R(y(\alpha),u(\alpha),\rho), \qquad (4.58)
\end{aligned}
$$

where $R(y(\alpha),u(\alpha),\rho)$ is a remainder term and ρ_* is defined in (4.26).

By the definition (4.33) of B_ϵ the equality

$$
I_{B_\epsilon}\big(P(u+\alpha s_u) - u\big) = \alpha I_{B_\epsilon}s_u
$$

holds for all $\alpha \leq \bar{\eta}$, where

$$\bar{\eta} = \min\Big\{ -\frac{u_i}{(s_u)_i} \mid i \in B_\epsilon,\ u_i > 0,\ (s_u)_i < 0 \Big\} > 0. \tag{4.59}$$

Using this equality and the assumption $\big(P(u+\alpha s_u) - u\big)_i = 0$ for all $i \notin B_\epsilon$ we find that

$$\alpha I_{B_\epsilon} s_u - \big(P(u+\alpha s_u) - u\big) = I_{B_\epsilon^c}\big(P(u+\alpha s_u) - u\big) = 0 \qquad \forall\ \alpha \leq \bar{\eta}\,.$$

Hence, the third term in the right-hand side of (4.58) vanishes for $\alpha \leq \bar{\eta}$ and we can apply the result of part (ii) to show that (4.30) is satisfied for all α_k satisfying

$$\alpha < \min\left\{ \eta,\ \bar{\eta},\ \frac{(1-\sigma)\rho - \rho_* - K_3}{K_1 L \rho},\ \frac{(1-\sigma)\lambda_{\min}}{2K_1 L \rho} \right\}. \tag{4.60}$$

(iiib) Suppose that $c(x) \neq 0$, that $\big(P(u+\alpha s_u) - u\big)_i = 0$ for all $i \notin B_{\epsilon,k}$, and that $(u(\alpha), y(\alpha))$ is defined by (4.14).

For $\alpha \leq \tilde{\alpha}_k$, where $\tilde{\alpha}_k$ is defined by (4.13) with B_k replaced by $B_{\epsilon,k}$, the definitions (4.12) and (4.14) are identical. Since $\tilde{\alpha}_k = \bar{\eta}$, where $\bar{\eta}$ is defined in (4.59), the assertion follows from (iiia).

(iv) If $d(x) = T(x)^T \nabla_{xx} L(x, \lambda(x)) r(x)$, then the term on the left-hand side of (4.44) vanishes and we can set $K = K_3 = 0$; cf. (4.56).

(v) Suppose that $c(x) = 0$. In this case it holds that

$$\begin{aligned} \nabla_y \Phi(x,\rho) &= 0, \\ \nabla_u \Phi(x,\rho) &= \nabla_u f(x) + C_u(x)\lambda(x) \ = \ T(x)^T \nabla f(x) \end{aligned}$$

(cf. (4.24)), and we do not have to distinguish among the definitions (4.9), (4.10), (4.12), or (4.14) of $y(\alpha)$.

Analogously to the derivation of (4.39) we obtain the estimate

$$\begin{aligned} &\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho) \\ &= \big(u - P(u+\alpha s_u)\big)^T T(x)^T \nabla f(x) + R(y(\alpha),u(\alpha),\rho), \end{aligned} \tag{4.61}$$

where $R(y(\alpha),u(\alpha),\rho)$ is a remainder term satisfying (4.40). The definition of s_y and $c(x) = 0$ imply the existence of $K_4 > 0$ depending on the derivatives of c such that

$$|R(y(\alpha),u(\alpha))| \leq K_4 L \rho \alpha^2 ||I_{A_\epsilon^{+c}} s_u||^2 + K_4 L \rho ||u - u(\alpha)||^2$$

holds. This estimate and (4.61) imply that

$$\begin{aligned} &\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho) \\ &\geq \big(u - P(u+\alpha s_u)\big)^T T(x)^T \nabla f(x) - K_4 L \rho \alpha^2 ||I_{A_\epsilon^{+c}} s_u||^2 - K_4 L \rho ||u - u(\alpha)||^2. \end{aligned}$$

Using (4.50)–(4.53) with $d(x) = 0$ we find that

$$\begin{aligned} &\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho) \\ &\geq \left(1 - \frac{K_4 L \rho \alpha}{\lambda_{\min}}\right) \big(u - P(u+\alpha s_u)\big)^T I_{A_\epsilon^+} T(x)^T \nabla f(x) \\ &\quad - \left(1 - \frac{2K_4 L \rho \alpha}{\lambda_{\min}}\right) \alpha s_u^T I_{A_\epsilon^{+c}} T(x)^T \nabla f(x) \end{aligned}$$

for all $\alpha \leq \eta$. From this equation we find that (4.30) is satisfied for all

$$\alpha < \min\Big\{\eta, \frac{(1-\sigma)\lambda_{\min}}{2K_4 L\rho}\Big\}. \qquad \square \tag{4.62}$$

Remark 4.9. In part (iii) of the previous theorem we assumed that $\big(P(u_k + \alpha s_u) - u_k\big)_i = 0$ for all $i \notin B_{\epsilon,k}$. This seems to be restrictive. However, if $B_\epsilon(x)$ is chosen such that equality holds in (4.33), then this condition is satisfied if $(u_k)_i = 0$ for all $i \in A^+_{\epsilon,k}$. To see this recall that $(s_u)_i < 0$ for all $i \in A^+_{\epsilon,k}$. Moreover, $\mathbb{R}^n \setminus (A^+_{\epsilon,k} \cup B_{\epsilon,k}) = \{i \notin A^+_{\epsilon,k} \mid (u_k)_i = 0, (s_u)_i \leq 0\}$. Hence $(u_k)_i = 0$, $(s_u)_i \leq 0$ for all $i \notin B_\epsilon$ and we find that $\big(u_k - P(u_k + \alpha s_u)\big)_i = 0$ for all $i \notin B_{\epsilon,k}$.

In Algorithm 4.7 we can enforce the condition $(u_k)_i = 0$ for all $i \in A^+_{\epsilon,k}$ by setting the corresponding components of u_k to zero after the determination of $A^+_{\epsilon,k}$.

Remark 4.10. The previous theorem does not cover the definition (4.10) of $y_k(\alpha)$ and we will comment on the theoretical difficulties related to this choice.

Suppose that $c(x) \neq 0$ and that $(u(\alpha), y(\alpha))$ is defined by (4.10). Analogously to the derivation on (4.58) we can prove that

$$\begin{aligned}
&\Phi(y,u,\rho) - \Phi(y(\alpha),u(\alpha),\rho)\\
&\geq (\rho-\rho_*)\alpha||c(x)||^2 + \big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla f(x) + d(x)\big)\\
&\quad + \big(u - P(u+\alpha s_u)\big)^T\big(T(x)^T\nabla_{xx}L(x,\lambda(x))r(x) - d(x)\big)\\
&\quad + \Big(\alpha\big(P(u+s_u)-u\big) - \big(P(u+\alpha s_u)-u\big)\Big)^T\\
&\qquad C_u(x)\Big(\rho I - C_y(x)^{-1}\nabla_{yy}L(x,\lambda(x))C_y(x)^{-T}c(x)\Big)c(x) + R(y(\alpha),u(\alpha),\rho)\,.
\end{aligned} \tag{4.63}$$

The difficulties are caused by the third term in the right-hand side in (4.63).

By the definition of A^+_ϵ and s_u it holds that $(s_u)_i < 0$ for all $i \in A^+_\epsilon$. If $(u)_i = 0$ for all $i \in A^+_\epsilon$ this implies that $\alpha I_{A^+_\epsilon}\big(P(u+s_u)-u\big) = I_{A^+_\epsilon}\big(P(u+\alpha s_u)-u\big) = 0$ for all $\alpha \geq 0$. Hence,

$$\alpha\big(P(u+s_u)-u\big) - \big(P(u+\alpha s_u)-u\big) = \alpha I_{A^{+c}_\epsilon}\big(P(u+s_u)-u\big) - I_{A^{+c}_\epsilon}\big(P(u+\alpha s_u)-u\big).$$

If $I_{A^{+c}_\epsilon}u + I_{A^{+c}_\epsilon}s_u \geq 0$, a condition which is satisfied for example if $||I_{A^{+c}_\epsilon}s_u|| \leq \eta_2 = \min\{u_i \mid i \notin A^+_\epsilon\}$, then $\alpha\big(P(u+s_u)-u\big) = \big(P(u+\alpha s_u)-u\big)$ for all $\alpha \in [0,1]$. However, this condition can be expected to hold only locally. Usually we have that $\alpha\big(P(u+s_u)-u\big) - \big(P(u+\alpha s_u)-u\big) = O(\alpha)$; cf. Figure 4.1. In this case the term $\rho\alpha\big(P(u+s_u)-u\big) - \big(P(u+\alpha s_u)-u\big)^T C_u(x)c(x)$ cannot be made small compared to $(\rho-\rho_*)\alpha||c(x)||^2$. The sum of these two terms cannot be guaranteed to be bigger than $\sigma\rho\alpha||c(x)||^2$; cf. (4.63).

This difficulty always occurs if the term $P(u+\alpha s_u) - u$ in the definition (4.9) of $y(\alpha)$ is replaced by another term. This is also the reason why the assumption $\big(P(u+\alpha s_u)-u\big)_i = 0$ for all $i \notin B_\epsilon$ had to be imposed in part (iii) of the previous theorem.

Remark 4.11. The constants γ and $K_1, \ldots, K_4$ in the proof of the previous theorem depend on the functions f, c and their derivatives evaluated at x_k. If these constants can be chosen independent of k, if the step s_u is bounded for all k, and if the scalar η in (4.46) is uniformly bounded away from zero, then the right-hand sides in (4.56), (4.62) are bounded away from zero. If in addition the scalar $\bar{\eta}$ in (4.59) is

uniformly bounded away from zero, then the estimate (4.60) is bounded away from zero.

In the following section we will investigate the convergence properties of these methods.

5. Convergence of projected SQP methods.

5.1. Global convergence of projected SQP methods and the identification of active indices. In this section we prove global convergence of the projected SQP method, Algorithm 4.7, and discuss the identification of the active indices. In addition to the general assumptions (A1), (A2) we need the following assumptions.

(A3) The penalty parameter ρ is fixed and satisfies (4.34). The level set

$$\mathcal{L}_0 = \{(y,u) \mid u \geq 0, \Phi(y,u,\rho) \leq \Phi(y_0,u_0,\rho)\}$$

is bounded.

(A4) Let $\{x_k\}$ be the sequence of iterates generated by Algorithm 4.7 and let $\{D_k\}$ be the corresponding sequence of projected reduced Hessians. Then the matrices D_k are symmetric and diagonal with respect to $A^+_{\epsilon(x_k)}$. Furthermore, they are uniformly bounded and are uniformly positive definite, i.e., there exists $\lambda_{\max} \geq \lambda_{\min} > 0$ such that

$$\lambda_{\min}||u||^2 \leq u^T D_k u \leq \lambda_{\max}||u||^2 \qquad \forall\, k \in I\!N.$$

(A5) There exists $\gamma_1 > 0$ such that

$$||d(x_k)|| \leq \gamma_1 ||c(x_k)|| \qquad \forall\, k \in I\!N.$$

(A6) There exist diagonal matrices $M(x_k)$ satisfying

$$\lambda^M_{\min}||u||^2 \leq u^T M(x_k) u \leq \lambda^M_{\max}||u||^2 \qquad \forall\, k \in I\!N,$$

with $\lambda^M_{\max} \geq \lambda^M_{\min} > 0$. Moreover, there exist scalars $\gamma_3 > \gamma_2 \geq 1$ such that

$$\begin{aligned}&\gamma_2\Big(||c(x_k)|| + ||u_k - P(u_k - M(x_k)(T(x_k)^T \nabla f(x_k) + d(x_k)))||\Big)\\ &\leq \epsilon(x_k) \leq \gamma_3\Big(||c(x_k)|| + ||u_k - P(u_k - M(x_k)(T(x_k)^T \nabla f(x_k) + d(x_k)))||\Big)\\ &\quad \forall\, k \in I\!N.\end{aligned}$$

(A7) The matrices $C_y(x)^{-1}$ are uniformly bounded on $\{(y,u) \mid u \geq 0\} \cap \mathcal{L}_0$.

The assumptions (A5) and (A6) imply (4.7) and (4.32), respectively. Therefore, Algorithm 4.7 is well defined under the assumptions (A1), (A2), and (A4)–(A6). Since the matrix D_k is a projection of H_k the assumption (A4) is implied by the following.

(A4′) The eigenvalues of the matrix H_k are uniformly bounded away from zero for all $k \in I\!N$, i.e., there exists $\lambda_{\max} \geq \lambda_{\min} > 0$ such that

$$\lambda_{\min}||u||^2 \leq u^T H_k u \leq \lambda_{\max}||u||^2 \qquad \forall\, k \in I\!N.$$

We assume that equality holds in (4.33), i.e., that

$$B_\epsilon(x) = \Big\{ i \notin A^+_\epsilon(x) \;\Big|\; u_i = 0 \text{ and } (s_u)_i \geq 0, \quad \text{or} \quad u_i > 0 \Big\}. \tag{5.1}$$

In this case the assumption that $\big(P(u+\alpha s_u)-u\big)_i = 0$ for all $i \notin B_{\epsilon,k}$ in Theorem 4.8(iii) holds true if $u_i = 0$ for all $i \in A_\epsilon^+(x)$; cf. Remark 4.9.

The assumptions (A1)–(A7) imply the existence of $K > 0$ with

$$\|s_{u,k}\| \le K \qquad \forall\, k \in \mathbb{N} \tag{5.2}$$

and the uniform boundedness of $c(x), C_y(x), C_u(x), \lambda(x), \nabla_{yy}L(x,\lambda(x)), \nabla_{yu}L(x,\lambda(x)), \nabla_{uu}L(x,\lambda(x))$ on $\mathcal{L}_0$. By assumption (A5) the estimate (4.42) is valid unformly for all k. Hence, the constants $K_1,\dots,K_4$ in the proof of Theorem 4.8 are independent of k. If the quantities η and $\bar\eta$ defined in (4.46) and (4.59), respectively, are eventually bounded away from zero, then the right-hand sides in (4.56), (4.60), and (4.62) are uniformly bounded away from zero. Hence, the step sizes α_k are uniformly bounded away from zero; cf. Remark 4.11. We formulate this as a lemma.

LEMMA 5.1. *Suppose that the assumptions* (A1)–(A7) *are valid. Let* $\{x_k\}$ *be the sequence of iterates generated by Algorithm* 4.7 *and define*

$$\eta_k = \min\left\{ -\frac{(u_k)_i}{(s_{u,k})_i} \;\middle|\; i \in \left\{\iota \notin A_{\epsilon,k}^+ \middle| \big(T(x_k)^T\nabla f(x_k) + d(x_k)\big)_\iota > 0,\ (s_{u,k})_\iota < 0\right\}\right\},$$

$$\bar\eta_k = \min\left\{ -\frac{(u_k)_i}{(s_{u,k})_i} \;\middle|\; i \notin A_{\epsilon,k}^+,\ (u_k)_i > 0,\ (s_{u,k})_i < 0\right\}.$$

If the steps in Algorithm 4.7 *are defined by* (4.9) *and if* $\liminf_{j\to\infty}\eta_{k_j} > 0$, *then there exists* $\underline{\alpha} > 0$ *with* $\alpha_{k_j} \ge \underline{\alpha}$ *for all* j.

Suppose that $B_\epsilon(x)$ *is given by* (4.33) *and that* $u_i = 0$ *for all* $i \in A_\epsilon^+(x)$. *If the steps in Algorithm* 4.7 *are defined by* (4.12) *or* (4.14) *and if* $\liminf_{j\to\infty}\eta_{k_j} > 0$, $\liminf_{j\to\infty}\bar\eta_{k_j} > 0$, *then there exists* $\underline{\alpha} > 0$ *with* $\alpha_{k_j} \ge \underline{\alpha}$ *for all* j.

We recall that

$$\eta_k > \frac{\epsilon_k}{\|s_{u,k}\|} \ge 0; \tag{5.3}$$

cf. (4.47). This property will be used to prove the following global convergence result.

THEOREM 5.2 (global convergence). *Suppose that the assumptions* (A1)–(A7) *are valid. If* $\{x_k\}$ *is the sequence generated by Algorithm* 4.7 *with* (4.9), *then*

$$\lim_{k\to\infty}\|c(x_k)\| = 0 \tag{5.4}$$

and every limit point of the sequence $\{x_k\}$ *is a critical point.*

Proof. Since all terms in the right-hand side of (4.30) are nonnegative and since $\Phi(y_k,u_k,\rho)$ is monotonically decreasing and bounded from below we find that

$$\begin{aligned}
&\lim_{k\to\infty}\alpha_k\|c(x_k)\|^2 = 0,\\
&\lim_{k\to\infty}\alpha_k\big(I_{A_{k,\epsilon}^{+c}}s_u\big)^T\big(T(x_k)^T\nabla f(x_k) + d(x_k)\big) = 0,\\
&\lim_{k\to\infty}\big(u_k - P(u_k+\alpha_k s_u)\big)^T I_{A_{k,\epsilon}^{+}}\big(T(x_k)^T\nabla f(x_k) + d(x_k)\big) = 0.
\end{aligned} \tag{5.5}$$

(i) Assume that there exists a subsequence $\{k_j\}$ and a positive constant $\underline{c} > 0$ such that

$$\|c(x_{k_j})\| \ge \underline{c} > 0 \qquad \forall j.$$

Then $\lim_{j\to\infty} \alpha_{k_j} = 0$. Because of assumption (A6) and the estimates (5.2), (5.3), the scalar η_{k_j} satisfies

$$\eta_{k_j} > \frac{\epsilon_{k_j}}{||s_{u,k_j}||} \geq \frac{\gamma_2 \underline{c}}{K} \qquad \forall\, k_j.$$

Hence, by Lemma 5.1 there exists an $\underline{\alpha}$ with

$$\alpha_{k_j} \geq \underline{\alpha}.$$

This contradicts $\lim_{j\to\infty} \alpha_{k_j} = 0$ and therefore (5.4) is true.

(ii) Assume there exists a limit point x_* of the sequence which is not a critical point. Then there exists i_0 such that

$$u_{*,i_0} = 0, \qquad \big(T(x_*)^T \nabla f(x_*)\big)_{i_0} < 0 \tag{5.6}$$

or

$$u_{*,i_0} > 0, \qquad \big(T(x_*)^T \nabla f(x_*)\big)_{i_0} \neq 0. \tag{5.7}$$

Let $\{k_j\}$ be the subsequence with $\lim_{j\to\infty} x_{k_j} = x_*$.

The conditions (5.6) and (5.7) are equivalent to $u_* - P(u_* + M(x_*)T(x_*)^T \nabla f(x_*)) \neq 0$. From this equation, part (i), and the assumptions (A5), (A6) we find that there exist $j_* \in I\!N$ and $\underline{\epsilon} > 0$ such that

$$\epsilon(x_{k_j}) \geq \underline{\epsilon} \qquad \forall j \geq j_*.$$

Because of assumption (A6) and the estimates (5.2), (5.3) this implies the estimate

$$\eta_{k_j} > \frac{\epsilon_{k_j}}{||s_{u,k_j}||} \geq \frac{\underline{\epsilon}}{K} \qquad \forall\, k_j.$$

Lemma 5.1 implies the existence of $\underline{\alpha} > 0$ such that

$$\alpha_{k_j} \geq \underline{\alpha} \qquad \forall j \geq j_*. \tag{5.8}$$

If there exists an infinite set of integers J such that $i_0 \in A^{+^c}_{k_j,\epsilon}$ for all $j \in J$, then $\big(T(x_*)^T \nabla f(x_*)\big)_{i_0} \neq 0$ and (A4) imply the existence of β such that $||I_{A^{+^c}_{\epsilon}} s_{u,k}|| \geq \beta$. The estimate (4.35) and (5.5) imply that $\lim_{j\to\infty} \alpha_{k_j} = 0$, which is a contradiction to (5.8).

If there exist an infinite set of integers J such that $i_0 \in A^{+}_{k_j,\epsilon}$ for all $j \in J$, then the definition of $A^{+}_{k_j,\epsilon}$ and (A5) imply that (5.7) must hold. More precisely, we find that

$$u_{*,i_0} > 0, \qquad \big(T(x_*)^T \nabla f(x_*)\big)_{i_0} > 0.$$

Since D_{k_j} is diagonal with respect to $A^{+}_{k_j,\epsilon}$ the assumptions (A4) and (A5) imply that

$$(s_{u,k_j})_{i_0} \leq -\lambda_{\max}^{-1} \big(T(x_{k_j})^T \nabla f(x_{k_j}) + d(x_{k_j})\big)_{i_0} \leq -\frac{1}{2\lambda_{\max}} \big(T(x_*)^T \nabla f(x_*)\big)_{i_0} < 0$$

for all j sufficiently large. By (4.36) we have that

$$\begin{aligned} &\big(u_{k_j} - P(u_{k_j} + \alpha_{k_j} s_{u,k_j})\big)^T I_{A^{+}_{k_j,\epsilon}} \big(T(x_{k_j})^T \nabla f(x_{k_j}) + d(x_{k_j})\big) \\ &\geq \big(u_{k_j} - P(u_{k_j} + \alpha_{k_j} s_{u,k_j})\big)_{i_0} \big(T(x_{k_j})^T \nabla f(x_{k_j}) + d(x_{k_j})\big)_{i_0}. \end{aligned}$$

Since $\left(T(x_{k_j})^T\nabla f(x_{k_j})+d(x_{k_j})\right)_{i_0} \geq \frac{1}{2}\left(T(x_*)^T\nabla f(x_*)\right)_{i_0} > 0$ for all sufficiently large j the previous inequality and (5.5) imply that

$$\lim_{j\to\infty}\left(u_{k_j} - P(u_{k_j}+\alpha_{k_j}s_{u,k_j})\right)_{i_0} = 0.$$

Since $(s_{u,k_j})_{i_0} < 0$ for all j sufficiently large the equality $\lim_{j\to\infty}\alpha_{k_j} = 0$ must hold. This contradicts (5.8). □

If we use Algorithm 4.7 with iterates defined by (4.12) or (4.14), then the $\bar{\eta}_k$ must also eventually be bounded away from zero. This is satisfied, e.g., if the $(u_k)_i$ are eventually strictly positive for all $i \in B_{k,\epsilon}$. In our numerical experiments this was always the case. Moreover, this will be the case under the assumptions made to prove the identification of active indices. In this case the following theorem is valid, which can be proven analogously to the previous one.

THEOREM 5.3. *Suppose that the assumptions* (A1)–(A7) *are valid. Let* $B_\epsilon(x)$ *be given by* (4.33) *and let* $\{x_k\}$ *be the sequence generated by Algorithm* 4.7 *with* (4.12) *or* (4.14).

If $u_i = 0$ *for all* $i \in A_\epsilon^+(x)$ *and if* $\liminf_{j\to\infty}\bar{\eta}_{k_j} > 0$, *then* $\lim_{k\to\infty}||c(x_k)|| = 0$ *and every limit point of the sequence* $\{x_k\}$ *is a critical point.*

Proof. The proof is identical to the proof of Theorem 5.2. □

The next theorem is concerned with the identification of the active indices. This result is important for the characterization of the local convergence of the algorithm. If the active indices are identified, then the projected SQP method is equivalent to the SQP method restricted to the subset of inactive indices.

To prove the identification of the active indices we need the following nondegeneracy condition to be satisfied at the critical point x_* under investigation:

$$\text{(A8)} \qquad A(x_*) = A(x_*)^+ = \left\{i \;\middle|\; (u_*)_i = 0, \quad \left(T(x_*)^T\nabla_x f(x_*)\right)_i > 0\right\}.$$

THEOREM 5.4 (identification of active indices). *Suppose that* (A1)–(A7) *are valid, that* $x_* = (y_*, u_*)$ *is a critical point satisfying* (A8), *and that* $B_\epsilon(x)$ *is given by* (5.1).

(i) *There exists* $\delta > 0$ *such that* $||x - x_*|| \leq \delta$ *implies* $A(x_*) = A^+_{\epsilon(x)}$ *and* $A(x_*)^c = A^{+^c}_{\epsilon(x)} = B_{\epsilon(x)}$.

(ii) *Let* $\{x_k\}$ *be the sequence of iterates defined by Algorithm* 4.7 *with* (4.9), (4.12), *or* (4.14). *If, in addition to the previous assumptions,* $f, c \in C^3(\mathcal{L}_0)$ *and* x_* *is an isolated stationary point and local minimum for which the conditions in Theorem* 4.5(ii) *are satisfied, then there exists* $\delta > 0$ *such that if*

$$||x_{\bar{k}} - x_*|| \leq \delta$$

for some $\bar{k}$, *then* x_k *converges to* x_* *and*

$$A^+_{k,\epsilon} = B_{k,\epsilon} = A(x_k) = A(x_*)$$

for all $k \geq \bar{k}$. *Moreover, the step size is bounded away from zero, i.e., there exists* $\bar{\alpha}$ *with* $\alpha_k \geq \bar{\alpha}$ *for all* $k \geq \bar{k}$. *For sufficiently large* k *the iterates defined by* (4.9), (4.12), *or* (4.14) *obey*

$$\begin{aligned} I_{A^c(x_*)}u_{k+1} &= I_{A^c(x_*)}u_k + \alpha_k I_{A^c(x_*)}s_u, \\ I_{A(x_*)}u_{k+1} &= 0, \\ y_{k+1} &= y_k - \alpha_k C_y(x_k)^{-1}\left(c(x_k) + C_u(x_k)I_{A^c(x_*)}s_u\right). \end{aligned} \tag{5.9}$$

Proof. (i) Define

$$\beta_1 = \min\{(u_*)_i \mid i \notin A(x_*)\},$$
$$\beta_2 = \min\{(T(x_*)^T \nabla f(x_*))_i \mid i \in A(x_*)\},$$

and

$$\delta_1 = \min\{\beta_1, \lambda_{\min}^M \beta_2\}.$$

Because of (A8) δ_1 is positive. The assumptions (A4)–(A8) guarantee the existence of $\delta_2 < \delta_1/2$ with

$$\begin{aligned} &\epsilon(x) \le \delta_1/2, \\ &\lambda_{\min}^M \Big(T(x)^T \nabla_x f(x) + d(x)\Big)_i > \delta_1/2 \qquad \forall\, i \in A(x_*) \end{aligned} \tag{5.10}$$

for all x with $||x - x_*|| \le \delta_2$.

If $||x - x_*|| \le \delta_2$ and $i \notin A(x_*)$, then

$$u_i \ge (u_*)_i - ||x - x_*|| > \delta_1 - \delta_1/2 \ge \epsilon(x). \tag{5.11}$$

On the other hand, if $i \in A(x_*)$, then

$$|u_i| = |u_i - (u_*)_i| \le ||x - x_*|| \le \delta_2 \le \delta_1/2$$

and the second equation in (5.10) shows that

$$P\big(u - M(x)(T(x)^T \nabla_x f(x) + d(x))\big)_i = 0 \quad i \in A(x_*), \tag{5.12}$$

provided $||x - x_*|| \le \delta_2$. Thus the assumption (A6) implies that if $||x - x_*|| \le \delta_2$, then

$$|u_i| = |\big(u - P(u - M(x)(T(x)^T \nabla_x f(x) + d(x)))\big)_i| \le \epsilon(x) \qquad \forall\, i \in A(x_*).$$

This shows that the implication

$$||x - x_*|| \le \delta_2 \Longrightarrow A(x_*) = A_\epsilon^+(x),\ A^c(x_*) = A_\epsilon^{+^c}(x) \tag{5.13}$$

is valid. Moreover, (5.11) shows that $||x - x_*|| \le \delta_2$ and $i \in A(x_*)^c = A_\epsilon^{+^c}(x)$ imply $u_i > \epsilon(x) \ge 0$. Hence, $B_\epsilon(x) = A_\epsilon^{+^c}(x) = A^c(x_*)$ for all x with $||x - x_*|| \le \delta_2$.

(ii) We use the parameters introduced in part (i). From Theorem 4.5 we know that if x_* is an isolated local minimum, then there exist $\delta_3 \le \delta_2$ and $\sigma > 0$ such that

$$\Phi(x, \rho) - \Phi(x_*, \rho) \ge \sigma ||x - x_*||^2 \qquad \forall\, x \in \{(y, u) \mid u \ge 0,\ ||x - x_*|| < \delta_3\}. \tag{5.14}$$

Since x_* is an isolated stationary point we can assume that x_* is the only critical point in $\{x \mid u \ge 0, ||x - x_*|| \le \delta_3\}$. Moreover, from part (i) we know that $A_{k,\epsilon}^+ = A(x_*)$ for all x_k with $||x_k - x_*|| \le \delta_3$. Since $(u_k)_i \ge \delta_1/2$ for all x_k with $||x_k - x_*|| \le \delta_3$ and all $i \in A_{k,\epsilon}^{+^c} = B_{k,\epsilon} = A^c(x_*)$, the scalars η_k and $\bar{\eta}_k$ in Lemma 5.1 obey

$$\eta_k \ge \frac{\delta_1}{2||s_{u,k}||} \ge \frac{\delta_1}{2K}, \quad \bar{\eta}_k \ge \frac{\delta_1}{2||s_{u,k}||} \ge \frac{\delta_1}{2K}.$$

Thus, by Lemma 5.1, there exists $\underline{\alpha} > 0$:

$$\alpha_k \ge \underline{\alpha} \tag{5.15}$$

for all x_k with $||x_k - x_*|| \leq \delta_3$.

Using $I_{A^c(x_*)} T(x_*)^T \nabla f(x_*) = 0$, $c(x_*) = 0$, and the assumptions (A4), (A5), one can establish the existence of $\delta_4 \leq \delta_3$ such that

$$I_{A^c(x_*)}\big(P(u_k + \alpha_k s_{u,k}) - u_k\big) = \alpha_k I_{A^c(x_*)} s_{u,k}$$

for all x_k with $||x_k - x_*|| \leq \delta_4$ and for all $\alpha_k \in [0,1]$. Moreover, $(s_{u,k})_i \geq \lambda_{\min}^{-1}\beta_2/2$ for all $i \in A_{k,\epsilon}^+$ and all $||x - x_*|| \leq \delta_4$. With $A_{k,\epsilon}^{+^c} = B_{k,\epsilon} = A^c(x_*)$ this implies

$$\begin{aligned} I_{A^c(x_*)} u_{k+1} &= I_{A^c(x_*)} u_k + \alpha_k I_{A^c(x_*)} s_u, \\ I_{A(x_*)} u_{k+1} &= 0 \;=\; I_{A(x_*)} u_*, \\ y_{k+1} &= y_k - \alpha_k C_y(x_k)^{-1}\big(c(x_k) + C_u(x_k) I_{A^c(x_*)} s_u\big) \end{aligned}$$

for all x_k with $||x_k - x_*|| \leq \delta_4$, where u_{k+1}, y_{k+1} are defined by (4.9), (4.12), or (4.14).

From the definition of s_u and (A4), (A5) we can deduce the existence of $\delta_5 \leq \delta_4/2$ such that

$$||s_u|| + ||c(x_k)|| \leq \frac{\delta_4}{2L} \qquad \forall\, ||x_k - x_*|| \leq \delta_5,$$

where L is some Lipschitz constant.

From the representation of the iterates (5.9) we find that

$$||x_{k+1} - x_*|| \leq ||x_k - x_*|| + L(||s_u|| + ||c(x_k)||) \leq \delta_4 \leq \delta_3 \qquad \forall\, ||x_k - x_*|| \leq \delta_5.$$

Let

$$||x_{\bar{k}} - x_*|| \leq \min\{\delta_5, \sigma\delta_5^2/L_\Phi\},$$

where L_Φ is the Lipschitz constant of $\Phi(\cdot,\cdot,\rho)$. Then, using (5.14), we find that

$$\begin{aligned} \sigma||x_{\bar{k}+1} - x_*||^2 &\leq \Phi(y_{\bar{k}+1}, u_{\bar{k}+1}, \rho) - \Phi(y_*, u_*, \rho) \\ &\leq \Phi(y_{\bar{k}}, u_{\bar{k}}, \rho) - \Phi(y_*, u_*, \rho) \leq L_\Phi ||x_{\bar{k}} - x_*|| \leq \sigma\delta_5^2. \end{aligned}$$

Hence $||x_{\bar{k}+1} - x_*|| \leq \delta_5$. Now we can use induction and the fact that $\Phi(y_k, u_k, \rho) \leq \Phi(y_{\bar{k}}, u_{\bar{k}}, \rho)$ for all $k \geq \bar{k}$ to show that

$$\begin{aligned} \sigma||x_{k+1} - x_*||^2 &\leq \Phi(y_{k+1}, u_{k+1}, \rho) - \Phi(y_*, u_*, \rho) \\ &\leq \Phi(y_{\bar{k}}, u_{\bar{k}}, \rho) - \Phi(y_*, u_*, \rho) \leq L_\Phi ||x_{\bar{k}} - x_*|| \leq \sigma\delta_5^2 \end{aligned}$$

for all $k \geq \bar{k}$. In particular we find that $||x_k - x_*|| \leq \delta_5 \leq \delta_3$ for all $k \geq \bar{k}$. Since x_* is the only critical point in $\{x \mid u \geq 0, ||x - x_*|| \leq \delta_3\}$, the global convergence results of Theorems 5.2 and 5.3 imply that $x_k \to x_*$. □

The previous theorem shows that under appropriate assumptions the active indices are identified. If this is the case, then the projected SQP method is equivalent to an SQP method restricted to the subset of inactive indices. Moreover, the second part of the previous theorem also shows that the definitions (4.9), (4.11), and (4.14) for the y-component of the iterate are locally equivalent.

5.2. Local convergence of projected SQP methods. The identification of the active indices can also be used to derive local convergence estimates. In fact, since after the active indices are identified the projected SQP method is equivalent to an SQP method restricted to the subset of inactive indices, the local convergence behavior is determined by the choice of the approximations H_k and d_k. For example, if exact second order information is used, i.e., if $H_k = T(x_k)^T \nabla^2_{xx} L(x_k, \lambda_k) T(x_k)$ and $d(x_k) = T(x_k)^T \nabla^2_{xx} L(x_k, \lambda_k) r(x_k)$, then one expects locally q-quadratic convergence.

One can also extend the quasi-Newton updates used in Algorithm 2.1. Again, the key observation is that if the active indices are identified, then the projected SQP method reduces to an SQP method in the variables $(y, I_{A_*^c} u)$. The components of u corresponding to active indices are zero: $I_{A_*} u = 0$. Computing the reduced gradient and the reduced Hessian for the variables $(y, I_{A_*^c} u)$ yields

$$I_{A_*^c} T(x)^T \nabla_x f(x), \quad I_{A_*^c} T(x)^T \nabla^2_{xx} L(x, \lambda)\, T(x) I_{A_*^c}. \tag{5.16}$$

An appropriate generalization of the BFGS update used in Algorithm 2.1 is given by

$$H_{k+1} = \begin{cases} H_k + \dfrac{vv^T}{v^T I_{A_*^c} s_u} - \dfrac{(H_k I_{A_*^c} s_u)(H_k I_{A_*^c} s_u)^T}{s_u^T I_{A_*^c} H_k I_{A_*^c} s_u} & \text{if } v^T I_{A_*^c} s_u > 0, \\ H_k & \text{else}, \end{cases} \tag{5.17}$$

where

$$v = T(x_k + T(x_k) I_{A_*^c} s_u)^T \nabla f(x_k + T(x_k) I_{A_*^c} s_u) - T(x_k)^T \nabla f(x_k). \tag{5.18}$$

It is easy to verify that H_{k+1} satisfies the secant equation

$$I_{A_*^c} H_{k+1} I_{A_*^c} s_u = I_{A_*^c} v$$

and, using a proof similar to the one for Lemma 9.2.1 in [13], that

$$z^T I_{A_*^c} H_{k+1} I_{A_*^c} z > 0 \qquad \forall\, I_{A_*^c} z \neq 0,$$

provided $z^T I_{A_*^c} H_k I_{A_*^c} z > 0$ for all $I_{A_*^c} z \neq 0$ and $v^T I_{A_*^c} s_u > 0$.

The update (5.17), (5.18) is derived using the observation that locally the bound constraint minimization problem reduces to a minimization problem in the variables $(y, I_{A_*^c} u)$ without bound constraints. In a practical algorithm, however, the transition from global to local convergence is not controlled explicitly, but done automatically by the iteration procedure. Therefore, we must replace (5.17), (5.18) by updates avoiding the active set A_*^c. Because of the result established in Theorem 5.4(ii), this can be done. Under the assumptions of Theorem 5.4(ii) the iterates eventually take the form (5.9) and the equality

$$u_{k+1} - u_k = \alpha_k I_{A_*^c} s_u$$

is valid. This leads to the update

$$H_{k+1} = \begin{cases} H_k + \dfrac{vv^T}{v^T(u_{k+1} - u_k)} - \dfrac{(H_k(u_{k+1} - u_k))(H_k(u_{k+1} - u_k))^T}{(u_{k+1} - u_k)^T H_k (u_{k+1} - u_k)} \\ \qquad \text{if } v^T(u_{k+1} - u_k) > 0, \\ H_k \\ \qquad \text{else}, \end{cases} \tag{5.19}$$

where

(5.20)
$$v = T(x_k + T(x_k)(u_{k+1} - u_k))^T \nabla f(x_k + T(x_k)(u_{k+1} - u_k)) - T(x_k)^T \nabla f(x_k).$$

We state the local convergence results for the case in which exact second order information is used and for the case in which the update (5.19), (5.20) and $d(x_k) = 0$ are used in the following theorem.

THEOREM 5.5 (local convergence). *Suppose that the assumptions of Theorem* 5.4(ii) *are valid and that eventually the step size* $\alpha_k = 1$ *is accepted.*

(i) *If exact second order information is used, i.e., if*

$$H_k = T(x_k)^T \nabla^2_{xx} L(x_k, \lambda_k) T(x_k), \quad d(x_k) = T(x_k)^T \nabla^2_{xx} L(x_k, \lambda_k) r(x_k),$$

then the iterates converge locally q-quadratic.

(ii) *If the update* (5.19), (5.20) *and* $d(x_k) = 0$ *are used, then there exists* $\delta > 0$ *such that if*

$$||x_{\bar{k}} - x_*|| \leq \delta$$

and

$$||I_{A_*^c} H_{\bar{k}} I_{A_*^c} - I_{A_*^c} T(x_*)^T \nabla^2_{xx} L(x_*, \lambda_*) T(x_*) I_{A_*^c}|| \leq \delta$$

for some $\bar{k}$, *then the local convergence rate of the iterates* x_k *is 2-step superlinear, i.e.,*

$$\lim_{k \to \infty} \frac{||x_{k+1} - x_*||}{||x_{k-1} - x_*||} = 0.$$

Proof. As mentioned before, the key observation for the proof of this local convergence result is that if the active indices are identified, then the projected SQP method reduces to an SQP method in the variables $(y, I_{A_*^c} u)$. In this case the reduced gradient and the reduced Hessian for the variables $(y, I_{A_*^c} u)$ are given by (5.16).

(i) If second order information is used then the convergence proof is based on the convergence of Newton's method and is similar to the one given in [18]. However, we use a different Lagrange multiplier estimate and must use different arguments.

Suppose that the active set is identified. Then

$$I_{A_*} T(x)^T \nabla_x f(x) = I_{A_*} (\nabla_u f(x) + C_u(x)\lambda) > 0, \quad I_{A_*} u = 0. \tag{5.21}$$

We set

$$\hat{s}_u = I_{A_*^c} s_u, \ \hat{u} = I_{A_*^c} u, \ \hat{x} = (y, \hat{u}).$$

Since $\lambda = \lambda(x)$ is defined by (4.16) and since (5.21) is satisfied for all $x = x_k$, $k \geq \bar{k}$, the optimality conditions for (1.7) are given by

$$\begin{aligned} I_{A_*^c} (\nabla_u f(x) + C_u(x)\lambda) &= 0, \\ c(x) &= 0; \end{aligned} \tag{5.22}$$

cf. (4.6b), (4.6c). Recall that $T(x)^T \nabla f(x) = \nabla_u L(x,\lambda) = \nabla_u f(x) + C_u(x)\lambda$. Newton's method for the optimality conditions (5.22) leads to the linear system (we omit the arguments $\hat{x}, \lambda$)

$$\begin{aligned} I_{A_*^c} \nabla_{uy} L \, s_y + I_{A_*^c} \nabla_{u\lambda} L \lambda_y \, s_y + I_{A_*^c} \nabla_{uu} L \, \hat{s}_u + I_{A_*^c} \nabla_{u\lambda} L \lambda_u \, \hat{s}_u &= -I_{A_*^c} T^T \nabla f, \\ C_y^T s_y + C_u^T \hat{s}_u &= -c, \end{aligned} \tag{5.23}$$

where the derivatives of λ are obtained by the implicit function theorem and are given by

$$\lambda_u = -C_y^{-1}(\hat{x}) \nabla_{yu} L(\hat{x}, \lambda), \quad \lambda_y = -C_y^{-1}(\hat{x}) \nabla_{yy} L(\hat{x}, \lambda).$$

If we also use that $\nabla_{u\lambda} L(\hat{x}, \lambda) = C_u(\hat{x})$, then the Newton system can be written as

$$\begin{aligned} I_{A_*^c} \nabla_{uy} L \, s_y - I_{A_*^c} C_u C_y^{-1} \nabla_{yy} L \, s_y + I_{A_*^c} \nabla_{uu} L \, \hat{s}_u - I_{A_*^c} C_u C_y^{-1} \nabla_{yu} L \, \hat{s}_u &= -I_{A_*^c} T^T \nabla f, \\ C_y^T s_y + C_u^T \hat{s}_u &= -c. \end{aligned}$$

Now we use $s_y = -C_y(\hat{x})^{-T} C_u(\hat{x})^T \hat{s}_u - C_y(\hat{x})^{-T} c(\hat{x})$ and $\hat{s}_u = I_{A_*^c} \hat{s}_u$ to get

$$I_{A_*^c} T^T(\hat{x}) \nabla_{xx} L(\hat{x}, \lambda) T(\hat{x}) I_{A_*^c} \hat{s}_u = -I_{A_*^c} T(\hat{x})^T \nabla f(\hat{x}) - I_{A_*^c} T(\hat{x})^T \nabla_{xx} L(\hat{x}, \lambda) r(\hat{x}) \, .$$

Hence, the projected SQP methods with full second order information and the Lagrange multiplier estimate $\lambda(x) = -C_y(x)^{-1} \nabla_y f(x)$ are equivalent to Newton's method in the free variables $(y, \hat{u})$ applied to (5.22). This gives local q-quadratic convergence in $(y, \hat{u})$.

(ii) The second part of the theorem follows by applying the local convergence results for equality constrained reduced SQP methods. See [24]. □

Remark 5.6. Of course, projected SQP methods are not restricted to the use of full second order information or the use of the particular quasi-Newton update (5.19), (5.20). Similar extensions for other quasi-Newton updates (see, e.g., [7], [8], [25], [31]) can be derived and analyzed. However, this is beyond the scope of this paper.

We conclude the section with a full statement of a particular projected SQP method which is used for the computations reported in §6. In this implementation the parameter $\epsilon(x)$ is chosen to be

$$\epsilon(x_k) = \min \left\{ \bar{\epsilon}, \; ||c(x_k)|| + ||u_k - P(u_k - T(x_k)^T \nabla_x f(x_k))|| \right\}. \tag{5.24}$$

We use the quasi-Newton update (5.19), (5.20) and $d(x_k) = 0$. Obviously, for these choices the assumptions (A5) and (A6) with $M(x) = I$ are satisfied. Moreover, we use the definition (4.14) for the y-component of the iterate. The resulting algorithm is given.

ALGORITHM 5.7.

Initialization:

Given a vector $x_0 = (y_0, u_0) \in \mathbb{R}^{m+n}$, a positive definite matrix $B_0 \in \mathbb{R}^{n \times n}$, a penalty parameter $\rho > 0$, and scalars $\bar{\epsilon} > 0$, $\sigma, \beta \in (0,1)$.

Set $k = 0$.

Step k:

Compute the Lagrange multiplier estimate and the reduced gradient:

Solve $\quad C_y(x_k)\lambda = -\nabla_y f(x_k)$,

and set $T(x_k)^T\nabla_x f(x_k) = \nabla_u f(x_k) + C_u(x_k)\lambda_k$.

Compute active set:

Set $\epsilon_k = \min\{\bar{\epsilon},\ ||c(x_k)|| + ||u_k - P(u_k - T(x_k)^T\nabla_x f(x_k))||\}$

and $A^+_{\epsilon,k} = A^+_\epsilon(x_k) = \{i \mid u_i \in [0,\epsilon_k],\quad (T(x_k)^T\nabla_x f(x_k))_i > 0\}$.

Set $(u_k)_i = 0$ for all $i \in A^+_{\epsilon,k}$.

Set $B_{\epsilon,k} = B_\epsilon(x_k) = \left\{ i \notin A^+_\epsilon(x_k) \;\middle|\; (u_k)_i = 0 \text{ and } (s_u)_i \geq 0, \text{ or } (u_k)_i > 0 \right\}$.

Compute

$$D_k = I_{A^+_{\epsilon,k}}\operatorname{diag}(H_k)I_{A^+_{\epsilon,k}} + I_{A^{+c}_{\epsilon,k}}H_k I_{A^{+c}_{\epsilon,k}},$$
$$s_u = -D_k^{-1}\big(T(x_k)^T\nabla_x f(x_k)\big).$$

Determine the step size $\alpha_k = \beta^{m_k}$, where m_k is the smallest integer satisfying

$$\Phi(y_k,u_k,\rho) - \Phi(y_k(\beta^m),u_k(\beta^m),\rho)$$
$$\geq \sigma\Big(\rho\beta^m||c(x_k)||^2 - \beta^m\big(I_{A^{+c}_{\epsilon,k}}s_u\big)^T\big(T(x_k)^T\nabla f(x_k)\big)$$
$$+\big(u_k - P(u_k+\beta^m s_u)\big)^T I_{A^+_{\epsilon,k}}\big(T(x_k)^T\nabla f(x_k)\big)\Big),$$

with

$$u_k(\alpha) = P(u_k + \alpha s_u)\ ,$$

$$y_k(\alpha) = \begin{cases} y_k - \alpha C_y(x_k)^{-T}\big(c(x_k) + C_u(x_k)^T I_{B_{\epsilon,k}}s_u\big) \\ \qquad \text{if } \alpha \in (0,\tilde{\alpha}_k], \\ y_k - \frac{\alpha-\tilde{\alpha}_k}{1-\tilde{\alpha}_k}C_y(x_k)^{-T}\Big(c(x_k) + C_u(x_k)^T\big(P(u_k+s_u)-u_k\big)\Big) \\ \quad -\big(1-\frac{\alpha-\tilde{\alpha}_k}{1-\tilde{\alpha}_k}\big)\tilde{\alpha}_k C_y(x_k)^{-T}\Big(c(x_k) + C_u(x_k)^T I_{B_{\epsilon,k}}s_u\Big) \\ \qquad \text{if } \alpha \in (\tilde{\alpha}_k,1], \end{cases}$$

$$\tilde{\alpha}_k = \max\{\alpha \in [0,1] \mid u_k + \alpha I_{B_{\epsilon,k}}s_u \geq 0\}.$$

Update H_k:

Compute

$$v = T(x_k + T(x_k)(u_{k+1}-u_k))^T\nabla f(x_k + T(x_k)(u_{k+1}-u_k))$$
$$-T(x_k)^T\nabla f(x_k).$$

If $v^T(u_{k+1}-u_k) > 0$, set

$$H_{k+1} = H_k + \frac{vv^T}{v^T(u_{k+1}-u_k)} - \frac{(H_k(u_{k+1}-u_k))(H_k(u_{k+1}-u_k))^T}{(u_{k+1}-u_k)^T H_k(u_{k+1}-u_k)}.$$

Else set $H_{k+1} = H_k$.

For the computation of v we have to perform the following steps:

Solve $C_y(x_k)^T\eta = -C_u(x_k)^T(u_{k+1}-u_k)$.

Solve $C_y(y_k+\eta,u_{k+1})\,\tilde{\lambda} = -\nabla_y f(y_k+\eta,u_{k+1})$.

Compute $T(y_k+\eta,u_{k+1})^T\nabla f(y_k+\eta,u_{k+1})$
$= \nabla_u f(y_k+\eta,u_{k+1}) + C_u(y_k+\eta,u_{k+1})\,\tilde{\lambda}$,

$v = T(y_k+\eta,u_{k+1})^T\nabla f(y_k+\eta,u_{k+1}) - T(y_k,u_k)^T\nabla f(y_k,u_k)$.

6. Application to a discretized control problem.

6.1. The discretized control problem. In this section we discuss the application of projected SQP methods to the discretization of the optimal control problem (1.2)–(1.4).

We assume that the specific heat capacity and the heat conduction satisfy $\tau, \kappa \in C^1(\mathbb{R})$ and

$$0 < \tau_1 \le \tau(t) \le \tau_2, \quad 0 < \kappa_1 \le \kappa(t) \le \kappa_2, \quad |\tau'(t)| \le \tau_3, \quad |\kappa'(t)| \le \kappa_3 \quad \forall\, t \in \mathbb{R}. \tag{6.1}$$

It is shown in [6] that if τ and κ satisfy (6.1) and if $u \in L^2(0,T)$, $y_0 \in H^1(\Omega)$, $q \in L^2(0,T;L^2(\Omega))$, then the state equation has a unique solution $y \in Y$, where the state space is given by $Y = \{y \mid y \in L^\infty(0,T;H^1(\Omega)),\ y' \in L^2(0,T;H^1(\Omega)')\}$. Moreover, it is proven in [6] that the unconstrained control problem (1.2)–(1.3) has a solution u_*, provided the penalty parameter γ is positive.

We first describe the discretization of the control problem using finite elements. For this control problem this discretization scheme was introduced and analyzed in [6]. We briefly outline the discretization scheme. For more detailed presentations we refer to [6], [27].

The variational form of the boundary value problem (1.2) is given by

$$\begin{aligned} &\langle \tau(y(\cdot,t))\tfrac{\partial y}{\partial t}(\cdot,t), v\rangle + \langle \kappa(y(\cdot,t)\partial_x y(\cdot,t)), \partial_x v\rangle + g\, y(0,t)v(0) \\ &= \langle q(\cdot,t), v\rangle + g\, u(t)v(0) \qquad\qquad \forall v \in H^1(\Omega), \end{aligned} \tag{6.2}$$

with initial conditions

$$y(\cdot,0) = y_0(\cdot).$$

Here $\langle \cdot,\cdot \rangle$ denotes the inner product in $L^2(\Omega)$.

If we introduce the functions

$$\Gamma(t) = \int_0^t \tau(s)\, ds, \quad \beta(t) = \int_0^t \kappa(s)\, ds,$$

then (6.2) can be written as

$$\begin{aligned} &\langle \tfrac{\partial}{\partial t}\Gamma(y(\cdot,t)), v\rangle + \langle \partial_x \beta(y(\cdot,t)), \partial_x v\rangle + g y(0,t)v(0) = \langle q(\cdot,t), v\rangle + g u(t)v(0), \\ &\qquad\qquad y(\cdot,0) = y_0(\cdot). \end{aligned} \tag{6.3}$$

If the variational form (6.3) is considered as a problem in the variable $\phi(\cdot,\cdot) = \Gamma(y(\cdot,\cdot))$, then the problem is linear with respect to differentiation in time. This is the main observation needed to obtain a discretization of (6.2). In the following we list the basic ingredients needed for the implementation. A more detailed presentation and error estimates for the scheme are given in [6].

To discretize the control problem the space interval is divided into N_x subintervals of equidistant length $h = 1/N_x$ with grid points $x_i = (i-1)h, i = 1, \ldots, N_x + 1$. The time discretization is performed by partitioning the interval $[0,T]$ into N_t equidistant subintervals of length $\Delta t = T/N_t$ with grid points $t^j = j\Delta t, j = 1, \ldots, N_t$. We define $V_h = \operatorname{span}\{\varphi_1, \ldots, \varphi_{N_x+1}\}$, where φ_i denotes the hat function satisfying $\varphi_i(x_j) = \delta_{ij}$, $i,j = 1, \ldots, N_x + 1$.

The state space Y is approximated by the subspace $Y_{h,\Delta t}$ of functions from $(0,T]$ into V_h which are constant on each interval $(t^{j-1}, t^j]$:

$$Y_{h,\Delta t} = \left\{ y(x,t) = \sum_{j=1}^{N_t} y^j \chi_j(t) \;\middle|\; y^j \in V_h \right\},$$

where χ_j denotes the characteristic function on $(t^{j-1}, t^j]$. The control space $U = L^2(0,T)$ is approximated by the subset $U_{\Delta t}$ of piecewise constant functions on $[0,T]$. Note that $y^j(x) = \sum_{i=1}^{N_x+1} y_i^j \varphi_i(x)$. We introduce the notation $\vec{y}^j = (y_1^j, \ldots, y_{N_x+1}^j)$.

The inhomogeneous term q and the initial temperature y_0 are replaced by the elements $q^j \in V_h$ and $y^0 \in V_h$, respectively, which satisfy

$$\begin{aligned} \langle q^j, v\rangle &= \frac{1}{\Delta t}\int_{t^{j-1}}^{t^j} \langle q(\cdot,t), v\rangle \, dt \quad \forall\, v \in V_h, \quad j = 1, \ldots, N_t\,, \\ \langle y^0, v\rangle &= \langle y_0, v\rangle \qquad\qquad\qquad\qquad \forall v \in V_h. \end{aligned}$$

We set $\vec{q}^j = (\langle q^j, \varphi_1\rangle, \ldots, \langle q^j, \varphi_{N_x+1}\rangle)^T$, $\vec{\Gamma}(\vec{y}^j) = (\Gamma(y_1^j), \ldots, \Gamma(y_{N_x+1}^j))^T$, and $\vec{\beta}(\vec{y}^j) = (\beta(y_1^j), \ldots, \beta(y_{N_x+1}^j))^T$.

If the control is given by $u = \sum_{j=1}^{N_t} u^j \chi_j(t)$, then (6.3) is discretized using

$$\Delta t^{-1} M\big(\vec{\Gamma}(\vec{y}^j) - \vec{\Gamma}(\vec{y}^{j-1})\big) + L\vec{\beta}(\vec{y}^j) + g e_1 e_1^T \vec{y}^j = \vec{q}^j + g u^j e_1, \tag{6.4}$$

where e_1 is the first unit vector in $\mathbb{R}^{N_x+1}$, and $M = (\langle \varphi_i, \varphi_j\rangle)_{i,j}$, $L = (\langle \partial_x \varphi_i, \partial_x \varphi_j\rangle)_{i,j} \in \mathbb{R}^{(N_x+1)\times(N_x+1)}$ denote the mass matrix and stiffness matrix, respectively.

The nonlinear system (6.4) can be written in the form

$$c(\vec{y}, \vec{u}) = 0, \tag{6.5}$$

where $\vec{y} = (y_1^1, \ldots, y_{N_x+1}^{N_t}) \in \mathbb{R}^{N_t(N_x+1)}$ and $\vec{u} = (u^1, \ldots, u^{N_t}) \in \mathbb{R}^{N_t}$.

For the approximation of the objective function in (1.3) we replace $y(1,\cdot)$, u, and y_d by their discrete expressions

$$y(1,\cdot) \approx \sum_{j=1}^{N_t} y_{N_x+1}^j \chi_j, \quad u \approx \sum_{j=1}^{N_t} u^j \chi_j, \quad \text{and } \; y_d \approx \sum_{j=1}^{N_t} y_d^j \chi_j,$$

where the data y_d^j are computed by taking the mean of y_d on the subinterval $(t^{j-1}, t^j]$:

$$y_d^j = \frac{1}{\Delta t}\int_{t^{j-1}}^{t^j} y_d(t)\, dt, \quad j = 1, \ldots, N_t. \tag{6.6}$$

We have the following objective in the discrete control problem:

$$f(\vec{y}, \vec{u}) = \frac{\Delta t}{2}\sum_{j=1}^{N_t} (y_{N_x+1}^j - y_d^j)^2 + \gamma (u^j)^2. \tag{6.7}$$

Using (6.5) and (6.7), the discretization of the control problem (1.2)–(1.4) is given by

$$\begin{array}{ll} \min & f(\vec{y}, \vec{u}) \\ \text{s.t.} & c(\vec{y}, \vec{u}) = 0, \\ & \vec{u}_{\text{low}} \le \vec{u} \le \vec{u}_{\text{upp}}. \end{array} \tag{6.8}$$

This problem is of the form (1.1) with $n = N_t$ and $m = N_t(N_x+1)$.

From (6.4), (6.5) one can see that the derivative of c with respect to y is a lower block bidiagonal matrix. The diagonal blocks are tridiagonal matrices. It is shown in [27, Lem. 3.1] that these diagonal blocks and, therefore, C_y are nonsingular provided

$$\frac{\Delta t}{h^2} < \frac{1}{6}\left(\frac{\tau_2}{\kappa_1} - \frac{\tau_1}{\kappa_2}\right)^{-1}.$$

6.2. Numerical results. We applied the projected SQP Algorithm 5.7 to the discrete optimal control problem discussed in §6.1. In all numerical tests we used the penalty parameter $\rho = 10^3$ and the parameters $\beta = 0.5$ and $\sigma = 10^{-4}$ in the Armijo line search. All computations were done on a Sun Sparcstation 10/30 in double precision FORTRAN. As we have seen in the previous section, the Jacobian $C_u(\vec{y}, \vec{u})$ is a block bidiagonal matrix with tridiagonal block matrices. Therefore, the adjoint equation and the linearized state equation arising in the computation of the Lagrange multiplier estimate $\lambda(x)$ and the iterate $y_k(\alpha)$ can be solved efficiently. In our implementation we use the LINPACK subroutine DGTSL to solve the subsystems of the form $G(\vec{y}^i)z = b$, $G(\vec{y}^i)^T z = b$.

In the tables shown next we use weighted norms instead of the ℓ_2-norm. More precisely, we use the following discretizations of the $L^2(0,T)$-norm and the $L^2(0,T;H^1)$-norm:

$$||\vec{u}||_U^2 = \Delta t \sum_{j=1}^{N_t} u_j^2, \quad ||\vec{y}||_Y^2 = \Delta t \sum_{j=1}^{N_t} (\vec{y}^j)^T (M + L)\, \vec{y}^j,$$

where $\vec{y}^T = ((\vec{y}^1)^T, \ldots, (\vec{y}^{N_t})^T)$, $\vec{y}^j \in I\!R^{N_x+1}$.

In our numerical example we use the functions

$$\tau(y) = q_1 + q_2 y, \quad y \in I\!R, \quad \kappa(y) = r_1 + r_2 y, \quad y \in I\!R,$$

with parameters $r_1 = q_1 = 4$, $r_2 = -1$, $q_2 = 1$. The desired temperature, the initial temperature, and the right-hand side are given by

$$\begin{aligned} y_d(t) &= 2 - e^{\rho t}, \\ y_0(x) &= 2 + \cos \pi x, \\ q(x,t) &= [\rho(q_1 + 2q_2) + \pi^2(r_1 + 2r_2)]e^{\rho t} \cos \pi x \\ &\quad -r_2 \pi^2 e^{2\rho t} + (2r_2\pi^2 + \rho q_2) e^{2\rho t} \cos^2 \pi x, \end{aligned}$$

with $\rho = -1$. The final temperature is chosen to be $T = 0.5$ and the scalar g entering the boundary condition in (1.2) is set to be one.

The functions in this example are equal to those used in [27, Ex. 4.1]. However, we use a positive regularization parameter γ. Therefore, the functions y_*, u_* are no longer solutions of the optimal control problem. The optimal control problem (1.2)–(1.4) is a nonlinear inverse heat conduction problem (IHCP) and it is well known that these problems are ill posed. Choosing $\gamma > 0$ sufficiently large or introducing bound constraints has a regularizing effect. The reason to choose a positive regularization parameter is to obtain a regularizing effect from the penalty term $\gamma/2 \int_0^T u^2(t)dt$ but not from the bound constraints to allow a reasonable comparison between the case with and without bound constraints.

The computations reported in Tables 6.1 and 6.2 are done using the regularization parameter: $\gamma = 10^{-3}$. The discretization is given by $N_t = 100, N_x = 20$ which yields a problem with $n + m = 2100$ variables, 2000 equality constraints, and 100 bound constraints. The starting values $u_0 = 0$, $y_0 = 0$, and $B_0 = \gamma I$ are used. The computed controls for the constrained and unconstrained problem are shown in Figure 6.1.

It is interesting to observe that in the initial iteration all indices are identified as active. This is because we set $\bar{\epsilon} = 1$, which yields $\epsilon(x_0) = 1$; cf. (5.24). In the first iteration all components of the reduced gradients are negative. Since $|(u_0)_i - 0.1| \leq \epsilon(x_0) = 1$ for all i this implies that all indices belong to the set $A^+_{\epsilon,0}$. However, even

Table 6.1
$-\infty \leq u \leq \infty$.

k	f_k	$\|\|c_k\|\|_Y$	$\|\|T_k^T \nabla f_k\|\|_U$	α	$\|\|s_k\|\|_U$	$\|\|s_k/s_{k-1}\|\|_U$	$\|\|s_k/s_{k-2}\|\|_U$
0	$0.371E+0$	$.295E+2$	$.341E-1$	$.25E+0$	$.853E+1$		
1	$0.757E-1$	$.269E+2$	$.811E-2$	$.10E+1$	$.811E+1$	$.950E+0$	
2	$0.214E-2$	$.641E+1$	$.631E-3$	$.10E+1$	$.534E+0$	$.659E-1$	$.626E-1$
3	$0.392E-4$	$.571E+1$	$.456E-4$	$.10E+1$	$.384E-1$	$.719E-1$	$.474E-2$
4	$0.400E-4$	$.153E+0$	$.452E-5$	$.10E+1$	$.442E-2$	$.114E+0$	$.826E-2$
5	$0.401E-4$	$.891E-4$	$.944E-7$	$.10E+1$	$.899E-4$	$.203E-1$	$.233E-2$
6	$0.401E-4$	$.683E-8$	$.108E-8$				

Table 6.2
$-\infty \leq u \leq 0.1$.

k	$\|A_{\epsilon,k}^+\|$	f_k	$\|\|c_k\|\|_Y$	$\|\|T_k^T \nabla f_k\|\|_U$	α	$\|\|s_k\|\|_U$	$\|\|s_k/s_{k-1}\|\|_U$	$\|\|s_k/s_{k-2}\|\|_U$
0	100	$0.371E+0$	$.295E+2$	$.341E-1$	$.10E+1$	$.172E+0$		
1	0	$0.118E+0$	$.154E+2$	$.115E-2$	$.50E+0$	$.576E+0$	$.334E+1$	
2	0	$0.289E-1$	$.795E+1$	$.332E-3$	$.10E+1$	$.165E+1$	$.286E+1$	$.958E+1$
3	0	$0.274E-2$	$.369E+1$	$.245E-2$	$.10E+1$	$.222E+1$	$.134E+1$	$.386E+1$
4	18	$0.179E-4$	$.169E+1$	$.314E-4$	$.10E+1$	$.316E-1$	$.141E-1$	$.191E-1$
5	36	$0.402E-4$	$.437E-1$	$.179E-4$	$.10E+1$	$.106E-1$	$.337E+0$	$.479E-2$
6	33	$0.403E-4$	$.947E-4$	$.102E-6$	$.10E+1$	$.102E-3$	$.955E-2$	$.322E-2$
7	33	$0.403E-4$	$.446E-8$	$.677E-9$				

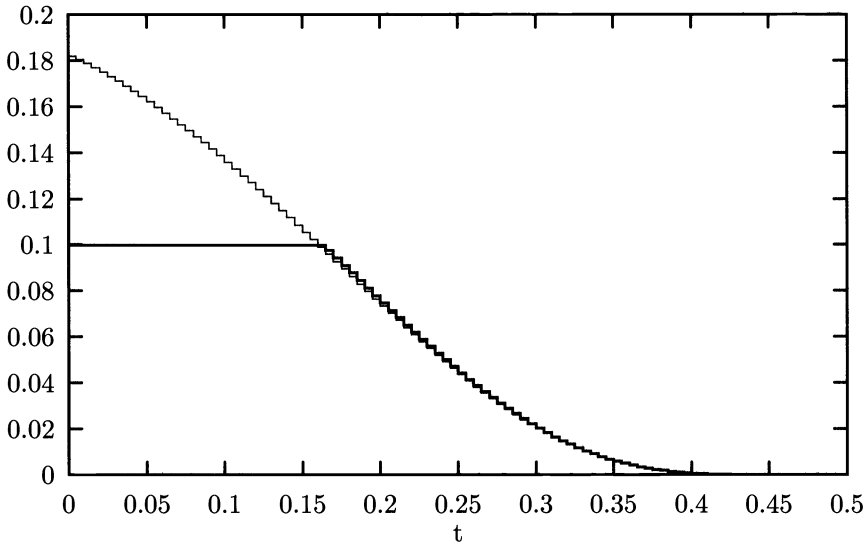


FIG. 6.1. *Unconstrained and constrained control.*

though this is obviously not a correct estimate for the active set, the algorithm is able to recover.

7. Conclusions. We have introduced and analyzed a method for the solution of optimization problems with nonlinear equality constraints and bound constraints, which arise, e.g., in optimal control problems with control constraints.

The method is based on a combination and extension of reduced SQP methods

and projected Newton methods. This approach has several advantages. The structure of optimal control problems can be used. By using the simple projection onto the set $\mathcal{U} = \{a \leq u \leq b\}$ steps can be computed without iterative schemes as an inner iteration for solving constrained quadratic programs. The iterates are feasible with respect to $\mathcal{U}$. The size of the reduced problem is independent of the number of active constraints. This allows the use of quasi-Newton updates for the reduced Hessian in a straightforward way. The active indices are identified after finitely many iterations. Therefore, local convergence results for SQP methods for equality constrained problems can be applied.

To obtain a global convergent algorithm we have extended Fletcher's exact penalty function. In the global convergence analysis the penalty parameter is assumed to be sufficiently large and fixed. The use of adaptive penalty parameter choices is the subject of current research.

The algorithm has been applied to an optimal control problem. More numerical tests are done and a numerical comparison with other methods is under investigation.

Acknowledgments. The author is indebted to J. E. Dennis, Rice University, for suggesting the choice (4.14) for $y_k(\alpha)$. The author also would like to thank Luis N. Vicente, Rice University, and the referees for their comments which led to improvements in the paper.

REFERENCES

[1] W. Alt, *The Lagrange–Newton method for infinite dimensional optimization problems*, Numer. Funct. Anal. Optim., 11 (1990), pp. 201–224.

[2] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[3] J. T. Betts and P. D. Frank, *A sparse nonlinear optimization algorithm*, Journal Optim. Theory Appl., 82 (1994), pp. 519–541.

[4] P. T. Boggs, J. W. Tolle, and A. J. Kearsley, *A truncated SQP algorithm for large scale nonlinear programming problems*, in Proc. Sixth Conference on Numerical Analysis and Optimization, S. Gomez and J.-P. Hennart, eds., Boston, 1994, Kluwer Academic Publishers, Norwell, MA.

[5] J. Bonnans and C. Pola, *A trust region interior point algorithm for linearly constrained optimization*, Tech. rep. 1948, INRIA, Domaine de Voluceau, 78153 Rocquencourt, France, 1993.

[6] J. Burger and M. Pogu, *Functional and numerical solution of a control problem originating from heat transfer*, J. Optim. Theory Appl., 68 (1991), pp. 49–73.

[7] R. H. Byrd, *On the convergence of constrained optimization methods with accurate Hessian information on a subspace*, SIAM J. Numer. Anal., 27 (1990), pp. 141–153.

[8] R. H. Byrd and J. Nocedal, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1991), pp. 285–323.

[9] T. F. Coleman and J. Liu, *An interior Newton method for quadratic programming*, CTC93TR153, Center for Theory and Simulation in Science and Engineering, Cornell University, Ithaca, NY, 1993.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[11] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, *On alternative problem formulations for multidisciplinary design optimization*, SIAM J. Optim., 4 (1994), pp. 754–776.

[12] R. S. Dembo and U. Tulowitzki, *Sequential truncated quadratic programming*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1985, pp. 83–101.

[13] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[14] M. EL-ALEM, *A global convergence theory for the Celis–Dennis–Tapia trust region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[15] A. S. EL-BAKRY, R. A. TAPIA, T. TSUCHIYA, AND Y. ZHANG, *On the formulation and theory of the primal–dual Newton interior-point method for nonlinear programming*, TR92–40, Dept. of Computational & Applied Mathematics, Rice University, Houston, TX, 1992 (revised May 1993).

[16] A. V. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, 1968.

[17] R. FLETCHER, *A class of methods for nonlinear programming with termination and convergence properties*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1970.

[18] ———, *Practical Methods of Optimization*, 2nd ed., John Wiley, New York, 1987.

[19] A. FRIEDLANDER, J. M. MARTÍNEZ, AND S. A. SANDROS, *On the resolution of linearly constrained convex minimization problems*, SIAM J. Optim., 4 (1994), pp. 331–339.

[20] J. C. GILBERT, *On the local and global convergence of a reduced quasi-Newton method*, Optimization, 20 (1989), pp. 421–450.

[21] P. E. GILL, W. MURRAY, D. B. PONCELEÓN, AND M. A. SAUNDERS, *Solving reduced KKT systems in barrier methods for linear and quadratic programming*, Tech. report SOL 91–7, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[22] P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *Some theoretical properties of an augmented Lagrangian merit function*, in Advances in Optimization and Parallel Computing, P. M. Pardalos, ed., Elsevier Science Publishers B.V., 1992, pp. 101–128.

[23] C. T. KELLEY AND E. W. SACHS, *Solution of optimal control problems by a pointwise projected Newton method*, SIAM J. Control Optim., 33 (1995), pp. 1731–1757.

[24] F.-S. KUPFER, *An infinite-dimensional convergence theory for reduced SQP methods in Hilbert space*, SIAM J. Optim., 6 (1996), pp. 126–163.

[25] ———, *Reduced Successive Quadratic Programming in Hilbert Space with Applications to Optimal Control*, Ph.D. thesis, Universität Trier, Fb–IV, Mathematik, D–54286 Trier, Germany, 1992.

[26] F.-S. KUPFER AND E. W. SACHS, *A prospective look at SQP methods for semilinear parabolic control problems*, in Optimal Control of Partial Differential Equations, Irsee 1990, K.-H. Hoffmann and W. Krabs, eds., vol. 149, Springer Lect. Notes in Control and Information Sciences, 1991, pp. 143–157.

[27] ———, *Numerical solution of a nonlinear parabolic control problem by a reduced SQP method*, Comput. Optim. Appl., 1 (1992), pp. 113–135.

[28] R. D. C. MONTEIRO, *A globally convergent primal–dual interior point algorithm for convex programming*, Math. Programming, 64 (1994), pp. 123–147.

[29] W. MURRAY AND F. J. PRIETO, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.

[30] S. G. NASH AND A. SOFER, *Algorithm 711: BTN: Software for parallel unconstrained optimization*, ACM Trans. Math. Software, 18 (1992), pp. 414–448.

[31] J. NOCEDAL AND M. L. OVERTON, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[32] M. J. D. POWELL AND Y. YUAN, *A trust region algorithm for equality constrained optimization*, Math. Programming, 24 (1991), pp. 189–211.

[33] K. SCHITTKOWSKI, *The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search. Part 1: Convergence analysis*, Numer. Math., 38 (1981), pp. 83–114.

[34] M. H. WRIGHT, *Interior point methods for constrained optimization*, in Acta Numerica 1992, A. Iserles, ed., Cambridge University Press, Cambridge, 1992, pp. 341–407.

[35] S. J. WRIGHT, *An interior-point algorithm for linearly constrained optimization*, SIAM J. Optim., 2 (1992), pp. 450–473.

[36] Y. YE AND E. TSE, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming, 44 (1989), pp. 157–179.

[37] Y. ZHANG, R. A. TAPIA, AND F. POTRA, *On the superlinear convergence of interior point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 418–445, May 1996
© 1996 Society for Industrial and Applied Mathematics
009

AN INTERIOR TRUST REGION APPROACH FOR NONLINEAR MINIMIZATION SUBJECT TO BOUNDS*

THOMAS F. COLEMAN† AND YUYING LI†

Abstract. We propose a new trust region approach for minimizing a nonlinear function subject to simple bounds. Unlike most existing methods, our proposed method does not require that a quadratic programming subproblem, with inequality constraints, be solved in each iteration. Instead, a solution to a trust region subproblem is defined by minimizing a quadratic function subject only to an ellipsoidal constraint. The iterates generated are strictly feasible. Our proposed method reduces to a standard trust region approach for the unconstrained problem when there are no upper or lower bounds on the variables. Global and local quadratic convergence is established. Preliminary numerical experiments are reported indicating the practical viability of this approach.

Key words. trust region method, interior Newton method, interior-point method

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction.

We consider the problem of computing a local minimizer of a smooth nonlinear function subject to bounds on the variables:

$$\min_{x\in\Re^n} f(x), \quad l \le x \le u, \tag{1.1}$$

where $l \in \{\Re \cup \{-\infty\}\}^n$, $u \in \{\Re \cup \{\infty\}\}^n$, $l < u$, and $f : \Re^n \to \Re^1$. We denote the feasible set $\mathcal{F} \stackrel{\text{def}}{=} \{x : \ l \le x \le u\}$ and the strict interior $\text{int}(\mathcal{F}) \stackrel{\text{def}}{=} \{x : \ l < x < u\}$.

We propose a strictly feasible trust region approach for problem (1.1). Global convergence to a second-order point is established under reasonable assumptions, and a local quadratic convergence rate is also obtained.

Minimization problems with upper and/or lower bounds on some of the variables form an important and common class of problems. There are many algorithms for this type of optimization problem (e.g., [1, 4, 5, 7–9, 11, 13–15, 19–21, 25]), some of which are restricted to quadratic (in some cases convex quadratic) objective functions and some of which are more general. Almost all of the existing methods for problem (1.1) are "active set" methods.

Trust region methods form a respected class of algorithms for solving unconstrained minimization problems. Their high regard is partially due to their strong convergence properties, partially due to their naturalness, and partially due to the recent development of reliable, efficient software. The idea behind a trust region method for $\min_{x\in\Re^n} f(x)$ is very simple. The increment $s_k = x_{k+1} - x_k$ is an approximate solution to a quadratic subproblem with a bound on the step:

$$\min_{s\in\Re^n} \left\{ \psi_k(s) \stackrel{\text{def}}{=} g_k^T s + \frac{1}{2} s^T B_k s \ : \ \|\bar{D}_k s\| \le \Delta_k \right\}. \tag{1.2}$$

* Received by the editors May 12, 1993; accepted for publication (in revised form) October 21, 1994. This research was partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U. S. Department of Energy under grant DE-FG02-86ER25013.A000, in part by the National Science Foundation, Air Force Office of Scientific Research, and Office of Naval Research through grant DMS-8920550, and by the Advanced Computing Research Institute, a unit of the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

† Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14850 (coleman@cs.cornell.edu and yuying@cs.cornell.edu).

Here $g_k \stackrel{\text{def}}{=} \nabla f(x_k)$, B_k is a symmetric approximation to the Hessian matrix $\nabla^2 f(x_k)$, $\bar{D}_k$ is a scaling matrix, and Δ_k is a positive scalar representing the trust region size. Throughout our presentation, $\|\cdot\|$ denotes the 2-norm.

A general scheme for the unconstrained minimization of $f(x)$, based on subproblem (1.2), is described in Figure 1.1. An iteration with $\rho_k^f > \mu$ is said to be *successful*. Otherwise, the iteration is *unsuccessful*. The aim of trust region size updating is to force $\rho_k^f > \mu$ and hence ensure sufficient reduction of the objective function.

Algorithm 0. Unconstrained trust region method
Let $0 < \mu < \eta < 1$.
For $k = 0, 1, \ldots$
1. Compute $f(x_k)$ and the model ψ_k.
2. Define an approximate solution s_k to subproblem (1.2).
3. Compute $\rho_k^f = (f(x_k + s_k) - f(x_k))/\psi_k(s_k)$.
4. If $\rho_k^f > \mu$ then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.
5. Update the scaling matrix $\bar{D}_k$ and Δ_k.

Updating trust region size
Let $0 < \gamma_1 < 1 < \gamma_2$.
1. If $\rho_k^f \leq \mu$ then set $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$.
2. If $\rho_k^f \in (\mu, \eta)$ then set $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$.
3. If $\rho_k^f \geq \eta$ then set $\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k]$.

FIG. 1.1. *Trust region method for unconstrained minimization.*

Computing a solution to the trust region problem (1.2) in a reliable and efficient way is a nontrivial task. There are several papers on this topic, e.g., [2, 3, 7, 8, 12, 18, 22–24].

Trust region methods have also been developed for the solution of linearly constrained optimization problems (e.g., [9] and [10]). A quadratic trust region subproblem with linear inequalities is usually approximately solved to obtain an improved point. An iterative procedure must be used to solve the subproblem. For example, Fletcher [10] proposes an algorithm for the linearly constrained optimization problem

$$\min\{f(x) \ : \ E^T x \leq d\}$$

in which the subproblem is of the form

$$\min_{s \in \Re^n} \left\{ g_k^T s + \frac{1}{2} s^T B_k s : E^T(x_k + s) \leq d, \|\bar{D}_k s\|_\infty \leq \Delta_k \right\}. \tag{1.3}$$

As pointed out in [17], convergence theory for trust region methods based on quadratic programming subproblems such as (1.3) usually requires that the computed trial step be a global solution to the subproblem. However, the subproblem is typically solved by methods which guarantee local optimality at best. Therefore, there is a mismatch between theory and practice for trust region methods based on quadratic programming subproblems (with linear inequalities).

In this paper, we propose a trust region approach for (1.1) that does not require the solution of a general quadratic programming subproblem at each iteration. Our proposal is related to the line search based reflection methods proposed by Coleman

and Li [6]—scaling strategies and the requirement of strict feasibility are common to both approaches. The primary difference is that in [6] a line search based method is proposed, along with a "reflection" strategy to guarantee sufficient descent, whereas here we propose a pure trust region method.

The proposed approach is developed by forming a quadratic model with an appropriate quadratic function and scaling matrix: there is no need to handle the constraints explicitly. It is then possible to obtain an approximate trust region solution which can guarantee second-order convergence by solving a trust region subproblem with a simple 2-norm constraint and then satisfying the strict feasibility requirement by further restricting the step, if necessary. Our proposed approach reduces to a standard trust region algorithm for unconstrained minimization when $l = -\infty$ and $u = +\infty$. Moreover, all the convergence proofs essentially reduce to established proofs for the unconstrained trust region approach when $l = -\infty$ and $u = +\infty$.

We motivate the method in §2 and establish convergence in §3. In §4, preliminary numerical results for small dense problems are presented. A more comprehensive computational investigation of the method, particularly for large problems, will be presented in a subsequent paper.

As a general rule for notation, unless indicated otherwise, the subscript i or j denotes a component of a vector. The subscript k denotes an index for an (infinite) sequence. For any function F, we use the notation F_k to denote $F(x_k)$ and, if F: $\Re^n \to R^n$, F_{ki} to denote $F_i(x_k)$. When clear from the context, we omit the argument, that is, we use the notation F for $F(x)$ or F_i for $F_i(x)$, for instance.

2. Trust region method for bound-constrained problems. In this section, we propose a trust region method for bound-constrained problems. Our method involves choosing a scaling matrix D_k and a quadratic model $\psi_k(s)$ (we reserve $\bar{D}_k$ to denote the "classical" scaling as in (1.2)). We motivate our choice of scaling matrix by examining the optimality conditions for (1.1).

Let $g(x) \stackrel{\text{def}}{=} \nabla f(x)$. We first define a vector function $v(x) : \Re^n \to \Re^n$ as follows.

DEFINITION 2.1. *The vector $v(x) \in \Re^n$ is defined as follows: for each component $1 \le i \le n$,*

(i) *if $g_i < 0$ and $u_i < \infty$ then $v_i \stackrel{\text{def}}{=} x_i - u_i$;*

(ii) *if $g_i \ge 0$ and $l_i > -\infty$ then $v_i \stackrel{\text{def}}{=} x_i - l_i$;*

(iii) *if $g_i < 0$ and $u_i = \infty$ then $v_i \stackrel{\text{def}}{=} -1$;*

(iv) *if $g_i \ge 0$ and $l_i = -\infty$ then $v_i \stackrel{\text{def}}{=} 1$.*

Following Matlab notation, for any $s \in \Re^n$, diag(s) denotes an n-by-n diagonal matrix with the vector s defining the diagonal entries in their natural order. Moreover, for any nonsingular matrix $A \in \Re^{n\times n}$ and any $l > 0$, A^{-l} denotes the inverse of A^l, where A^l is the lth power of A. Using this notation, we define

$$D(x) \stackrel{\text{def}}{=} \text{diag}(|v(x)|^{-\frac{1}{2}}), \tag{2.1}$$

i.e., D^{-2} is a diagonal matrix with the ith diagonal component equal to $|v_i|$.

Optimality conditions for problem (1.1) are well established. Assuming feasibility and $g_* \stackrel{\text{def}}{=} g(x_*)$, *first-order necessary conditions* for x_* to be a local minimizer are

$$\text{first-order:} \begin{cases} g_{*i} = 0 & \text{if } l_i < x_{*i} < u_i, \\ g_{*i} \le 0 & \text{if } x_{*i} = u_i, \\ g_{*i} \ge 0 & \text{if } x_{*i} = l_i. \end{cases} \tag{2.2}$$

Equivalently, $D(x_*)^{-2}g_* = 0$. Second-order conditions involve the Hessian matrix of f. Let $Free_*$ denote the set of indices corresponding to "free" variables at point x_*:

$$Free_* = \{i : l_i < x_{*i} < u_i\}.$$

Second-order necessary conditions can be written:[1] if a feasible point x_* is a local minimizer of (1.1) then $D(x_*)^{-2}g_* = 0$ and $H_*^{Free_*} \geq 0$, where $H_*^{Free_*}$ is the submatrix of $H_* \stackrel{\text{def}}{=} \nabla^2 f(x_*)$ corresponding to the index set $Free_*$.

These conditions are necessary but not sufficient. Sufficiency conditions that are achievable in practice often require a nondegeneracy assumption. This is the case here.

DEFINITION 2.2. *A point $x \in \mathcal{F}$ is* nondegenerate *if, for each index i,*

$$g_i = 0 \implies l_i < x_i < u_i. \tag{2.3}$$

A problem (1.1) *is nondegenerate if* (2.3) *holds for every $x \in \mathcal{F}$.*

Using this definition we can state *second-order sufficiency conditions*: if a nondegenerate feasible point x_* satisfies (2.2) and $H_*^{Free_*} > 0$, then x_* is a local minimizer of (1.1).

Similar to the view expressed in [7], we consider the following diagonal system of nonlinear equations:

$$D(x)^{-2}g(x) = 0. \tag{2.4}$$

It is easy to see that system (2.4) is an equivalent statement of the first-order necessary conditions. System (2.4) is continuous but not everywhere differentiable. Nondifferentiability occurs when $v_i = 0$; we avoid such points by restricting $x_k \in \text{int}(\mathcal{F})$. Discontinuity of v_i may also occur when $g_i = 0$; however, $D(x)^{-2}g(x)$ is continuous at such points. Moreover, Coleman and Li [7] show that it is possible to generate a second-order Newton process for (2.4). The proof in [7] is based on the following observations.

Assume that $x_k \in \text{int}(\mathcal{F})$. A Newton step for (2.4) satisfies

$$(D_k^{-2}\nabla^2 f(x_k) + \text{diag}(g_k)J_k^v)d_k = -D_k^{-2}g_k, \tag{2.5}$$

where $J^v(x) \in \Re^{n\times n}$ is the Jacobian matrix of $|v(x)|$ whenever $|v(x)|$ is differentiable. Since $|v_k| > 0$ by strict feasibility, following Definition 2.1, the only nondifferentiable points of possible concern occur when $g_{ki} = 0$ for some index i. If $g_i = 0$, we define the ith row J_i^v of J^v to be zero, i.e., $J_i^v \stackrel{\text{def}}{=} 0$. Nondifferentiability of this type is not a cause for concern because, for such a component, the value of v_i is not significant—local quadratic convergence can be achieved with nonlinear systems of this type [7]. It is clear that $J^v(x)$ is a diagonal matrix. Moreover, if all the components of l and u are finite, $J^v = \text{diag}(\text{sgn}(g))$. If a variable x_i has a finite lower bound and an infinite upper bound (or vice versa) and $g_i = 0$, then $J_{ii}^v = 0$.

Let $B(x)$ be an approximation to $\nabla^2 f(x)$. Based on the Newton step (2.5) for system (2.4), we define our quadratic model in the same way as in [7]:

$$\psi_k(s) \stackrel{\text{def}}{=} g_k^T s + \frac{1}{2}s^T M_k s, \tag{2.6}$$

[1] Notation: if a matrix A is a symmetric matrix then we write $A > 0$ to mean that A is positive definite; $A \geq 0$ means that A is positive semidefinite.

where

$$M_k \stackrel{\text{def}}{=} B_k + C_k, \tag{2.7}$$

$$C_k \stackrel{\text{def}}{=} D_k \text{diag}(g_k) J_k^v D_k. \tag{2.8}$$

It is clear that C_k is a positive semidefinite diagonal matrix.

Define

$$\hat{g}_k \stackrel{\text{def}}{=} D_k^{-1} g_k = \text{diag}(|v_k|^{\frac{1}{2}}) g_k,$$
$$\hat{M}_k \stackrel{\text{def}}{=} D_k^{-1} M_k D_k^{-1} = \text{diag}(|v_k|^{\frac{1}{2}}) B_k \text{diag}(|v_k|^{\frac{1}{2}}) + \text{diag}(g_k) J_k^v,$$
$$\hat{\psi}_k(\hat{s}) \stackrel{\text{def}}{=} \hat{g}_k^T \hat{s} + \frac{1}{2} \hat{s}^T \hat{M}_k \hat{s}.$$

The following lemma can be easily proved.

LEMMA 2.3. *Assume that $x_* \in \mathcal{F}$ and $B(x_*) = H_*$. Then*

(a) $\hat{g}_* = 0$ *if and only if* (2.2) *is satisfied*;

(b) $\hat{M}_*$ *is positive definite and* $\hat{g}_* = 0$ *if and only if the second-order sufficiency conditions are satisfied at* x_*;

(c) $\hat{M}_*$ *is positive semidefinite and* $\hat{g}_* = 0$ *if and only if the second-order necessary conditions are satisfied.*

Lemma 2.3 implies that x_k is a local minimizer of (1.1) if and only if $\hat{s} = 0$ is a solution to

$$\min_{\hat{s} \in \Re^n} \{\hat{\psi}_k(\hat{s}) : \ \|\hat{s}\| \leq \Delta_k\}. \tag{2.9}$$

Therefore, a solution of the subproblem (2.9) should yield a reasonable (trial) step when x_k is not a local minimizer. Let $s = D_k^{-1} \hat{s}$. Subproblem (2.9) is equivalent to the following problem in the original variable space:

$$\min_{s \in \Re^n} \{\psi_k(s) : \ \|D_k s\| \leq \Delta_k\}. \tag{2.10}$$

Moreover, in the neighborhood of a local minimizer, the Newton step defined by (2.5) for (2.4) is a solution to the trust region subproblem (2.10) if the trust region size Δ_k is sufficiently large.

The purpose of the scaling matrix D_k in (2.10) is distinctively different from the scaling matrix $\bar{D}_k$ used in unconstrained trust region methods, e.g., (1.2). The scaling matrix D_k is related to the distance to the boundary of the feasible region. Its purpose is to prevent a step directly toward a boundary point. In contrast, the scaling matrix $\bar{D}_k$, used in unconstrained trust region methods, is usually employed for numerical reasons—the scaling matrix $\bar{D}_k$ improves the conditioning of the problem.

If a bound-constrained problem (1.1) is badly scaled, the subproblem (2.10) can be replaced with

$$\min_{s \in \Re^n} \{\psi_k(s) : \ \|D_k \bar{D}_k s\| \leq \Delta_k\},$$

where $\bar{D}_k$ is chosen to improve the scaling and is a diagonal matrix with the property that $\{\bar{D}_k^{-1}\}$ is bounded and $\{\bar{D}_k\}$ is uniformly bounded. However, to emphasize the role of the scaling matrix D_k, we assume that $\bar{D}_k = I$.

Next we illustrate that it is possible to develop trust region methods for the bound-constrained problem (1.1) based on (2.10). First we introduce some notation and assumptions.

In the subsequent presentation, p_k denotes a global solution to (2.10). Assume that $d_k \in \Re^n$. The scalar α_k denotes the stepsize along d_k to the boundary:

$$\alpha_k \stackrel{\text{def}}{=} \min\left\{\max\left\{\frac{l_i - x_{ki}}{d_{ki}}, \frac{u_i - x_{ki}}{d_{ki}}\right\} : \ 1 \le i \le n\right\}, \tag{2.11}$$

and $\frac{l_i - x_{ki}}{d_{ki}} \equiv \frac{u_i - x_{ki}}{d_{ki}} \stackrel{\text{def}}{=} +\infty$ if $d_{ki} = 0$. (This is reasonable since $l_i < x_{ki} < u_i$ always holds.) If problem (1.1) is unconstrained, i.e., $l = -\infty$ and $u = \infty$, we also define $\alpha_k \stackrel{\text{def}}{=} +\infty$. We use $\psi_k^*[d_k]$ to denote the minimum value of $\psi_k(s)$ along the direction d_k within the feasible trust region, i.e.,

$$\psi_k^*[d_k] \stackrel{\text{def}}{=} \psi_k(\tau_k^* d_k) \stackrel{\text{def}}{=} \min\{\psi_k(\tau d_k) : \|\tau D_k d_k\| \le \Delta_k, x_k + \tau d_k \in \mathcal{F}\}. \tag{2.12}$$

Since we always require $x_k \in \text{int}(\mathcal{F})$, a possible step-back may be necessary to stay strictly feasible. Assume $\theta_l \in (0,1)$. We require that[2]

$$\theta_k \in [\theta_l, 1), \quad \theta_k - 1 = O(\|d_k\|), \quad \text{and} \quad \theta_k = 1 \text{ if } x_k + \tau_k^* d_k \in \text{int}(\mathcal{F}). \tag{2.13}$$

We use $\alpha_k^*[d_k]$ to denote the step obtained from d_k with a possible step-back. The exact definition of $\alpha_k^*[d_k]$ is

$$\alpha_k^*[d_k] \stackrel{\text{def}}{=} \theta_k \tau_k^* d_k. \tag{2.14}$$

We now state a few assumptions.

(AS.1) Given an initial point $x_0 \in \mathcal{F}$, it is assumed that $\mathcal{L}$ is compact, where $\mathcal{L}$ is the level set, i.e., $\mathcal{L} = \{x : \ x \in \mathcal{F} \text{ and } f(x) \le f(x_0)\}$.

(AS.2) There exists a positive scalar χ_B such that $\|B_k\| \le \chi_B$ for all k.

(AS.3) There exists a positive scalar χ_g such that for $x \in \mathcal{L}$, $\|g(x)\|_\infty < \chi_g$.

Assumption (AS.2) is also required in the convergence analysis of trust region methods for unconstrained problems. Assumption (AS.1) is needed for the boundedness of the scaling matrices $\{D_k^{-1}\}$. Condition (AS.3) is weak. It is satisfied, for example, if the gradient $g(x)$ is continuous on $\mathcal{L}$. Assumptions (AS.1) and (AS.2) imply that there exist positive scalars χ_D, χ_M such that

$$\|D_k^{-1}\| \le \chi_D, \quad \|\hat{M}_k\| \le \chi_M.$$

Note that $\{M_k\}$ is unbounded in general.

Next we present two trust region algorithms for the bound-constrained problem (1.1). The first, called the double-trust region method, is theoretically interesting. It illustrates that bound constraints can be handled by adjusting the trust region size for a 2-norm trust region subproblem with appropriate scaling matrices and the quadratic model. The second method, which we describe in §2.2, represents a more efficient approach.

2.1. The double-trust region method. Our objective is to develop a trust region method for (1.1) based on the trust region subproblem (2.10): a solution p_k to the trust region subproblem (2.10) is obtained and then truncated, i.e., $s_k = \alpha_k^*[p_k]$, to ensure strict feasibility.

The essential idea behind trust region methods is to adjust the trust region size to ensure a sufficient decrease of the objective function. Consider the unconstrained

[2] The notation $\eta_k = O(\rho_k)$ means that there exists a constant $\chi > 0$ such that $|\eta_k| \le \chi|\rho_k|$.

setting: the trust region size is updated to ensure that the reduction of the nonlinear function $f(x)$ is at least a fraction of the reduction of the quadratic model within the trust region. Specifically, the updating of the trust region size forces the condition

$$\rho_k^f = \frac{f(x_k + s_k) - f(x_k)}{\psi_k(s_k)} > \mu$$

for some constant $\mu > 0$. (We use the superscript f to emphasize the dependence on $f(x)$.) Since our quadratic model $\psi_k(s)$ is defined to include the constraint information, a natural extension of the definition of ρ_k^f to a bound-constrained problem (1.1) is given by

$$\rho_k^f \stackrel{\text{def}}{=} \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k}{\psi_k(s_k)}.$$

Similar to unconstrained trust region methods, ρ_k^f measures the agreement between the nonlinear function $f(x)$ and its quadratic approximation.

To obtain first-order convergence of unconstrained trust region methods, a sufficient reduction of the quadratic model $\psi_k(s)$ within the trust region is guaranteed if

$$\psi_k(s_k) < \beta \min\{\psi_k(s) : s = -\tau \bar{D}_k^{-1} \bar{D}_k^{-T} g_k, \|\bar{D}_k s\| \le \Delta_k\}, \ \|\bar{D}_k s_k\| \le \beta_0 \Delta_k \tag{2.15}$$

for some constants $\beta, \beta_0 > 0$. In our notation, (2.15) is equivalent to

$$\psi_k(s_k) < \beta \psi_k^*[-\bar{D}_k^{-1} \bar{D}_k^{-T} g_k], \quad \|\bar{D}_k s_k\| \le \beta_0 \Delta_k. \tag{2.16}$$

For problem (1.1), as we will prove in Lemma 3.1 in §3, if s_k satisfies

$$\psi_k(s_k) < \beta \psi_k^*[-D_k^{-2} g_k], \ \|D_k s_k\| \le \beta_0 \Delta_k, \ x_k + s_k \in \text{int}(\mathcal{F}), \tag{2.17}$$

the trust region model is sufficiently reduced to yield first-order convergence for our approach.

Unfortunately, a truncated step along an exact solution p_k of (2.10) may not sufficiently reduce the quadratic model $\psi_k(s)$ because of the effect of truncation, i.e., (2.17) is not guaranteed when $s_k = \alpha_k^*[p_k]$ and p_k solves (2.10). However, for any trust region subproblem with a nonzero gradient, the trust region solution approaches the gradient direction if the trust region size is reduced to zero. Since a step along the scaled steepest descent direction $-D_k^{-2} g_k$ does produce a sufficient reduction of $\psi_k(s)$, adjustment of the trust region size Δ_k can be used to ultimately guarantee a sufficient decrease. In particular, trust region size reduction can be used to force (2.17), i.e.,

$$\rho_k^c \stackrel{\text{def}}{=} \frac{\psi_k(s_k)}{\psi_k^*[-D_k^{-2} g_k]} > \beta.$$

Hence, we can adjust the trust region size so that both the quadratic model function $\psi_k(s)$ and the nonlinear function $f(x)$ are sufficiently reduced. This gives us the trust region algorithm, Algorithm 1, described in Figure 2.1. We call it the *double-trust region method* because the trust region size is adjusted for both nonlinearity and feasibility. For this method, an iteration is *successful* if both $\rho_k^f > \mu$ and $\rho_k^c > \beta$. Otherwise, an iteration is *unsuccessful.*

Algorithm 1. The double-trust region method
Let $0 < \mu, \beta < \eta < 1$, $x_0 \in \text{int}(\mathcal{F})$.
For $k = 0, 1, \ldots$

1. Compute $f(x_k)$, g_k, B_k and C_k; define the quadratic model $\psi_k(s) = g_k^T s + \frac{1}{2} s^T (B_k + C_k) s$.
2. Compute p_k, a solution to (2.10).
3. Compute

$$s_k = \alpha_k^*[p_k],$$
$$\rho_k^c = \frac{\psi_k(s_k)}{\psi_k^*[-D_k^{-2} g_k]},$$
$$\rho_k^f = \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k}{\psi_k(s_k)}.$$

4. If $\rho_k^f > \mu$ and $\rho_k^c > \beta$ then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.
5. Update the scaling matrix D_k and Δ_k as specified.

Updating Δ_k for the double-trust region method
Let $0 < \gamma_1 < 1 < \gamma_2$.

1. If $\rho_k^f \leq \mu$ or $\rho_k^c \leq \beta$ then set $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$.
2. If $\rho_k^f \in (\mu, \eta)$ then set $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$.
3. If $\rho_k^f \geq \eta$ then
   if $\rho_k^c \geq \eta$ set $\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k]$,
   if $\beta < \rho_k^c < \eta$ set $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$,
   if $\rho_k^c < \beta$ set $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$.

FIG. 2.1. *Double-trust region method for minimization subject to bounds.*

In §3, we will prove that the double-trust region method has reasonable convergence properties under the nondegeneracy assumption. Although the nondegeneracy assumption is impractical, we believe that the double-trust region method in Figure 2.1 is of theoretical interest. It indicates that if we allow the trust region size to be adjusted according to both accuracy of the quadratic approximation to the nonlinear objective function *and* how well the bound restriction is handled, we can have a trust method for the bound-constrained problem (1.1) based on a 2-norm trust region subproblem. In other words, bound constraints are dealt with *implicitly* using an approach modeled on trust region methods for unconstrained problems. Moreover, the double-trust region method motivates our allowance for possible trust region size reduction in the more practical model algorithm which is presented next, even when $f(x)$ is well represented by its quadratic approximation, i.e., $\rho_k^f > \eta$.

2.2. A practical trust region method. In the last section, we proposed a double-trust region method for bound-constrained problems (1.1) by solving an ellipsoidal trust region subproblem (2.10). In this algorithm, a sufficient decrease of the quadratic model is achieved by monitoring the ratios ρ_k^c and ρ_k^f and adjusting the trust region size accordingly. Since ρ_k^c is determined by $\psi_k(s_k)$, an exact solution to the subproblem (2.10) is assumed in order for ρ_k^c to be reliable. However, for large problems, the assumption that s_k be in the direction of the exact solution of the trust

region problem (2.10) is impractical. Moreover, the convergence of the method is established under the assumption that problem (1.1) is nondegenerate. In this section, we suggest a more efficient and practical model algorithm.

As we will see, a sufficient reduction of the quadratic model within the feasible trust region is not difficult to achieve; for example, moving along the scaled gradient $-D_k^{-2}g_k$ guarantees this. If we assume the availability of a step which sufficiently decreases the quadratic function within the feasible trust region, the trust region size is needed only to force the condition $\rho_k^f > \mu$ in the next iteration.

Algorithm 2. A more practical model

Let $0 < \mu < \eta < 1$ and $x_0 \in \text{int}(\mathcal{F})$.

For $k = 0, 1, \ldots$

1. Compute $f(x_k)$, g_k, B_k, and C_k; define the quadratic model $\psi_k(s) = g_k^T s + \frac{1}{2}s^T(B_k + C_k)s$.
2. Compute s_k, based on (2.10), such that $x_k + s_k \in \text{int}(\mathcal{F})$.
3. Compute

$$\rho_k^f = \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2}s_k^T C_k s_k}{\psi_k(s_k)}.$$

4. If $\rho_k^f > \mu$ then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.
5. Update the scaling matrix D_k and Δ_k as specified.

Updating trust region size Δ_k

Let $0 < \gamma_1 < 1 < \gamma_2$ and $\Lambda_l > 0$ be given.

1. If $\rho_k^f \leq \mu$ then set $\Delta_{k+1} \in (0, \gamma_1\Delta_k]$.
2. If $\rho_k^f \in (\mu, \eta)$ then set $\Delta_{k+1} \in [\gamma_1\Delta_k, \Delta_k]$.
3. If $\rho_k^f \geq \eta$ then
   if $\Delta_k > \Lambda_l$ then
   set $\Delta_{k+1} \in$ either $[\gamma_1\Delta_k, \Delta_k]$ or $[\Delta_k, \gamma_2\Delta_k]$;
   otherwise,
   set $\Delta_{k+1} \in [\Delta_k, \gamma_2\Delta_k]$.

FIG. 2.2. *Trust region method for minimization subject to bounds.*

In Figure 2.2, we describe a trust region method for bound-constrained problems in which the trust region size is primarily updated according to ρ_k^f. However, motivated by the double-trust region method in Figure 2.1, we allow more freedom than usual in the adjustment of Δ_k to permit further reduction in Δ_k even when $\rho_k^f \geq \eta$, thus encouraging the use of the trust region step (2.10).

To satisfy the first-order necessary conditions, given two positive constants β and β_0, it is required that the approximate trust region solution s_k satisfy

$$\begin{aligned} &\psi_k(s_k) < \beta\psi_k^*[-D_k^{-2}g_k], \\ &\|D_k s_k\| \leq \beta_0\Delta_k, \ x_k + s_k \in \text{int}(\mathcal{F}). \end{aligned} \tag{AS.4}$$

In other words, we require that $\psi_k(s_k)$ be less than a fraction of the minimum of $\psi_k(s)$ along the scaled gradient $-D_k^{-2}g_k$ within the feasible trust region. We point out that condition (AS.4) is satisfied for every successful iteration of Algorithm 1. For Algorithm 2, an iteration is *successful* if the condition $\rho_k^f > \mu$ holds. Otherwise, an iteration is *unsuccessful.*

Condition (AS.4) can be easily satisfied for $0 < \beta < 1$. Let d_k be the solution to $\min\{\psi_k(s) : D_k^2 s = -\nu g_k,\ \|D_k s\| \le \Delta_k,\ x_k + s \in \mathcal{F}\}$. Then $s_k = d_k$ satisfies (AS.4), except for the possible violation of $x_k + s_k \in \text{int}(\mathcal{F})$. Assume that $x_k + d_k \notin \text{int}(\mathcal{F})$. Since $\psi_k(s)$ is continuous, a small step-back $s_k = \theta_k d_k$, where $\theta_l < \theta_k < 1$, can ensure both the condition $x_k + s_k \in \text{int}(\mathcal{F})$ and $\psi_k(s_k) < \beta\psi_k^*[-D_k^{-2} g_k]$.

Assumption (AS.4) will not necessarily guarantee a solution at which the second-order necessary conditions are satisfied. To achieve this we make the following stronger assumptions on the quadratic model and the approximate solution:

(AS.5) $$\psi_k(s) = g_k^T s + \frac{1}{2} s^T(\nabla^2 f(x_k) + C_k)s, \text{i.e., } B_k = \nabla^2 f(x_k).$$

(AS.6) Assume that p_k is a solution to $\min_{s\in\Re^n}\{\psi_k(s) : \|D_k s\| \le \Delta_k\}$ and β^q and β_0^q are two positive constants. Then s_k satisfies $\psi_k(s_k) < \beta^q\psi_k^*[p_k]$, $\|D_k s_k\| \le \beta_0^q \Delta_k$, $x_k + s_k \in \text{int}(\mathcal{F})$.

Since both conditions (AS.4) and (AS.6) can be satisfied by simply solving a quadratic trust region subproblem $\min_{s\in\Re^n}\{\psi_k(s) : \|D_k s\| \le \Delta_k\}$, it is *not* necessary to solve a quadratic programming subproblem to achieve convergence. For example, one can first compute a solution p_k to the unconstrained trust region problem

$$\min_{s\in\Re^n}\{\psi_k(s) : \|D_k s\| \le \Delta_k\}$$

and then choose s_k so that $x_k + s_k \in \text{int}(\mathcal{F})$ and $\psi_k(s_k)$ is the minimum of the values $\psi_k^*[p_k]$ and $\psi_k^*[-D_k^{-2} g_k]$. However, requirements (AS.4) and (AS.6) are not very restrictive. There are many ways of computing such approximations. As another example, one can consider the reflection techniques used in [7]. It is also possible to have a subspace adaptation of this trust region approach in which low-dimension subspace trust region problems are solved. We leave the investigation of these computational issues to a subsequent paper.

Before we study the convergence properties of the two trust region methods proposed, we make the following important observation. If we assume that $l = -\infty$ and $u = +\infty$, then $C_k = 0$ and $D_k = I$ and the quadratic model is the same as that for unconstrained problems. Moreover, the conditions (AS.4) and (AS.6) are the same as the conditions required for unconstrained trust region methods (e.g., [17]) since the feasibility constraints are always satisfied.

3. Convergence properties. The convergence proofs for the double-trust region algorithm in Figure 2.1 and the practical algorithm in Figure 2.2 follow the same main steps. Lemmas 3.2 and 3.3 make it possible to present the proofs for both algorithms simultaneously in a clean fashion. The major results are Theorems 3.4, 3.5, 3.10, and 3.11.

The main difference between the practical method and the double-trust region method is that the condition (AS.4) is assumed by the first but satisfied for the latter by adjustment of the trust region size by monitoring the ratio ρ_k^c. However, a common property of the two methods is that, for any successful iteration, (AS.4) is satisfied. Moreover, condition (AS.6) is always satisfied for the double-trust region method.

The convergence results of the two methods are similar. However, the assumptions required by the double-trust region method are stronger—for Algorithm 1 we assume that problem (1.1) is nondegenerate. This nondegeneracy assumption is not needed for Algorithm 2.

The following result is required to express (AS.4) in a manageable form. It is similar to Lemma (4.8) in [17].

LEMMA 3.1. *Assume that* (AS.1)–(AS.3) *are satisfied. If* s_k *satisfies* (AS.4) *then*

$$-\psi_k(s_k) \geq \frac{1}{2}\beta\|\hat{g}_k\| \min\left\{\Delta_k, \frac{\|\hat{g}_k\|}{\|\hat{M}_k\|}, \frac{\|\hat{g}_k\|}{\|g_k\|_\infty}\right\}.$$

Proof. Define $\phi(\tau) : \Re \to \Re$ by setting $d_k = -D_k^{-1}\frac{\hat{g}_k}{\|\hat{g}_k\|}$ and

$$\phi(\tau) \stackrel{\text{def}}{=} \psi_k(\tau d_k).$$

Let τ_k^* be the minimizer of ϕ on $[0, \min\{\Delta_k, \alpha_k\}]$, where α_k is the stepsize along d_k to the boundary as defined in (2.11):

$$\alpha_k = \min\left\{\max\left\{\frac{l_i - x_{ki}}{d_{ki}}, \frac{u_i - x_{ki}}{d_{ki}}\right\} : \ 1 \leq i \leq n\right\}.$$

Since $\alpha_k > 0$ (recall that $x_k \in \text{int}(\mathcal{F})$) and the components of d_k have the same sign as that of $-g_k$, it is easy to verify that

$$\alpha_k = \frac{|v_{kj}|}{|d_{kj}|} = \frac{|v_{kj}|\|\hat{g}_k\|}{|v_{kj}||g_{kj}|} \quad \text{for some } j.$$

Hence

$$\alpha_k \geq \frac{\|\hat{g}_k\|}{\|g_k\|_\infty}.$$

By the definition of $\phi(\tau)$,

$$\phi(\tau) = -\tau\|\hat{g}_k\| + \frac{1}{2}\tau^2\mu_k, \quad \mu_k \stackrel{\text{def}}{=} \frac{\hat{g}_k^T\hat{M}_k\hat{g}_k}{\|\hat{g}_k\|^2}.$$

If $\tau_k^* \in [0, \min\{\Delta_k, \alpha_k\})$, then $\mu_k \geq 0$ (thus $\hat{g}_k^T\hat{M}_k\hat{g}_k \geq 0$) and $\tau_k^* = \frac{\|\hat{g}_k\|}{\mu_k}$. Thus

$$\phi(\tau_k^*) = -\frac{1}{2}\frac{\|\hat{g}_k\|^2}{\mu_k} \leq -\frac{1}{2}\frac{\|\hat{g}_k\|^2}{\|\hat{M}_k\|}.$$

Assume $\tau_k^* = \Delta_k$. Since $\mu_k\Delta_k \leq \|\hat{g}_k\|$ when $\mu_k > 0$, and $\phi(\tau_k^*) \leq -\Delta_k\|\hat{g}_k\|$ otherwise, we have

$$\phi(\tau_k^*) = \phi(\Delta_k) \leq -\frac{1}{2}\Delta_k\|\hat{g}_k\|.$$

Assume $\tau_k^* = \alpha_k$. Since $\mu_k\alpha_k \leq \|\hat{g}_k\|$ when $\mu_k > 0$, and $\phi(\tau_k^*) \leq -\alpha_k\|\hat{g}_k\|$ otherwise, we have

$$\phi(\tau_k^*) = \phi(\alpha_k) \leq -\frac{1}{2}\alpha_k\|\hat{g}_k\| \leq -\frac{1}{2}\frac{\|\hat{g}_k\|^2}{\|g_k\|_\infty}.$$

Since $\psi_k(s_k) \leq \beta\phi(\tau_k^*)$ by (AS.4), the result follows from the two previous estimates. □

Assume that the kth iteration is successful from either Algorithm 1 or Algorithm 2. From Lemma 3.1,

$$f(x_k) - f(x_{k+1}) - \frac{1}{2}s_k^T C_k s_k > -\mu\psi_k(s_k)$$
$$\geq \frac{1}{2}\beta\mu\|\hat{g}_k\| \min\left\{\Delta_k, \frac{\|\hat{g}_k\|}{\|\hat{M}_k\|}, \frac{\|\hat{g}_k\|}{\|g_k\|_\infty}\right\}.$$

Hence, under assumptions (AS.1), (AS.2), (AS.3), and (AS.4),

$$f(x_k) - f(x_{k+1}) > \frac{1}{2}s_k^T C_k s_k + \frac{1}{2}\beta\mu\|\hat{g}_k\| \min\left\{\Delta_k, \frac{\|\hat{g}_k\|}{\chi_M}, \frac{\|\hat{g}_k\|}{\chi_g}\right\}.$$

Note that $s_k^T C_k s_k \geq 0$. The reduction in f is guaranteed to be better than a multiple of the reduction achieved in the (negative) scaled gradient direction, i.e.,

$$f(x_k) - f(x_{k+1}) > \frac{1}{2}\beta\mu\|\hat{g}_k\| \min\left\{\Delta_k, \frac{\|\hat{g}_k\|}{\chi_M}, \frac{\|\hat{g}_k\|}{\chi_g}\right\}. \tag{3.1}$$

This inequality is important for the convergence proof.

Next, in Theorems 3.4 and 3.5, we prove that the first-order necessary conditions are satisfied at every limit point of $\{x_k\}$. Several technical results are required first.

Recall that p_k is a global solution to the trust region subproblem (2.10). Using Theorem (3.11) in [17], there exists a parameter λ_k and an upper triangular matrix $R_k \in \Re^{n\times n}$ such that

$$\hat{M}_k + \lambda_k I = R_k^T R_k, \quad (\hat{M}_k + \lambda_k I)D_k p_k = -\hat{g}_k, \ \lambda_k \geq 0, \tag{3.2}$$

with $\lambda_k(\Delta_k - \|D_k p_k\|) = 0$. Equivalently, p_k is the solution to

$$(\lambda_k I + D_k^{-1} C_k D_k^{-1})p_k = -D_k^{-2}(g_k + B_k p_k). \tag{3.3}$$

LEMMA 3.2. *Suppose that $\{x_k\}$ is a sequence generated by Algorithm 1. Assume that problem (1.1) is nondegenerate; (AS.1), (AS.2), and (AS.3) hold; and $\{x_k\}$ converges. If $\{\Delta_k\}$ converges to zero and $\liminf_{k\to\infty}\|\hat{g}_k\| > 0$, then $\rho_k^c \geq 1$ for sufficiently large k.*

Proof. Since $\{\Delta_k\}$ converges to zero and $\|D_k p_k\| \leq \Delta_k$, it follows that $\{D_k p_k\}$ converges to zero. But $\|p_k\| \leq \chi_D\|D_k p_k\|$. Hence $\{p_k\}$ converges to zero. Using (AS.2), $\{B_k p_k\}$ converges to zero. From (2.8), $D_k^{-1}C_k D_k^{-1} = \operatorname{diag}(g_k)J_k^v$ is positive semidefinite and bounded. Together with the assumption $\liminf_{k\to\infty}\|\hat{g}_k\| > 0$ and (3.3), it is clear that $\{\lambda_k\}$ converges to $+\infty$.

Assume that α_k is the stepsize to the boundary of the constraints along p_k. From (2.11),

$$\alpha_k = \min\left\{\max\left\{\frac{l_i - x_{ki}}{p_{ki}}, \frac{u_i - x_{ki}}{p_{ki}}\right\} : \ 1 \leq i \leq n\right\}.$$

Consider a limit point of $\{x_k\}$. Since $\{p_k\}$ converges to zero and the problem (1.1) is nondegenerate, it is clear that

$$\lim_{k\to\infty} \max\left\{\frac{l_j - x_{kj}}{p_{kj}}, \frac{u_j - x_{kj}}{p_{kj}}\right\} = +\infty \quad \forall j \text{ with } \ g_{*j} = 0.$$

Assume now that $g_{*j} \neq 0$. Since $\{B_k p_k\}$ converges to zero and using (3.3), for k sufficiently large, $-g_{kj}$ and p_{kj} have the same sign. Hence, if

$$\alpha_k = \max\left\{\frac{l_j - x_{kj}}{p_{kj}}, \frac{u_j - x_{kj}}{p_{kj}}\right\},$$

for some j with $g_{*j} \neq 0$, $\alpha_k = \frac{|v_{kj}|}{|p_{kj}|}$. Using $\alpha_k = \frac{|v_{kj}|}{|p_{kj}|}$, (3.3), the boundedness of $\{(g_k + B_k p_k)\}$, and $\{\lambda_k\}$ converging to $+\infty$, we conclude that $\lim_{k\to\infty} \alpha_k = +\infty$.

Subsequently, $s_k = \alpha_k^*[p_k] = p_k$. Thus, for k sufficiently large,

$$\min\{\psi_k(s) : s = \tau p_k,\ \|D_k s\| \leq \Delta_k,\ x_k + s \in \mathcal{F}\} = \min\{\psi_k(s) : s = \tau p_k,\ \|D_k s\| \leq \Delta_k\}.$$

Hence

$$\psi_k(p_k) = \min\{\psi_k(s) : s = \tau p_k,\ \|D_k s\| \leq \Delta_k,\ x_k + s \in \mathcal{F}\} \leq \psi_k^*[-D_k^{-2} g_k],$$

and therefore

$$\rho_k^c = \frac{\psi_k(s_k)}{\psi_k^*[-D_k^{-2} g_k]} = \frac{\psi_k(p_k)}{\psi_k^*[-D_k^{-2} g_k]} \geq 1. \qquad \square$$

LEMMA 3.3. *Assume that $\{\Delta_k\}$ is updated by Algorithm 2. If $\rho_k^f \geq \eta$ for sufficiently large k, then $\{\Delta_k\}$ is bounded away from zero.*

Proof. By assumption, there exists $\bar{k}$ such that when $k \geq \bar{k}$, $\rho_k^f \geq \eta$. We prove, by induction, that for $k \geq \bar{k}$,

$$\Delta_k \geq \min\{\gamma_1 \Lambda_l, \Delta_{\bar{k}}\}. \tag{3.4}$$

First, it is clear that (3.4) is true when $k = \bar{k}$.

Assume that (3.4) is true for $k \geq \bar{k}$, i.e., $\Delta_k \geq \min\{\gamma_1 \Lambda_l, \Delta_{\bar{k}}\}$. If $\Delta_k \leq \Lambda_l$, $\Delta_{k+1} \geq \Delta_k \geq \min\{\gamma_1 \Lambda_l, \Delta_{\bar{k}}\}$. If $\Delta_k > \Lambda_l$, $\Delta_{k+1} \geq \min\{\gamma_1 \Lambda_l, \Delta_{\bar{k}}\}$.

Hence (3.4) is true for all $k \geq \bar{k}$ and $\{\Delta_k\}$ is bounded away from zero. $\square$

The proof of the following theorem is a slight modification of Theorem (4.10) in [17].

THEOREM 3.4. *Assume that $f : \Re^n \to \Re$ is continuously differentiable on $\mathcal{F}$ and* (AS.1), (AS.2), *and* (AS.3) *hold. For Algorithm 2, if $\{s_k\}$ satisfies* (AS.4), *then*

$$\liminf_{k\to\infty} \|\hat{g}_k\| = 0. \tag{3.5}$$

For Algorithm 1, (3.5) *is true under the further assumption that problem* (1.1) *is nondegenerate.*

Proof. We must show that $\{\|\hat{g}_k\|\}$ is not bounded away from zero. Assume that there is an $\epsilon > 0$ such that $\|\hat{g}_k\| \geq \epsilon$ for all sufficiently large k. We now show that

$$\sum_{k=1}^{\infty} \Delta_k < +\infty. \tag{3.6}$$

If there are a finite number of successful iterations, then $\Delta_{k+1} \leq \gamma_1 \Delta_k$ for all k sufficiently large and then (3.6) clearly holds. Assume now that there is an infinite

sequence $\{k_i\}$ of successful iterations. Since $\{f(x_k)\}$ is nonincreasing and bounded below,

$$0 \leq \sum_{k=0}^{\infty} (f(x_k) - f(x_{k+1})) < +\infty.$$

This last inequality, (3.1), and $\|\hat{g}_k\| \geq \epsilon$ imply that

$$\sum_{i=1}^{\infty} \Delta_{k_i} < +\infty.$$

Similar to the updating rule in Figure 1.1 for unconstrained trust region methods, the updating rules of Algorithms 1 and 2 specify that $\Delta_{k+1} \leq \gamma_1 \Delta_k$ for an unsuccessful iteration and $\Delta_{k+1} \leq \gamma_2 \Delta_k$ for a successful iteration. It is easy to verify that the following holds:

$$\sum_{k=1}^{\infty} \Delta_k \leq \left(1 + \frac{\gamma_2}{1-\gamma_1}\right) \sum_{i=1}^{\infty} \Delta_{k_i},$$

and thus (3.6) holds in this case as well.

Next we prove that (3.6) implies that $\{|\rho_k^f - 1|\}$ converges to zero. First,

$$\|x_{k+1} - x_k\| \leq \|s_k\| \leq \chi_D \beta_0 \Delta_k,$$

and hence (3.6) shows that $\{x_k\}$ converges. Now (AS.1) and (AS.2) imply that

$$\begin{aligned} \left|\psi_k(s_k) - g_k^T s_k - \frac{1}{2} s_k^T C_k s_k\right| &= \left|\frac{1}{2} s_k^T B_k s_k\right| \\ &\leq \frac{1}{2} \chi_B \chi_D^2 \|D_k s_k\|^2. \end{aligned}$$

But $\|D_k s_k\| \leq \beta_0 \Delta_k$. Therefore, using $f(x_k + s_k) - f(x_k) = \bar{g}_k^T s_k$, where $\bar{g}_k \stackrel{\text{def}}{=} \nabla f(x_k + \xi_k s_k)$ with $0 \leq \xi_k \leq 1$, we have

$$\begin{aligned} &\left|f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k - \psi_k(s_k)\right| \\ &\leq \left|\psi_k(s_k) - g_k^T s_k - \frac{1}{2} s_k^T C_k s_k\right| + |(\bar{g}_k - g_k)^T s_k| \\ &\leq \frac{1}{2} \chi_B \beta_0^2 \chi_D^2 \Delta_k^2 + \chi_D \beta_0 \Delta_k \|g_k - \bar{g}_k\|. \end{aligned}$$

This inequality, the continuity of $g(x)$, and the fact that $\{x_k\}$ converges indicate that there exists a sequence $\{\epsilon_k\}$ converging to zero such that

$$\left|f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k - \psi_k(s_k)\right| \leq \epsilon_k \Delta_k.$$

Since Lemma 3.1 implies that

$$-\psi_k(s_k) \geq \frac{1}{2} \beta \epsilon \Delta_k,$$

we readily obtain that $\{|\rho_k^f - 1|\}$ converges to zero.

Using Lemma 3.3, $\{\Delta_k\}$ cannot converge to zero in Algorithm 2. This contradicts (3.6) and establishes the result.

For Algorithm 1, using Lemma 3.2, Δ_k is not decreased for sufficiently large k. Thus $\{\Delta_k\}$ cannot converge to zero, contradicting (3.6). The result follows. □

The next theorem establishes that $\{D_k^{-1}g_k\}$ converges to zero. This result is obtained despite the fact that $\{D_k\}$ is not uniformly bounded. This may be somewhat surprising since the analogous convergence result in the unconstrained setting requires the sequence of diagonal scaling matrices to be uniformly bounded. The proof of this theorem is similar to that of Theorem (4.14) in Moré [17].

THEOREM 3.5. *Assume* (AS.1) *and* (AS.2) *hold and* $g(x)$ *is continuous on* $\mathcal{F}$. *If* $\{x_k\}$ *is generated by Algorithm* 2 *and* (AS.4) *holds for* s_k, *then*

$$\lim_{k\to\infty} \|D_k^{-1}g_k\| = 0. \tag{3.7}$$

Result (3.7) *also holds for Algorithm* 1 *when problem* (1.1) *is nondegenerate.*

Proof. The proof is by contradiction and is the same for Algorithms 1 and 2. (The nondegeneracy assumption is needed for Algorithm 1 because the proof uses Theorem 3.4.)

Let ϵ_1 in $(0,1)$ be given and assume that there is a sequence $\{m_i\}$ such that $\|\hat{g}_{m_i}\| \geq \epsilon_1$. Theorem 3.4 guarantees that for any ϵ_2 in $(0,\epsilon_1)$ there is a subsequence of $\{m_i\}$ (without loss of generality we assume that it is the full sequence) and a sequence $\{l_i\}$ such that

$$\|\hat{g}_k\| \geq \epsilon_2, \quad m_i \leq k < l_i, \quad \|\hat{g}_{l_i}\| < \epsilon_2. \tag{3.8}$$

If the kth iteration is successful, then according to (3.1),

$$f(x_k) - f(x_{k+1}) > \frac{1}{2}\beta\mu\epsilon_2 \min\left\{\Delta_k, \frac{\epsilon_2}{\chi_M}, \frac{\epsilon_2}{\chi_g}\right\}, \quad m_i \leq k < l_i.$$

Since $f(x)$ is bounded below on $\mathcal{L}$ and the sequence $\{f(x_k)\}$ is nonincreasing, $\{f(x_k)\}$ converges and $\{f(x_k) - f(x_{k+1})\}$ converges to zero. From $\|x_{k+1} - x_k\| \leq \beta_0\chi_D\Delta_k$, it follows that, for sufficiently large i,

$$f(x_k) - f(x_{k+1}) \geq \epsilon_3\|x_{k+1} - x_k\|, \quad m_i \leq k < l_i, \tag{3.9}$$

where $\epsilon_3 = (\frac{1}{2}\beta\mu\epsilon_2)/(\beta_0\chi_D)$. Using (3.9) and the triangle inequality,

$$f(x_{m_i}) - f(x_{k_i}) \geq \epsilon_3\|x_{k_i} - x_{m_i}\|, \quad m_i \leq k_i \leq l_i.$$

The uniform continuity of $g(x)$ (since $\mathcal{L}$ is compact) and the convergence of $\{f(x_k)\}$ can now be used to deduce that

$$\|g_{m_i} - g_{l_i}\| \leq \epsilon_2 \tag{3.10}$$

for i sufficiently large.

Consider a subsequence of l_i (without loss of generality assume that it is the full sequence) such that $\{x_{l_i}\}$ converges to x_*. Then $\{x_{m_i}\}$ converges to x_*. Based on the definition of $v(x)$, if the jth component of g_* is nonzero, then, for i sufficiently large, the corresponding component of $||v_{m_i}| - |v_{l_i}||$ is no greater than that of $|x_{m_i} - x_{l_i}|$. Thus $\{\mathrm{diag}(|v_{m_i}|^{\frac{1}{2}} - |v_{l_i}|^{\frac{1}{2}})g_{l_i}\}$ converges to zero. Therefore, for i sufficiently large,

$$\|(D_{m_i}^{-1} - D_{l_i}^{-1})g_{l_i}\| = \|\mathrm{diag}(|v_{m_i}|^{\frac{1}{2}} - |v_{l_i}|^{\frac{1}{2}})g_{l_i}\| \leq \epsilon_2. \tag{3.11}$$

Using the triangle inequality for any m and l,

$$\|\hat{g}_m\| \leq \|D_m^{-1}\|\|g_m - g_l\| + \|(D_m^{-1} - D_l^{-1})g_l\| + \|\hat{g}_l\|. \tag{3.12}$$

Combining (3.12) with (3.8), (3.10), and (3.11), we obtain that

$$\epsilon_1 \leq (\chi_D + 2)\epsilon_2.$$

Since ϵ_2 can be any number in $(0, \epsilon_1)$, this is a contradiction. □

Next we consider the second-order necessary conditions. As mentioned in (AS.5) of §2.2, we assume the following quadratic model: $\psi_k(s) = g_k^T s + \frac{1}{2}s^T(\nabla^2 f(x_k) + C_k)s$. Moreover, in addition to (AS.4), the condition (AS.6) holds, i.e., the reduction of the quadratic model satisfies

$$\psi_k(s_k) < \beta^q \min\{\psi_k(s) : s = \tau p_k,\ \|D_k s\| \leq \Delta_k,\ x_k + s \in \mathcal{F}\},$$
$$\|D_k s_k\| \leq \beta_0^q \Delta_k,\ x_k + s_k \in \operatorname{int}(\mathcal{F}),$$

where p_k is a global solution to the unconstrained subproblem

$$\min_{s \in \Re^n} \{\psi_k(s) : \|D_k s\| \leq \Delta_k\}.$$

Before we state the second-order convergence result, several technical lemmas are required. First, we quote Lemma (4.10) in [18].

LEMMA 3.6. *Let x_* be an isolated limit point of a sequence $\{x_k\}$ in $\Re^n$. If $\{x_k\}$ does not converge, then there is a subsequence x_{l_j} which converges to x_* and an $\epsilon > 0$ such that*

$$\|x_{l_j+1} - x_{l_j}\| \geq \epsilon.$$

Now we examine the consequences of (AS.6) in greater detail. Recall (3.2): there exists a parameter λ_k such that

$$(\hat{M}_k + \lambda_k I) = R_k^T R_k,\ \ (\hat{M}_k + \lambda_k I)D_k p_k = -\hat{g}_k,\ \ \lambda_k \geq 0$$

with $\lambda_k(\Delta_k - \|D_k p_k\|) = 0$. Furthermore, as mentioned before, p_k satisfies

$$(\lambda_k I + D_k^{-2} C_k)p_k = -D_k^{-2}(g_k + B_k p_k),$$

which is equation (3.3). These equations will be used repeatedly in the subsequent proofs.

LEMMA 3.7. *Assume that* (AS.6) *is satisfied. Then*

$$-\psi_k(s_k) \geq \beta^q \psi_k^*[p_k] \geq \frac{\beta^q}{2}[\min\{1, \alpha_k^2\}\lambda_k \Delta_k^2 + \min\{1, \alpha_k\}\|R_k D_k p_k\|^2],$$

where α_k is the stepsize along p_k to the boundary and p_k is a global solution to the trust region subproblem (2.10).

Proof. Let $\phi(\tau) \overset{\text{def}}{=} \psi_k(\tau p_k)$ and $\tau \in [0, \min\{1, \alpha_k\}]$, where α_k is the stepsize along p_k to the boundary.

It is easy to see that

$$\begin{aligned}
\phi(\tau) &= \tau g_k^T p_k + \frac{1}{2}\tau^2 p_k^T M_k p_k \\
&= \tau \hat{g}_k^T D_k p_k + \frac{1}{2}\tau^2 (D_k p_k)^T \hat{M}_k D_k p_k \\
&= \tau \hat{g}_k^T D_k p_k - \frac{1}{2}\tau^2 \hat{g_k}^T D_k p_k - \frac{1}{2}\tau^2 \lambda_k \|D_k p_k\|^2 \quad \text{(from (3.2))} \\
&= -\tau \|R_k D_k p_k\|^2 + \frac{1}{2}\tau^2 \|R_k D_k p_k\|^2 - \frac{1}{2}\tau^2 \lambda_k \Delta_k^2 \quad \text{(from (3.2))}.
\end{aligned}$$

But $\tau^2 \leq \tau$ since $\tau \leq 1$. Let τ_k^* be the minimizer of $\phi(\tau)$ in $[0, \min\{1, \alpha_k\}]$. Then $\tau_k^* = \min\{1, \alpha_k\}$ since p_k solves the trust region subproblem (2.10). From (AS.6),

$$-\psi_k(s_k) \geq -\beta^q \phi(\tau_k^*) \geq \frac{\beta^q}{2}[\min\{1, \alpha_k^2\}\lambda_k \Delta_k^2 + \min\{1, \alpha_k\}\|R_k D_k p_k\|^2]. \qquad \square$$

The following lemma provides estimates of the reductions in the objective function and the quadratic model. We emphasize that the results hold for *any subsequence* generated by the algorithms. (Consequently, they hold for the entire sequence as well.)

LEMMA 3.8. *Assume that the conditions of Theorem* 3.5 *and* (AS.6) *hold. Furthermore,* $\{x_k\}$ *is any subsequence generated by either Algorithm* 1 *or Algorithm* 2. *If every limit point of* $\{x_k\}$ *is nondegenerate, then there exists* $0 < \epsilon_0 < 1$ *such that, for* k *sufficiently large,*

$$-\psi_k(s_k) \geq \frac{\beta^q}{2} \min\left\{1, \frac{\lambda_k^2 \epsilon_0^2}{[(\chi_g + \Delta_k \chi_B \chi_D)\chi_D^2]^2}, \frac{\lambda_k^2}{(\chi_g + \Delta_k \chi_B \chi_D)^2}\right\} \lambda_k \Delta_k^2,$$

and if the k*th iteration is successful, then*

$$f(x_k) - f(x_{k+1}) > \frac{\beta^q}{2} \mu \min\left\{1, \frac{\lambda_k^2 \epsilon_0^2}{[(\chi_g + \Delta_k \chi_B \chi_D)\chi_D^2]^2}, \frac{\lambda_k^2}{(\chi_g + \Delta_k \chi_B \chi_D)^2}\right\} \lambda_k \Delta_k^2.$$

Proof. Using Lemma 3.7,

$$-\psi_k(s_k) \geq \frac{\beta^q}{2} \min\{1, \alpha_k^2\}\lambda_k \Delta_k^2, \tag{3.13}$$

where α_k is the stepsize along p_k to the boundary as defined in (2.11):

$$\alpha_k = \min\left\{\max\left\{\frac{l_i - x_{ki}}{p_{ki}}, \frac{u_i - x_{ki}}{p_{ki}}\right\} : 1 \leq i \leq n\right\}. \tag{3.14}$$

Since the problem is nondegenerate at every limit point and $\{x_k\}$ is bounded, there exists $0 < \epsilon_0 < 1$ and $2\epsilon_0 < \min(u - l)$, such that, for sufficiently large k,

$$\min(x_k - l, u - x_k) + |g_k| > 2\epsilon_0 e, \quad e = [1, \ldots, 1]^T \in \Re^n. \tag{3.15}$$

(Otherwise, there would be a degenerate limit point of $\{x_k\}$.)

Following Theorem 3.5, $\{D_k^{-1} g_k\}$ converges to zero. Hence, for sufficiently large k,

$$\|D_k^{-2} g_k\|_\infty < \epsilon_0^2. \tag{3.16}$$

Assume that k is sufficiently large and

$$\alpha_k = \max\left\{\frac{l_j - x_{kj}}{p_{kj}}, \frac{u_j - x_{kj}}{p_{kj}}\right\} \quad \text{for some } j. \tag{3.17}$$

Recall that

$$\operatorname{diag}(g_k) J_k^v \geq 0 \tag{3.18}$$

and J_k^v is a diagonal matrix.

If $|g_{kj}| \leq \epsilon_0$, using (3.15), we have

$$\min\{x_{kj} - l_j, u_j - x_{kj}\} > \epsilon_0.$$

Hence, from (3.3), (3.17), (3.18), and

$$\|g_k + B_k p_k\|_\infty \leq \chi_g + \chi_B \|D_k^{-1}\| \|D_k p_k\| \leq \chi_g + \chi_B \chi_D \Delta_k,$$

we obtain

$$\alpha_k \geq \frac{\lambda_k \epsilon_0}{(\chi_g + \Delta_k \chi_B \chi_D)\chi_D^2}.$$

If $|g_{kj}| > \epsilon_0$, then $|v_{kj}| < \epsilon_0$ because of (3.16). If $\alpha_k = \frac{|v_{kj}|}{|p_{kj}|}$, then from (3.17), (3.3), and (3.18), and noting that $|v_{kj}|$ in the numerator is cancelled with D_{kjj}^{-2} in the denominator,

$$\alpha_k \geq \frac{\lambda_k}{\chi_g + \Delta_k \chi_B \chi_D}.$$

Assume that $\alpha_k \neq \frac{|v_{kj}|}{|p_{kj}|}$. Since $|v_{kj}| < \epsilon_0$, $u_j - l_j > 2\epsilon_0$, and $\alpha_k \neq \frac{|v_{kj}|}{|p_{kj}|}$, the magnitude of the numerator determining α_k is greater than ϵ_0. Hence, using (3.3), (3.17), and (3.18),

$$\alpha_k \geq \frac{\lambda_k \epsilon_0}{(\chi_g + \Delta_k \chi_B \chi_D)\chi_D^2}.$$

Using (3.13), we have

$$-\psi_k(s_k) \geq \frac{\beta^q}{2} \min\left\{1, \frac{\lambda_k^2 \epsilon_0^2}{[(\chi_g + \Delta_k \chi_B \chi_D)\chi_D^2]^2}, \frac{\lambda_k^2}{(\chi_g + \Delta_k \chi_B \chi_D)^2}\right\} \lambda_k \Delta_k^2.$$

If the kth iteration is successful, $\rho_k^f > \mu$. Hence

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq -\mu\psi_k(s_k) + \frac{1}{2} s_k^T C_k s_k \\ &\geq -\mu\psi_k(s_k) \qquad (\text{since } s_k^T C_k s_k \geq 0) \\ &> \frac{\beta^q}{2} \mu \min\left\{1, \frac{\lambda_k^2 \epsilon_0^2}{[(\chi_g + \Delta_k \chi_B \chi_D)\chi_D^2]^2}, \frac{\lambda_k^2}{(\chi_g + \Delta_k \chi_B \chi_D)^2}\right\} \lambda_k \Delta_k^2. \end{aligned}$$

The proof is completed. □

When $\hat{M}_k$ is positive definite, we denote the Newton step for (2.10) by

$$s_k^N \stackrel{\text{def}}{=} -D_k^{-1} \hat{M}_k^{-1} \hat{g}_k, \quad \text{i.e.,} \quad \hat{M}_k D_k s_k^N = -\hat{g}_k. \tag{3.19}$$

LEMMA 3.9. *Assume* (AS.1), (AS.4), (AS.5), *and* (AS.6) *hold and* $f(x)$ *is twice continuously differentiable on* $\mathcal{L}$. *If the sequence of trust region subproblem* (2.10) *solutions* $\{p_k\}$ *converges to zero,* $\{x_k\}$ *converges to* x_*, *and* $\hat{M}_*$ *is positive definite, then*

$$\liminf_{k \to \infty} \frac{\psi_k(\alpha_k^*[p_k])}{\psi_k^*[p_k]} \geq 1, \quad \liminf_{k \to \infty} \frac{\psi_k^*[p_k]}{\psi_k(p_k)} \geq 1.$$

Moreover, for sufficiently large k,

$$|\psi_k^*[p_k]| \geq \epsilon \min\{\Delta_k^2, \|D_k s_k^N\|^2\}$$

for some constant $\epsilon > 0$.

Proof. Let α_k be the stepsize along p_k to the boundary as defined in (2.11):

$$\alpha_k = \min\left\{\max\left\{\frac{l_i - x_{ki}}{p_{ki}}, \frac{u_i - x_{ki}}{p_{ki}}\right\} : \ 1 \leq i \leq n\right\}.$$

Since p_k is a solution to (2.10), $\tau_k^* = \min\{1, \alpha_k\}$ in (2.12). By condition (2.13) on θ_k, $\theta_l \leq \theta_k < 1, \theta_k - 1 = O(\|p_k\|)$. Since $\hat{M}_k$ is positive definite for sufficiently large k, we have that $p_k^T M_k p_k > 0$. In addition, $\psi_k^*[p_k] < 0$. Therefore

$$\begin{aligned}
\liminf_{k\to\infty} \frac{\psi_k(\alpha_k^*[p_k])}{\psi_k^*[p_k]} &= \liminf_{k\to\infty} \frac{\tau_k^*\theta_k g_k^T p_k + \frac{1}{2}\tau_k^{*2}\theta_k^2 p_k^T M_k p_k}{\tau_k^* g_k^T p_k + \frac{1}{2}\tau_k^{*2} p_k^T M_k p_k} \\
&\geq \liminf_{k\to\infty} \frac{\tau_k^*\theta_k g_k^T p_k + \frac{1}{2}\tau_k^{*2}\theta_k p_k^T M_k p_k}{\tau_k^* g_k^T p_k + \frac{1}{2}\tau_k^{*2} p_k^T M_k p_k} \text{ (using } \theta_k^2 \leq \theta_k \text{ and } \psi_k^*[p_k] < 0) \\
&\geq \lim_{k\to\infty} \theta_k \\
&= 1.
\end{aligned}$$

The last equality $\lim_{k\to\infty} \theta_k = 1$ comes from $\lim_{k\to\infty} p_k = 0$ (by assumption).

Since $\hat{M}_*$ is positive definite, x_* is nondegenerate. If all variables are free at the limit point x_*, then from the assumption that $\{p_k\}$ converges to zero it is clear that

$$\liminf_{k\to\infty} \alpha_k = +\infty.$$

Assume now that there exist variables on the boundary at x_*. Recall (3.3):

$$(\lambda_k I + D_k^{-2} C_k) p_k = -D_k^{-2}(g_k + B_k p_k).$$

Since $\{B_k p_k\}$ converges to zero (note that $B_k = \nabla^2 f_k$ under the assumption (AS.5)), $\lambda_k I + D_k^{-2} C_k$ is a positive semidefinite diagonal matrix, and x_* is nondegenerate with $D_*^{-1} g_* = 0$ (because of Theorem 3.5), for any i with $v_{*i} = 0$, p_{ki} and g_{ki} have the same sign for k sufficiently large. Hence, if α_k is defined by some $v_{*j} = 0$ and $g_{*j} \neq 0$, then $\alpha_k = \frac{|v_{kj}|}{|p_{kj}|}$ for sufficiently large k. Using (3.3) again,

$$\alpha_k = \frac{|g_{kj}| + \lambda_k}{|g_{kj} + (B_k p_k)_j|}.$$

This means that

$$\liminf_{k\to\infty} \alpha_k \geq 1. \tag{3.20}$$

Using Lemma 3.7, we have

$$-\psi_k^*[p_k] \geq \frac{1}{2} \min\{1, \alpha_k\} \|R_k D_k p_k\|^2.$$

Assume that $\epsilon_0 > 0$ is a lower bound on the eigenvalues of $\hat{M}_*$. From (3.2), we have

$$\|R_k D_k p_k\|^2 \geq \epsilon_0 \|D_k p_k\|^2 + \lambda_k \|D_k p_k\|^2.$$

But $\|D_k p_k\| \leq \Delta_k$ and, for sufficiently large k, $p_k = s_k^N$ if $\|D_k p_k\| < \Delta_k$. Hence

$$|\psi_k^*[p_k]| \geq \frac{1}{2}\epsilon_0 \min\{1, \alpha_k\} \min\{\Delta_k^2, \|D_k s_k^N\|^2\},$$

where $D_k s_k^N = -\hat{M}_k^{-1}\hat{g}_k$. Let $0 < \epsilon < \frac{1}{2}\epsilon_0$. Then, using (3.20), for k sufficiently large,

$$|\psi_k^*[p_k]| \geq \epsilon \min\{\Delta_k^2, \|D_k s_k^N\|^2\}.$$

In addition, from $\tau_k^{*2} \leq \tau_k^*$ (since $0 \leq \tau_k^* = \min\{\alpha_k, 1\} \leq 1$), $p_k^T M_k p_k > 0$ for sufficiently large k, and $\psi_k^*(p_k) < 0$, we have

$$\begin{aligned}\liminf_{k\to\infty} \frac{\psi_k^*[p_k]}{\psi_k(p_k)} &= \liminf_{k\to\infty} \frac{\tau_k^* g_k^T p_k + \frac{1}{2}\tau_k^{*2} p_k^T M_k p_k}{g_k^T p_k + \frac{1}{2} p_k^T M_k p_k} \\ &\geq \liminf_{k\to\infty} \min\{\alpha_k, 1\} \\ &= 1.\end{aligned}$$

Hence

$$\liminf_{k\to\infty} \frac{\psi_k^*[p_k]}{\psi_k(p_k)} \geq 1. \qquad \square$$

The next theorem indicates that the first-order and second-order necessary conditions can be satisfied.

THEOREM 3.10. *Assume* (AS.1) *holds and* $f : \mathcal{F} \to \Re$ *is twice continuously differentiable on* $\mathcal{F}$. *Let* $\{x_k\}$ *be the sequence generated by Algorithm* 2 *under assumption* (AS.5) *on the model* ψ_k *and under assumptions* (AS.4) *and* (AS.6) *on the step* s_k. *Then*

(i) *the sequence* $\{\hat{g}_k\}$ *converges to zero*;

(ii) *if every limit point is nondegenerate, then there is a limit point* x_* *with* $\hat{M}_*$ *positive semidefinite*;

(iii) *if* x_* *is an isolated nondegenerate limit point, then* $\hat{M}_*$ *is positive semidefinite*;

(iv) *if* $\hat{M}_*$ *is nonsingular for some limit point* x_* *of* $\{x_k\}$, *then* $\hat{M}_*$ *is positive definite,* $\{x_k\}$ *converges to* x_*, *all iterations are eventually successful, and* $\{\Delta_k\}$ *is bounded away from zero.*

Under the additional assumption that problem (1.1) *is nondegenerate, equivalent results hold for the sequence generated by Algorithm* 1.

Proof. We prove each result in order.

(i) The sequence $\{\hat{g}_k = D_k^{-1} g_k\}$ converges to zero—this was proved in Theorem 3.5.

(ii) First we consider the case when $\liminf_{k\to\infty} \lambda_k = 0$. Let $\lambda_{\min}(M_k)$ denote the minimum eigenvalue of M_k. Since $\lambda_k \geq \max(-\lambda_{\min}(M_k), 0)$, it is clear that, when $\liminf_{k\to\infty} \lambda_k = 0$, there must exist a limit point x_* at which M_* is positive semidefinite.

Next we prove by contradiction that $\liminf_{k\to\infty} \lambda_k = 0$. Assume that $\lambda_k \geq \epsilon > 0$ for all k sufficiently large. First we show that $\{\Delta_k\}$ converges to zero.

From Lemma 3.8, we have that, for sufficiently large k,

$$-\psi_k(s_k) \geq \frac{\beta^q}{2}\hat{\epsilon}_k \epsilon \Delta_k^2,$$

where

$$\hat{\epsilon}_k \stackrel{\text{def}}{=} \min\left\{1, \frac{\epsilon^2\epsilon_0^2}{[(\chi_g + \Delta_k\chi_D\chi_B)\chi_D^2]^2}, \frac{\epsilon^2}{(\chi_g + \Delta_k\chi_B\chi_D)^2}\right\}.$$

Moreover, for sufficiently large k and successful iterations,

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2}\beta^q\mu\hat{\epsilon}_k\epsilon\Delta_k^2. \tag{3.21}$$

The remaining arguments are similar to the proof of Theorem 3.4. If there is a finite number of successful iterations, $\{\Delta_k\}$ converges to zero. Otherwise, let $\{k_i\}$ be the infinite sequence of successful iterations. The definition of $\hat{\epsilon}_k$, inequality (3.21), and the convergence of $\{f(x_k)-f(x_{k+1})\}$ to zero imply that there exists a constant $\epsilon_1 > 0$ such that $\hat{\epsilon}_{k_i} > \epsilon_1$. This fact and inequality (3.21) imply that

$$\sum_{i=1}^{\infty}\Delta_{k_i}^2 < \infty.$$

Similar to the updating rule in Figure 1.1 for unconstrained trust region methods, the updating rules of Algorithms 1 and 2 specify that $\Delta_{k+1} \leq \gamma_1\Delta_k$ for an unsuccessful iteration and $\Delta_{k+1} \leq \gamma_2\Delta_k$ for a successful iteration. It is easy to verify that the following holds:

$$\sum_{k=1}^{\infty}\Delta_k^2 \leq \left(1 + \frac{\gamma_2^2}{1-\gamma_1}\right)\sum_{i=1}^{\infty}\Delta_{k_i}^2.$$

Hence $\{\Delta_k\}$ converges to zero. Since $\|s_k\| = \|D_k^{-1}D_ks_k\| \leq \chi_D\beta_0^q\Delta_k$ and $\|p_k\| \leq \chi_D\Delta_k$, we conclude that both $\{s_k\}$ and $\{p_k\}$ converge to zero.

From the fact that $\{\Delta_k\}$ converges to zero,

$$\hat{\epsilon}_k \geq \hat{\epsilon} \qquad \text{for some } \hat{\epsilon} > 0.$$

Hence,

$$-\psi_k(s_k) \geq \frac{\beta^q}{2}\hat{\epsilon}\epsilon\Delta_k^2.$$

Now a standard estimate is that

$$\begin{aligned}&\left|f(x_k + s_k) - f(x_k) + \frac{1}{2}s_k^TC_ks_k - \psi_k(s_k)\right| \\ \leq &\|s_k\|^2 \max_{0\leq\xi\leq 1}\|\nabla^2 f(x_k + \xi s_k) - \nabla^2 f(x_k)\|,\end{aligned}$$

and thus the last two inequalities and the fact that $\{s_k\}$ converges to zero (thus $\{\|\nabla^2 f(x_k + \xi s_k) - \nabla^2 f(x_k)\|\}$ converges to zero) show that $\{|\rho_k^f - 1|\}$ converges to zero.

We conclude that the entire sequence $\{\rho_k^f\}$ converges to unity.

For Algorithm 2, using Lemma 3.3, $\{\Delta_k\}$ cannot converge to zero, which is a contradiction.

Now we consider Algorithm 1. Let α_k be the stepsize to the boundary along p_k. According to the definition (2.11),

$$\alpha_k = \min\left\{\max\left\{\frac{l_i - x_{ki}}{p_{ki}}, \frac{u_i - x_{ki}}{p_{ki}}\right\} : 1 \leq i \leq n\right\}.$$

If all variables are free at a limit point x_*, then it is clear that $\liminf_{k\to\infty} \alpha_k = +\infty$. Otherwise, consider a limit point x_* with some variables at their bounds. Since this limit point is nondegenerate, $\lim_{k\to\infty} D_k^{-2} g_k = 0$, and p_k converges to zero, $-g_{kj}$ and p_{kj} have the same sign for any $g_{*j} \neq 0$. Using (3.3), we have

$$\alpha_k = \frac{|v_{kj}|}{|p_{kj}|} = \frac{|g_{kj}| + \lambda_k}{|g_{kj} + (B_k p_k)_j|}$$

for sufficiently large k with $g_{*j} \neq 0$. But $\lambda_k > \epsilon$. Thus the corresponding limit of α_k is greater than 1. Hence

$$\liminf_{k\to\infty} \alpha_k > 1.$$

In other words, $s_k = \alpha_k^*[p_k] = p_k$ for sufficiently large k. Therefore $\rho_k^c \geq 1$ for sufficiently large k. Hence, $\{\Delta_k\}$ cannot converge to zero, which is a contradiction.

In conclusion, there is a limit point with $\hat{M}_*$ positive semidefinite.

(iii) If $\{x_k\}$ converges to x_*, the result follows from (ii). If $\{x_k\}$ does not converge then Lemma 3.6 applies. Thus, if $\{x_{l_j}\}$ is the subsequence guaranteed by Lemma 3.6, then $\Delta_{l_j} \geq \frac{1}{\chi_D \beta_0^q}\epsilon$ and the l_jth iteration is successful. From Lemma 3.8 (note that Lemma 3.8 holds for *any subsequence*),

$$f(x_{l_j}) - f(x_{l_j+1}) > \frac{\beta^q}{2}\mu \min\left\{1, \frac{\lambda_{l_j}^2 \epsilon_0^2}{[(\chi_g + \Delta_{l_j}\chi_B\chi_D)\chi_D^2]^2}, \frac{\lambda_{l_j}^2}{(\chi_g + \Delta_{l_j}\chi_B\chi_D)^2}\right\} \lambda_{l_j} \Delta_{l_j}^2.$$

From $\Delta_{l_j} \geq \frac{1}{\chi_D \beta_0^q}\epsilon$, it is straightforward to verify that there exist positive constants $\epsilon_1, \epsilon_2, \epsilon_3$ such that

$$f(x_{l_j}) - f(x_{l_j+1}) > \frac{\beta^q}{2}\mu \min\{\epsilon_1 \lambda_{l_j}, \epsilon_2 \lambda_{l_j}^3, \epsilon_3 \lambda_{l_j}^3\}.$$

Since $\{f(x_k)\}$ is monotonically nonincreasing and bounded below, $\{f(x_k) - f(x_{k+1})\}$ converges to zero. Hence $\{\lambda_{l_j}\}$ converges to zero. Thus $\hat{M}_*$ is positive semidefinite.

(iv) If $\hat{M}_*$ is nonsingular at a limit point x_* of $\{x_k\}$, then x_* is an isolated limit point and hence $\hat{M}_*$ is positive definite following (iii). Since $\psi_k(s_k) = g_k^T s_k + \frac{1}{2} s_k^T M_k s_k < 0$, we have that

$$\hat{g}_k^T D_k s_k < -\frac{1}{2}(D_k s_k)^T \hat{M}_k D_k s_k < 0$$

whenever $\hat{M}_k$ is positive definite. Hence

$$\frac{1}{2}(D_k s_k)^T \hat{M}_k D_k s_k < \|D_k s_k\| \|\hat{g}_k\|.$$

But $(D_k s_k)^T \hat{M}_k D_k s_k \geq \frac{1}{\|\hat{M}_k^{-1}\|}\|D_k s_k\|^2$. Therefore

$$\frac{1}{2}\|D_k s_k\| \leq \|\hat{M}_k^{-1}\| \|\hat{g}_k\|$$

whenever $\hat{M}_k$ is positive definite. This means that

$$\frac{1}{2}\|s_k\| \leq \frac{1}{2}\chi_D \|D_k s_k\| \leq \chi_D \|\hat{M}_k^{-1}\| \|\hat{g}_k\|$$

whenever $\hat{M}_k$ is positive definite. But $\{D_k^{-1}g_k\}$ converges to zero. Following Lemma 3.6, $\{x_k\}$ converges. Since $\lim_{k\to\infty}\hat{g}_k = 0$, $\{x_k\}$ converges to x_*, and $\hat{M}_*$ is positive definite, $\{p_k\}$ and $\{s_k\}$ converge to zero.

Next we prove that the iterations are eventually successful and $\{\Delta_k\}$ is bounded away from zero. Using Lemma 3.9, there exists $\epsilon > 0$ such that, for sufficiently large k,

$$|\psi_k^*[p_k]| \geq \epsilon \min\{\Delta_k^2, \|D_k s_k^N\|^2\}. \tag{3.22}$$

But recall that whenever $\hat{M}_k$ is positive definite

$$\frac{1}{2}\|D_k s_k\| \leq \|\hat{M}_k^{-1}\|\|\hat{g}_k\|.$$

Let κ be an upper bound on the condition number of $\hat{M}_k$. From $\hat{g}_k = -\hat{M}_k D_k s_k^N$ based on (3.19),

$$\frac{1}{2}\|D_k s_k\| \leq \kappa\|D_k s_k^N\|.$$

Hence, using (3.22) and (AS.6), there exists $\bar{\epsilon} > 0$ such that

$$-\psi_k(s_k) \geq \bar{\epsilon}\|D_k s_k\|^2 \geq \frac{\bar{\epsilon}}{\chi_D^2}\|s_k\|^2.$$

This estimate and

$$\begin{aligned}&\left|f(x_k + s_k) - f(x_k) + \frac{1}{2}s_k^T C_k s_k - \psi_k(s_k)\right| \\ \leq &\|s_k\|^2 \max_{0\leq\xi\leq 1}\|\nabla^2 f(x_k + \xi s_k) - \nabla^2 f(x_k)\|\end{aligned}$$

yield that $\rho_k^f > \eta$ for k sufficiently large since $\{s_k\}$ converges to zero. (Thus $\{\|\nabla^2 f(x_k + \xi s_k) - \nabla^2 f(x_k)\|\}$ converges to zero.)

For Algorithm 2, using Lemma 3.3, we immediately conclude that $\{\Delta_k\}$ is bounded away from zero.

Now we consider Algorithm 1. Since p_k is a solution of the trust region subproblem, it is clear that

$$\psi_k(p_k) \leq \min\{\psi_k(\tau D_k^{-2}g_k) : \|\tau D_k^{-1}g_k\| \leq \Delta_k\} \leq \psi_k^*[-D_k^{-2}g_k] \leq 0.$$

Using Lemma 3.9, for k sufficiently large,

$$\rho_k^c = \frac{\psi_k(p_k)}{\psi_k^*[-D_k^{-2}g_k]} \times \frac{\psi_k^*[p_k]}{\psi_k(p_k)} \times \frac{\psi_k(\alpha_k^*[p_k])}{\psi_k^*[p_k]} > \eta.$$

Therefore all the iterations are eventually successful. According to the trust region size updating rules, $\{\Delta_k\}$ is bounded away from zero. The results are established. □

The next result establishes quadratic convergence of the sequence generated by Algorithm 2, provided the truncated Newton step is chosen when it is admissible. (Note that Theorem 13 in [7] establishes that ultimately the truncated Newton step will always be admissible.)

THEOREM 3.11. *Assume the conditions of Theorem* 3.10 *hold,* $0 < \beta < 1$, *and* $\hat{M}_*$ *is nonsingular for some limit point* x_* *of* $\{x_k\}$. *Let* s_k^N *be the Newton step* (3.19) *when it exists. Further, assume* $s_k = \alpha_k^*[s_k^N]$ *whenever* $\|D_k s_k^N\| < \Delta_k$ *and* $\alpha_k^*[s_k^N]$ *satisfies* (AS.4). *Then* $\{x_k\}$ *converges to* x_* *quadratically.*

Proof. From Theorem 3.10, $\{\Delta_k\}$ is bounded away from zero. But under our assumptions, $\{D_k^{-1} g_k\}$ converges to zero and $\{x_k\}$ converge to x_*. Hence $\{D_k s_k^N\}$ converges to zero where s_k^N is the Newton step:

$$\hat{M}_k D_k s_k^N = -\hat{g}_k.$$

Next we prove that $s_k = \alpha_k^*[s_k^N]$ for sufficiently large k. It suffices to prove that, for sufficiently large k, $\|D_k \alpha_k^*[s_k^N]\| < \Delta_k$ and $\alpha_k^*[s_k^N]$ satisfies (AS.4). Since $\{D_k s_k^N\}$ converges to zero, $\|D_k \alpha_k^*[s_k^N]\| < \Delta_k$ for sufficiently large k. Using definition (2.14), for sufficiently large k,

$$\alpha_k^*[s_k^N] - s_k^N = \tau_k^* \theta_k s_k^N - \tau_k^* s_k^N + \tau_k^* s_k^N - s_k^N,$$

where $\tau_k^* = \min\{1, \alpha_k\}$. From Lemma 11 in [7], $|\tau_k^* - 1| = O(\|x_k - x_*\|)$. But $|\theta_k - 1| = O(\|s_k^N\|)$. Since $0 < \beta < 1$ by assumption, for sufficiently large k, $\beta < \theta_k \tau_k^* \leq 1$ and $\|\alpha_k^*[s_k^N] - s_k^N\| = O(\|x_k - x_*\|^2)$. This implies that $0 \leq \theta_k^2 \tau_k^{*2} \leq \theta_k \tau_k^*$. Using (2.14), for k sufficiently large,

$$\psi_k(\alpha_k^*[s_k^N]) \leq \theta_k \tau_k^* \left(g_k^T s_k^N + \frac{1}{2} s_k^{N^T} M_k s_k^N \right) < \beta \psi_k^*[-D_k^{-2} g_k],$$

i.e., $\alpha_k^*[s_k^N]$ satisfies (AS.4) for sufficiently large k. Hence, under the assumptions, $s_k = \alpha_k^*[s_k^N]$ for k sufficiently large. In addition, following (iv) in Theorem 3.10, $\alpha_k^*[s_k^N]$ yields a successful iteration for sufficiently large k, i.e.,

$$f(x_k + \alpha_k^*[s_k^N]) - f(x_k) < \mu \left(g_k^T s_k^N + \frac{1}{2} s_k^{N^T} M_k s_k^N \right).$$

Using Theorem 13 in [7], $\{x_k\}$ converges quadratically to x_*. □

4. Preliminary numerical experiments. In this section we report on preliminary experiments with the practical trust region method, Algorithm 2, on a set of standard test problems of low dimension. The method solves these problems quite satisfactorily, indicating that this approach has practical potential.

We implemented the "practical trust region algorithm" described in Figure 2.2 in a straightforward manner. Either $s_k = \alpha_k^*[p_k]$, where p_k is a solution to the trust region subproblem (2.10) or $s_k = \alpha_k^*[-D_k^{-2} g_k]$. The exact implementation is described in Figure 4.1.

The computed step s_k satisfies the conditions

$$\psi_k(s_k) < \beta \psi_k(\alpha_k^*[-D_k^{-2} g_k]), \ \|D_k s_k\| \leq \Delta_k, \ x_k + s_k \in \text{int}(\mathcal{F})$$

and

$$\psi_k(s_k) < \psi_k(\alpha_k^*[p_k]), \ \|D_k s_k\| \leq \Delta_k, \ x_k + s_k \in \text{int}(\mathcal{F}).$$

Note that it is easy to verify that condition (AS.4) can be replaced with

$$\psi_k(s_k) < \beta \psi_k(\alpha_k^*[-D_k^{-2} g_k]), \ \|D_k s_k\| \leq \beta_0 \Delta_k, \ x_k + s_k \in \text{int}(\mathcal{F})$$

The method implemented

Let $\mu = 0.25, \beta = 0.1, \eta = 0.75$, $\gamma_0 = 0.0625, \gamma_1 = 0.5, \gamma_2 = 2$, $\Lambda_l = 1$, and $x_0 \in \text{int}(\mathcal{F})$

For $k = 0, 1, \ldots$

1. Compute $f(x_k)$, g_k, $\nabla^2 f(x_k)$, and C_k; define the quadratic model $\psi_k(s_k) = g_k^T s_k + \frac{1}{2} s_k^T (\nabla^2 f(x_k) + C_k) s_k$.
2. Compute a solution p_k of (2.10). Compute

$$\rho_k^c = \frac{\psi_k(\alpha_k^*[p_k])}{\psi_k(\alpha_k^*[-D_k^{-2} g_k])}.$$

   If $\rho_k^c > \beta$, $s_k = \alpha_k^*[p_k]$. Otherwise, $s_k = \alpha_k^*[-D_k^{-2} g_k]$.
3. Compute

$$\rho_k^f = \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k}{\psi_k(s_k)}.$$

4. If $\rho_k^f > \mu$ then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
5. Update the scaling matrix D_k and Δ_k as specified.

Updating trust region size Δ_k

1. If $\rho_k^f < 0$ then $\Delta_{k+1} = \gamma_0 \Delta_k$.
2. If $0 \le \rho_k^f \le \mu$ then $\Delta_{k+1} = \max\{\gamma_0 \Delta_k, \gamma_1 \|D_k s_k\|\}$.
3. If $\rho_k^f \ge \eta$ then
   if $\rho_k^c > \eta$ then
   $\Delta_{k+1} = \max\{\Delta_k, \gamma_2 \|D_k s_k\|\}$
   else
   if $\Delta_k > \Lambda_l$ and $\rho_k^c \le \mu$ then
   $\Delta_{k+1} = \max\{\gamma_1 \Delta_k, \|D_k s_k\|\}$.
4. Otherwise, $\Delta_{k+1} = \Delta_k$.

FIG. 4.1. *The interior and trust region method implemented.*

and condition (AS.6) can be replaced with

$$\psi_k(s_k) < \beta^q \psi_k(\alpha_k^*[p_k]), \ \|D_k s_k\| \le \beta_0^q \Delta_k, \ x_k + s_k \in \text{int}(\mathcal{F}).$$

Thus, the implemented method has the convergence properties listed in Theorems 3.10 and 3.11.

The experiments were done on a Sun (Sparc) workstation using Matlab 4.0 [16]. The stopping criteria used were

$$\hat{M}_k > 0 \text{ and } \psi_k(s_k) < 0.5 * 10^{-12}.$$

The test problems are taken from [9]. However, the starting points as described in [9] may not be strictly feasible. Assume that x_{start} is the starting point specified in [9]. We modify the starting points as follows:

$$x_{0i} = l_i + 0.1 * (u_i - l_i) \text{ if } x_{start\,i} < l_i + 100\epsilon,$$
$$x_{0i} = u_i - 0.1 * (u_i - l_i) \text{ if } x_{start\,i} > u_i - 100\epsilon,$$

TABLE 4.1
Experiments with a practical trust region method for bounded constrained problems.

Number of Function/Gradient Evaluations				
PROB	n	NEW		CGT
GENROSE U	8	45	38	42
GENROSE C	8	15	14	15
CHAINROSE U	25	23	19	20
CHAINROSE C	25	28	23	18
DEGENROSE U	25	30	30	95
DEGENROSE C	25	28	23	17
GENSING U	20	27	27	10
GENSING C	20	23	22	4
CHAINSING U	20	26	26	18
CHAINSING C	20	22	21	3
DEGENSING U	20	26	26	155
DEGENSING C	20	35	34	3
GENWOOD U	8	68	57	107
GENWOOD C	8	10	9	5
CHAINWOOD U	8	57	48	77
CHAINWOOD C	8	10	9	5
HOSC45 U	10	28	27	19
HOSC45 C	10	10	9	12
BROYDEN1A U	30	14	14	11
BROYDEN1A C	30	14	13	8
BROYDEN1B U	30	7	7	7
BROYDEN1B C	30	9	8	6
BROYDEN2A U	30	19	16	14
BROYDEN2A C	30	24	22	10
BROYDEN2B U	30	9	9	9
BROYDEN2B C	30	15	14	9
TOINTBROY U	30	8	8	8
TOINTBROY C	30	12	11	8
TRIG U	10	13	11	7
TRIG C	10	16	14	8
TOINTTRIG U	10	8	6	13
TOINTTRIG C	10	8	6	10
CRAGGLEVY U	8	33	31	24
CRAGGLEVY C	8	30	29	20
PENALTY U	15	24	24	27
PENALTY C	15	29	29	80
AUGMLAGN U	15	29	26	31
AUGMLAGN C	15	46	44	31
BROWN1 U	10	20	20	27
BROWN1 C	10	31	30	27
BROWN3 U	10	9	9	7
BROWN3 C	10	10	9	6
BVP U	10	21	21	4
BVP C	10	20	20	4
VAR U	20	12	12	6
VAR C	20	12	11	6

where $\epsilon \approx 10^{-16}$ is the machine precision.

In Table 4.1, we report the number of function and gradient evaluations taken by the method to obtain the required accuracy. The number of function evaluations required by the method in [9] is given in the last column as a relative comparison (the exact Hessian is used to obtain these numbers). Our implemented method computes

the trust region solutions while the method in [9] uses conjugate gradient. We point out that the starting points and the stopping criteria of the two methods are different. The approximate solutions obtained by the two methods may also differ.

A subspace version of this interior trust region method is currently under investigation for large-scale problems. Preliminary results are very encouraging and will be reported subsequently.

5. Conclusions. We have proposed a trust region approach to the bound-constrained nonlinear minimization problem. This approach generates strictly feasible iterates and possesses strong convergence characteristics. In particular, we have established second-order convergence properties. Moreover, the convergence results match the implementation in the sense that a global solution to a quadratic programming problem, with linear inequality constraints, is not required by the theory. Instead, an approximate minimization of a quadratic function subject to an ellipsoidal constraint is required (and achievable).

Our computational experiments on a well-known test collection of small-dimensional problems indicate that Algorithm 2 has practical potential. However, from a practical computational point of view we believe the real promise of the underlying ideas presented here is in the large-scale setting. The method as described is not directly suitable for large-scale problems—the computation of a (suitably) accurate solution to the trust region problem in high dimensions is probably too costly. Nevertheless, there is considerable scope for modifying and adapting the basic idea, with efficiency in mind, to the large-scale setting. This line of research is currently under investigation.

Finally, we remark that the trust region ideas developed in this paper for box constraints can be extended to the case where there are also linear equality constraints present, i.e., $\min\{f(x) : Ax = b, l \leq x \leq u\}$. This generalization is also the subject of current research.

REFERENCES

[1] A. Bjorck, *A direct method for sparse least squares problems with lower and upper bounds*, Numer. Math., 54 (1988), pp. 19–32.

[2] R. H. Byrd, R. B. Schnabel, and G. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[3] T. F. Coleman and C. Hempel, *Computing a trust region step for a penalty function*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 180–201.

[4] T. F. Coleman and L. A. Hulbert, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Math. Programming, 45 (1989), pp. 373–406.

[5] ———, *A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds*, SIAM J. Optim., 3 (1993), pp. 298–321.

[6] T. F. Coleman and Y. Li, *A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on the Variables*, Tech. report TR 92-1315, Computer Science Department, Cornell University, Ithaca, NY, 1992.

[7] ———, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds*, Math. Programming, 67 (1994), pp. 189–224.

[8] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[9] ———, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[10] R. Fletcher, *An algorithm for solving linearly constrained optimization problems*, Math. Programming, 2 (1972), pp. 133–165.

[11] R. FLETCHER AND M. P. JACKSON, *Minimization of a quadratic function of many variables subject only to lower and upper bounds*, J. Inst. Math. Appl., 14 (1974), pp. 159–174.

[12] D. M. GAY, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[13] P. GILL AND W. MURRAY, *Minimization Subject to Bounds on the Variables*, Tech. report NAC 71, National Physical Laboratory, England, 1976.

[14] J. J. JÚDICE AND F. M. PIRES, *Direct methods for convex quadratic programs subject to box constraints*, Investigacao Peracional, 9 (1989), pp. 23–56.

[15] P. LOTSTEDT, *Solving the minimal least squares problem subject to bounds on the variables*, BIT, 24 (1984), pp. 206–224.

[16] C. B. MOLER, J. LITTLE, S. BANGERT, AND S. KLEIMAN, *ProMatlab User's guide*, MathWorks, Sherborn, MA, 1987.

[17] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art, M. G. A. Bachem and E. B. Dorte, eds., Springer-Verlag, Berlin, 1983.

[18] J. J. MORÉ AND D. SORENSEN, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[19] J. J. MORÉ AND G. TORALDO, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[20] D. P. O'LEARY, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[21] U. ÖREBORN, *A Direct Method for Sparse Nonnegative Least Squares Problems*, Ph.D. thesis, Department of Mathematics, Linköping University, Linköping, Sweden, 1986.

[22] G. A. SCHULTZ, R. B. SCHNABEL, AND R. H. BYRD, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[23] D. SORENSEN, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[24] T. STEIHAUG, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[25] E. K. YANG AND J. W. TOLLE, *A class of methods for solving large convex quadratic programs subject to box constraints*, Math. Programming, 51 (1991), pp. 223–228.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 446–460, May 1996
© 1996 Society for Industrial and Applied Mathematics
010

EQUIVALENCE OF COMPLEMENTARITY PROBLEMS TO DIFFERENTIABLE MINIMIZATION: A UNIFIED APPROACH*

PAUL TSENG†, NOBUO YAMASHITA‡, AND MASAO FUKUSHIMA‡

Abstract. We consider two merit functions for a generalized nonlinear complementarity problem (GNCP) based on quadratic regularization of the standard linearized gap function. The first extends Fukushima's merit function for variational inequality problems [Fukushima, *Math. Programming*, 53 (1992), pp. 99–110] and the second extends Mangasarian and Solodov's implicit Lagrangian for complementarity problems [Mangasarian and Solodov, *Math. Programming*, 62 (1993), pp. 277–297]. We show, among other things, that the second merit function is in the order of the natural residual squared and we give conditions under which the stationary points of this function are the solutions to GNCP. These results extend those of Luo et al. [*Math. Oper. Res.*, 19 (1994), pp. 880–892] and of Yamashita and Fukushima [*J. Optim. Theory Appl.*, 84 (1995), pp. 653–663] on the properties of the implicit Lagrangian.

Key words. complementarity problem, merit function, error bound, natural residual, quadratic regularization

AMS subject classifications. 65K10, 90C33

1. Introduction.

There have recently been a number of studies on casting a nonlinear complementarity problem (NCP) and a variational inequality problem equivalently as the minimization of a "nice" function over a simple set. Notable among these is a work by Fukushima [Fuk92], which cast the variational inequality problem equivalently as the minimization of a regularized gap function over the feasible set and designed feasible descent methods for solving this minimization problem. Closely related is an independent work by Auchmuty [Auc89], whose function $\mathcal{G}$ (with f a chosen quadratic) coincides with the regularized gap function when both are specialized to NCP. Mangasarian and Solodov [MaS93] cast NCP equivalently as the unconstrained minimization of an implicit Lagrangian, which resembles a primal-dual version of the regularized gap function. Subsequently, Luo et al. [LMRS94] showed that the implicit Lagrangian is in the order of the natural residual squared (so the square root of the implicit Lagrangian provides an error bound for NCP if and only if the natural residual does). Yamashita and Fukushima [YaF95] showed that, in the case in which the function in NCP is strongly monotone, the implicit Lagrangian has compact level sets and a unique stationary point that is the solution to NCP. Both the regularized gap function and the implicit Lagrangian are differentiable if the function in the problem is differentiable and, under additional assumptions, their minimum may be computed using second-order methods (see [Kan96], [MaS93], [TFI93]).

The preceding studies raise a number of intriguing questions. What is the connection between the regularized gap function and the implicit Lagrangian? Is there a unified approach to generating these and other functions with similar features? Does the connection between the implicit Lagrangian and the natural residual extend beyond NCP? Does the connection between the stationary points of the implicit La-

* Received by the editors February 18, 1994; accepted for publication (in revised form) October 12, 1994. The research of the first author was supported by National Science Foundation grant CCR-9103804. The research of the third author was supported by the Scientific Research Grant-in-Aid from the Ministry of Education, Science, and Culture, Japan.

† Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

‡ Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara 630-01, Japan.

grangian and the solutions of the problem extend beyond NCP? In this paper, we answer these questions in the affirmative. And our argument is surprisingly simple, using only basic facts about projection onto a closed convex cone and its polar.

We take as our problem setting the following generalized nonlinear complementarity problem (GNCP):

(GNCP)

$$\text{Find an } x \in \Re^m \text{ satisfying} \quad G(x) \in \mathcal{K}, \quad H(x) \in \mathcal{K}^\circ, \quad \langle G(x), H(x) \rangle = 0,$$

where $\mathcal{K}$ is a nonempty closed convex cone in $\Re^n$, $\mathcal{K}^\circ$ denotes the polar of $\mathcal{K}$, and G and H are continuous functions from $\Re^m$ to $\Re^n$. Here and throughout, we denote by $\langle \cdot, \cdot \rangle$ the Euclidean inner product on $\Re^n$ and by $\|\cdot\|$ the norm induced by this inner product. (Much of our analysis also extends to the Hilbert space setting, but for simplicity of exposition we will discuss only the Euclidean space setting.) In the special case where $m = n$, $\mathcal{K} = \Re^n_+$, $G(x) = x - E(x)$, and $H(x) = -F(x)$ for all x, with E and F some functions from $\Re^m$ to $\Re^n$, GNCP reduces to the so-called implicit complementarity problem [DoM80], [Pan81] (also called the quasi-complementarity problem [Noo88]). In the further special case where $E(x) = 0$ for all x, GNCP reduces to NCP:

$$\text{(NCP)} \qquad \text{Find an } x \in \Re^m \text{ satisfying} \quad x \geq 0, \quad F(x) \geq 0, \quad \langle x, F(x) \rangle = 0.$$

(See [Pan95] for a recent survey of NCP.)

Our goal is to cast GNCP equivalently as the minimization of a merit function over a simple set. (We say a real-valued function on $\Re^m$ is a merit function for GNCP over some subset of $\Re^m$ if the global minima of this function over the subset coincide with the solutions to GNCP.) Moreover, we like this merit function to be differentiable whenever G and H are. Certainly GNCP is equivalent to the minimization of the so-called "gap" function

$$p(G(x), H(x))$$

over all $x \in \Re^m$ satisfying $G(x) \in \mathcal{K}$ and $H(x) \in \mathcal{K}^\circ$, where $p : \Re^n \times \Re^n \mapsto \Re$ is given by

$$p(u, v) = -\langle u, v \rangle,$$

which is differentiable and nonnegative for all $(u, v) \in \mathcal{K} \times \mathcal{K}^\circ$ and equals 0 if and only if in addition $\langle u, v \rangle = 0$. However, this minimization problem suffers the serious drawback that its feasible set is very complicated. We can circumvent this difficulty by borrowing an idea from (partial) proximal minimization, namely, we can release a subset of the constraints from the minimization problem (and thus simplify the latter's feasible set) by adding to p a quadratic regularization term associated with these constraints. In particular, we can release the constraints $H(x) \in \mathcal{K}^\circ$ by adding a quadratic regularization term on v to p or, more precisely, GNCP is equivalent to the minimization of the function $\hat{m}_\alpha : \Re^m \mapsto \Re$ given by

$$\hat{m}_\alpha(x) = \hat{\pi}_\alpha(G(x), H(x))$$

over all $x \in \Re^m$ satisfying $G(x) \in \mathcal{K}$, where $\hat{\pi}_\alpha : \Re^n \times \Re^n \mapsto \Re$ is given by

$$\begin{aligned} \hat{\pi}_\alpha(u, v) &= \min_{\zeta \in \mathcal{K}^\circ} \left\{ p(u, \zeta) + \frac{1}{2\alpha} \|v - \zeta\|^2 \right\} \\ &= -\langle u, [v + \alpha u]^+_{\mathcal{K}^\circ} \rangle + \frac{1}{2\alpha} \|v - [v + \alpha u]^+_{\mathcal{K}^\circ}\|^2 \end{aligned}$$

and $\alpha \in (0, \infty)$. (Throughout, for any nonempty closed convex set C in $\Re^n$, we denote by $[\cdot]_C^+$ the orthogonal projection onto C.) It can be seen that $\hat{\pi}_\alpha(u, v)$ is differentiable and nonnegative for all $(u, v) \in \mathcal{K} \times \Re^n$ and equals 0 if and only if in addition $v \in \mathcal{K}^\circ$ and $\langle u, v \rangle = 0$. Thus, $\hat{\pi}_\alpha$ may be viewed as a smooth exact penalty function for the constraints $v \in \mathcal{K}^\circ$. (We can alternatively release the constraints $G(x) \in \mathcal{K}$ by adding a quadratic regularization term on u to p. The choice depends on whether G or H is more complex.) Finally, we can release all the constraints by adding a quadratic regularization term on both u and v to p or, more precisely, GNCP is equivalent to the minimization of

$$\tilde{p}_\alpha(G(x), H(x))$$

over all $x \in \Re^m$, where $\tilde{p}_\alpha : \Re^n \times \Re^n \mapsto \Re$ is given by

$$\tilde{p}_\alpha(u, v) = \min_{\substack{\psi \in \mathcal{K} \\ \zeta \in \mathcal{K}^\circ}} \left\{ p(\psi, \zeta) + \frac{1}{2\alpha}(\|u - \psi\|^2 + \|v - \zeta\|^2) \right\}$$

and $\alpha \in (0, \infty)$. It can be seen that $\tilde{p}_\alpha(u, v)$ is differentiable and nonnegative for all $(u, v) \in \Re^n \times \Re^n$ and equals 0 if and only if in addition $(u, v) \in \mathcal{K} \times \mathcal{K}^\circ$ and $\langle u, v \rangle = 0$. Thus, $\tilde{p}_\alpha$ may be viewed as a smooth exact penalty function for the constraints $(u, v) \in \mathcal{K} \times \mathcal{K}^\circ$. Although $\tilde{p}_\alpha$ is useful because it is differentiable, it suffers the drawback of being too difficult to compute. One way to simplify the computation is to replace $p(\psi, \zeta)$ in the definition of $\tilde{p}_\alpha(u, v)$ by the linear approximation $p(u, v) + \langle \nabla_u p(u, v), \psi - u \rangle + \langle \nabla_v p(u, v), \zeta - v \rangle$. This yields the function $\tilde{m}_\alpha : \Re^m \mapsto \Re$ given by

$$\tilde{m}_\alpha(x) = \tilde{\pi}_\alpha(G(x), H(x)),$$

where $\tilde{\pi}_\alpha : \Re^n \times \Re^n \mapsto \Re$ is given by

$$\begin{aligned}
\tilde{\pi}_\alpha(u, v) &= \min_{\substack{\psi \in \mathcal{K} \\ \zeta \in \mathcal{K}^\circ}} \left\{ p(u, v) - \langle v, \psi - u \rangle - \langle u, \zeta - v \rangle + \frac{1}{2\alpha}(\|u - \psi\|^2 + \|v - \zeta\|^2) \right\} \\
&= \langle u, v \rangle - \langle v, [u + \alpha v]_{\mathcal{K}}^+ \rangle - \langle u, [v + \alpha u]_{\mathcal{K}^\circ}^+ \rangle \\
&\quad + \frac{1}{2\alpha}(\|u - [u + \alpha v]_{\mathcal{K}}^+\|^2 + \|v - [v + \alpha u]_{\mathcal{K}^\circ}^+\|^2)
\end{aligned}$$

and $\alpha \in (0, \infty)$. The two functions $\hat{m}_\alpha$ and $\tilde{m}_\alpha$ will be the focus of our study.

Our main contributions are as follows. First, we show that, for any $\alpha \in (0, \infty)$, $\hat{m}_\alpha$ is a merit function for GNCP on $G^{-1}(\mathcal{K})$, is at least in the order of the natural residual squared on $G^{-1}(\mathcal{K})$, and is differentiable whenever G and H are (see Theorem 3.1). Similarly, we show that, for any $\alpha \in (0, 1)$ (respectively, $\alpha \in (1, \infty)$), $\tilde{m}_\alpha$ (respectively, $-\tilde{m}_\alpha$) is a merit function for GNCP on $\Re^m$, is in the order of the natural residual squared on $\Re^m$, and is differentiable whenever G and H are (see Theorem 3.2). Moreover, in the case in which $G(x) = x$ for all x, the function $\hat{m}_\alpha$ reduces to the regularized gap function and, in the case of NCP, the function $-\tilde{m}_\alpha$ reduces to the implicit Lagrangian. The preceding results on $\tilde{m}_\alpha$ extend [MaS93, Thm. 2.1] and [LMRS94, Thms. 3.1, 3.2] and, in particular, show that the implicit Lagrangian acts as a merit function for NCP on $\Re^m$ not only for $\alpha > 1$ but also for $\alpha < 1$. We also obtain, as a byproduct, an extension of [LMRS94, Thm. 5.1].

Second, we derive conditions on G and H under which the stationary points of $\tilde{m}_\alpha$ coincide with the solutions to GNCP (see Theorems 4.1 and 4.2). These results

extend [YaF95, Thm. 2.2], which considers only NCP and further assumes $\alpha > 1$. We also extend [MaS93, Thm. 2.3] to show that, in the case $\mathcal{K} = \Re^n_+$, the global minima (respectively, global maxima) of $\tilde{m}_\alpha$, for $\alpha \in (0,1)$ (respectively, $\alpha \in (1,\infty)$), are isolated under a certain linear independence assumption on the columns of ∇G and ∇H (see Theorem 5.1).

Finally, we consider the case $\mathcal{K} = \Re^n_+$ and compare $\tilde{m}_\alpha$ with the implicit Lagrangian applied to a reformulation of GNCP as an NCP.

We now state a few words about notation. We denote by $\Re^n$ the n-dimensional real space and by $\Re^n_+$ the nonnegative orthant in $\Re^n$. For any function $F : \Re^m \mapsto \Re^n$, we denote by F_j the jth component of F, $j = 1, \ldots, n$, and by ∇F the $m \times n$ matrix whose jth column is the gradient function ∇F_j, $j = 1, \ldots, n$. For any real matrix A, we denote by A^* its transpose.

2. Properties of $\hat{\pi}_\alpha$ and of $\tilde{\pi}_\alpha$. It is well known and easily shown that a $(u,v) \in \Re^n \times \Re^n$ satisfies $(u,v) \in \mathcal{K} \times \mathcal{K}^\circ$ and $\langle u, v\rangle = 0$ if and only if they satisfy the fixed point equations $u = [u+v]^+_{\mathcal{K}}$ or, equivalently (by using the fact that $u+v = [u+v]^+_{\mathcal{K}} + [u+v]^+_{\mathcal{K}^\circ}$), $v = [u+v]^+_{\mathcal{K}^\circ}$. Thus, the quantity

$$\|u - [u+v]^+_{\mathcal{K}}\| \tag{1}$$

measures how close (u,v) comes to satisfying the equations. In this section we analyze the differentiability of the functions $\hat{\pi}_\alpha$ and $\tilde{\pi}_\alpha$ and express their growth rate in terms of the quantity (1). Our analysis will frequently use the following fact about projection (see [GaB84, Lem. 1]):

$$\begin{array}{rcccl} \min\{1,\alpha\}\|u - [u+v]^+_{\mathcal{K}}\| & \le & \|u - [u+\alpha v]^+_{\mathcal{K}}\| & \le & \max\{1,\alpha\}\|u - [u+v]^+_{\mathcal{K}}\|, \\ \min\{1,\alpha\}\|u - [u+v]^+_{\mathcal{K}}\| & \le & \|v - [v+\alpha u]^+_{\mathcal{K}^\circ}\| & \le & \max\{1,\alpha\}\|u - [u+v]^+_{\mathcal{K}}\| \end{array} \tag{2}$$

for all $(u,v) \in \Re^n \times \Re^n$ and all $\alpha \in (0,\infty)$.

The following proposition states the key properties of $\hat{\pi}_\alpha$, showing that $\hat{\pi}_\alpha$ is differentiable and $\hat{\pi}_\alpha(u,v)$ is at least in the order of the quantity (1) squared on $\mathcal{K} \times \Re^n$.

PROPOSITION 2.1. *For any $\alpha \in (0,\infty)$, the function $\hat{\pi}_\alpha$ is differentiable everywhere, with*

$$\nabla \hat{\pi}_\alpha(u,v) = \begin{bmatrix} -[v+\alpha u]^+_{\mathcal{K}^\circ} \\ \dfrac{1}{\alpha}(v - [v+\alpha u]^+_{\mathcal{K}^\circ}) \end{bmatrix},$$

and satisfies

$$\hat{\pi}_\alpha(u,v) \ge \frac{\min\{1,\alpha\}^2}{2\alpha}\|u - [u+v]^+_{\mathcal{K}}\|^2 \tag{3}$$

for all $(u,v) \in \mathcal{K} \times \Re^n$. Moreover, $\hat{\pi}_\alpha(u,v)$ is nonnegative for all $(u,v) \in \mathcal{K} \times \Re^n$ and equals 0 if and only if in addition $v \in \mathcal{K}^\circ$ and $\langle u, v\rangle = 0$.

Proof. That $\hat{\pi}_\alpha$ is differentiable everywhere and its gradient has the given form follow from a result of Danskin (see [Aus76, Ch. 4, Thm. 1.7] or [BeT89, p. 649]). Since $u \in \mathcal{K}$, we have $\langle u, [v+\alpha u]^+_{\mathcal{K}^\circ}\rangle \le 0$ and the formula for $\hat{\pi}_\alpha$ yields

$$\hat{\pi}_\alpha(u,v) \ge \frac{1}{2\alpha}\|v - [v+\alpha u]^+_{\mathcal{K}^\circ}\|^2.$$

Using (2) to bound below the right-hand side yields (3). The remaining results readily follow from (3). □

On the other hand, except in the case in which $\mathcal{K}$ is a subspace (so $\mathcal{K}$ and $\mathcal{K}^\circ$ are orthogonal complements), the function $\hat{\pi}_\alpha(u,v)$ always grows faster than the residual (1) squared (or raised to any power) on $\mathcal{K} \times \Re^n$. To see this, assume that $\mathcal{K}$ is not a subspace, so there is some point z in the span of $\mathcal{K}$ (i.e., the smallest subspace containing $\mathcal{K}$) that is not in $\mathcal{K} \cup \mathcal{K}^\circ$. Let $w = z - [z]^+_{\mathcal{K}}$ (so w is an outer normal vector to $\mathcal{K}$ at $[z]^+_{\mathcal{K}}$ and, since $\mathcal{K}$ is a convex cone, $\langle [z]^+_{\mathcal{K}}, w\rangle = 0$) and let

$$u = [z]^+_{\mathcal{K}} - \epsilon w,$$

where $\epsilon \in (0,\infty)$ is chosen sufficiently small so that u is in $\mathcal{K}$. (Such an ϵ exists by our choice of z.) Since w is an outer normal vector to $\mathcal{K}$ at $[z]^+_{\mathcal{K}}$, we have that for any $\lambda \in (0,\infty)$, the point

$$v(\lambda) = (\lambda + \epsilon)w$$

satisfies $[u+v(\lambda)]^+_{\mathcal{K}} = [z]^+_{\mathcal{K}}$ and $[u+v(\lambda)]^+_{\mathcal{K}^\circ} = \lambda w$. The former yields $\|[u+v(\lambda)]^+_{\mathcal{K}} - u\| = \|[z]^+_{\mathcal{K}} - u\| = \epsilon\|w\|$ and the latter yields $-\langle u, [u+v(\lambda)]^+_{\mathcal{K}^\circ}\rangle = -\langle [z]^+_{\mathcal{K}} - \epsilon w, \lambda w\rangle = \epsilon\lambda\|w\|^2$, which, as $\lambda \to \infty$, grows faster than any power of $\epsilon\|w\|$.

We also note that by using the following properties of projection onto a convex cone and its polar:

$$\left\langle \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}, [v+\alpha u]^+_{\mathcal{K}^\circ} \right\rangle = 0, \quad [v+\alpha u]^+_{\mathcal{K}^\circ} = v+\alpha u - [v+\alpha u]^+_{\mathcal{K}} = v+\alpha\left(u - \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}\right),$$

we may rewrite $\hat{\pi}_\alpha(u,v)$ as

$$\begin{aligned}
\hat{\pi}_\alpha(u,v) &= -\left\langle u - \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}, v + \alpha\left(u - \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}\right)\right\rangle + \frac{\alpha}{2}\left\|u - \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}\right\|^2 \\
&= -\left\langle u - \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}, v\right\rangle - \frac{\alpha}{2}\left\|u - \left[\frac{v}{\alpha} + u\right]^+_{\mathcal{K}}\right\|^2 \\
&= \max_{\psi\in\mathcal{K}}\left\{-\langle u-\psi, v\rangle - \frac{\alpha}{2}\|u-\psi\|^2\right\}. \qquad (4)
\end{aligned}$$

The alternative formula (4) for $\hat{\pi}_\alpha(u,v)$ will be used in the proof of Proposition 2.2 and in relating $\hat{m}_\alpha$ to the regularized gap function.

The following proposition states the key properties of $\tilde{\pi}_\alpha$, showing that $\tilde{\pi}_\alpha$ is differentiable, has nice symmetry with respect to α, and $\tilde{\pi}_\alpha(u,v)$ is in the order of the quantity (1) squared on $\Re^n \times \Re^n$.

PROPOSITION 2.2. *For any $\alpha \in (0,\infty)$, the function $\tilde{\pi}_\alpha$ is differentiable everywhere, with*

$$\nabla\tilde{\pi}_\alpha(u,v) = \begin{bmatrix} v - [v+\alpha u]^+_{\mathcal{K}^\circ} + \dfrac{1}{\alpha}(u - [u+\alpha v]^+_{\mathcal{K}}) \\ u - [u+\alpha v]^+_{\mathcal{K}} + \dfrac{1}{\alpha}(v - [v+\alpha u]^+_{\mathcal{K}^\circ}) \end{bmatrix},$$

and satisfies

$$(5)\quad (1-\alpha)\|u - [u+v]^+_{\mathcal{K}}\|^2 \le \tilde{\pi}_\alpha(u,v) = -\tilde{\pi}_{1/\alpha}(u,v) \le \left(\frac{1}{\alpha-1}\right)\|u - [u+v]^+_{\mathcal{K}}\|^2$$

for all $(u,v) \in \Re^n \times \Re^n$. *Moreover, for* $\alpha \in (0,1)$ *(respectively,* $\alpha \in (1,\infty)$*), we have that* $\tilde{\pi}_\alpha(u,v)$ *is nonnegative (respectively, nonpositive) for all* $(u,v) \in \Re^n \times \Re^n$ *and equals* 0 *if and only if* $(u,v) \in \mathcal{K} \times \mathcal{K}^\circ$ *and* $\langle u,v\rangle = 0$. *For* $\alpha = 1$, *we have* $\tilde{\pi}_\alpha \equiv 0$.

Proof. That $\tilde{\pi}_\alpha$ is differentiable everywhere and its gradient has the given form follow from a result of Danskin (see [Aus76, Ch. 4, Thm. 1.7] or [BeT89, p. 649]). To prove (5), consider any $(u,v) \in \Re^n \times \Re^n$. We have from (4) that

$$-\langle u, [v+\alpha u]^+_{\mathcal{K}^\circ}\rangle + \frac{1}{2\alpha}\|v - [v+\alpha u]^+_{\mathcal{K}^\circ}\|^2 = \max_{\psi\in\mathcal{K}}\left\{-\langle u-\psi, v\rangle - \frac{\alpha}{2}\|u-\psi\|^2\right\}$$

and analogously (by switching the roles of u and v and of $\mathcal{K}$ and $\mathcal{K}^\circ$ in the above equation) that

$$-\langle v, [u+\alpha v]^+_{\mathcal{K}}\rangle + \frac{1}{2\alpha}\|u - [u+\alpha v]^+_{\mathcal{K}}\|^2 = \max_{\zeta\in\mathcal{K}^\circ}\left\{-\langle v-\zeta, u\rangle - \frac{\alpha}{2}\|v-\zeta\|^2\right\},$$

which when used in the formula for $\tilde{\pi}_\alpha(u,v)$ yields

$$\begin{aligned}\tilde{\pi}_\alpha(u,v) &= \langle u,v\rangle + \max_{\psi\in\mathcal{K}}\left\{-\langle u-\psi,v\rangle - \frac{\alpha}{2}\|u-\psi\|^2\right\} + \max_{\zeta\in\mathcal{K}^\circ}\left\{-\langle v-\zeta,u\rangle - \frac{\alpha}{2}\|v-\zeta\|^2\right\}\\ &= -p(u,v) - \min_{\psi\in\mathcal{K}}\left\{\langle u-\psi,v\rangle + \frac{\alpha}{2}\|u-\psi\|^2\right\} - \min_{\zeta\in\mathcal{K}^\circ}\left\{\langle v-\zeta,u\rangle + \frac{\alpha}{2}\|v-\zeta\|^2\right\}\\ &= -\tilde{\pi}_{1/\alpha}(u,v),\end{aligned}$$

and the equality in (5) is proven. Next, we note that $\tilde{\pi}_\alpha(u,v)$ equals the minimum of the following quantity:

$$\begin{aligned}&p(u,v) - \langle\psi-u,v\rangle - \langle u,\zeta-v\rangle + \frac{1}{2\alpha}(\|u-\psi\|^2 + \|v-\zeta\|^2)\\ &\quad= \langle u,v\rangle - \langle\psi,v\rangle - \langle u,\zeta\rangle + \frac{1}{2\alpha}(\|u-\psi\|^2 + \|v-\zeta\|^2)\\ &\quad= \langle\psi-u,\zeta-v\rangle - \langle\psi,\zeta\rangle + \frac{1}{2\alpha}(\|u-\psi\|^2 + \|v-\zeta\|^2)\\ &\quad= \frac{1}{2}\|\psi-u+\zeta-v\|^2 - \langle\psi,\zeta\rangle + \frac{1}{2}\left(\frac{1}{\alpha}-1\right)(\|u-\psi\|^2 + \|v-\zeta\|^2) \qquad (6)\end{aligned}$$

over all (ψ,ζ) in $\mathcal{K}\times\mathcal{K}^\circ$, so $\tilde{\pi}_\alpha(u,v)$ is less than or equal to the value of this quantity at $\psi = [u+v]^+_{\mathcal{K}}$ and $\zeta = [u+v]^+_{\mathcal{K}^\circ}$. With this choice of ψ and ζ, the first two terms of (6) equal zero and $u-\psi = \zeta - v$, so the quantity (6) has a value equal to the right-hand side of (5) and the second inequality in (5) follows. To prove the first inequality in (5), we note that the second inequality in (5) still holds when α is replaced by $1/\alpha$, so that

$$\tilde{\pi}_\alpha(u,v) = -\tilde{\pi}_{1/\alpha}(u,v) \geq -(\alpha-1)\|u - [u+v]^+_{\mathcal{K}}\|^2.$$

The remaining results readily follow from (5). □

An interesting question concerns whether $\tilde{\pi}_\alpha$ and Proposition 2.2 extend to other choices of the function p. This seems impossible for the equality or the second inequality in (5) but is possible for the first inequality under a very general setting. In particular, let $\mathcal{C}$ be any closed convex set in $\Re^l$ for some l and let p be any continuously differentiable function on $\Re^l$ that is nonnegative on $\mathcal{C}$ and has at most quadratic growth, i.e.,

$$p(\zeta) \leq p(z) + \langle\nabla p(z), \zeta - z\rangle + \frac{1}{2a}\|\zeta - z\|^2 \quad \forall z,\ \zeta \in \Re^l,$$

for some scalar $a > 0$. Then it is readily seen that $\tilde{\pi}_\alpha$ defined by

$$\tilde{\pi}_\alpha(z) = \min_{\zeta \in \mathcal{C}} \left\{ p(z) + \langle \nabla p(z), \zeta - z \rangle + \frac{1}{2\alpha} \|\zeta - z\|^2 \right\}$$

satisfies

$$\tilde{\pi}_\alpha(z) \geq \frac{1}{2} \left(\frac{1}{\alpha} - \frac{1}{a} \right) \|z - [z - \alpha \nabla p(z)]_{\mathcal{C}}^+\|^2$$

for all $z \in \Re^l$. Thus, the first inequality in (5) depends on nothing more than the fact that the bilinear function $(u, v) \mapsto -\langle u, v \rangle$ is nonnegative on $\mathcal{K} \times \mathcal{K}^\circ$ and has at most quadratic growth.

3. $\hat{m}_\alpha$ and $\tilde{m}_\alpha$: Merit functions and error bounds for GNCP. In this section, we study how $\hat{m}_\alpha$ and $\tilde{m}_\alpha$ act as merit functions and provide error bounds for GNCP on, respectively, $G^{-1}(\mathcal{K})$ and $\Re^m$. We will also use the relationship between $\hat{\pi}_\alpha$ and $\tilde{\pi}_\alpha$ and the quantity (1) (see Propositions 2.1 and 2.2) to relate $\hat{m}_\alpha$ and $\tilde{m}_\alpha$ to the norm of the natural residual functions $R_\alpha : \Re^m \mapsto \Re^n$ and $S_\alpha : \Re^m \mapsto \Re^n$ given by

$$R_\alpha(x) = G(x) - [G(x) + \alpha H(x)]_{\mathcal{K}}^+, \qquad S_\alpha(x) = H(x) - [H(x) + \alpha G(x)]_{\mathcal{K}^\circ}^+$$

for all $x \in \Re^m$, with $\alpha \in (0, \infty)$. Note that an $x \in \Re^m$ is a solution to GNCP if and only if $R_\alpha(x) = S_\alpha(x) = 0$, so $\|R_\alpha(x)\|$ (or $\|S_\alpha(x)\|$) raised to any positive power is a merit function for GNCP on $\Re^m$. However, such a merit function has the serious drawback that it is not differentiable even when G and H are.

As a consequence of Proposition 2.1, we have the first main result showing that $\hat{m}_\alpha$ is a merit function for GNCP on $G^{-1}(\mathcal{K})$, is at least in the order of $\|R_1\|$ squared, and is differentiable whenever G and H are.

THEOREM 3.1. *For any $\alpha \in (0, \infty)$, the function $\hat{m}_\alpha$ satisfies*

$$\hat{m}_\alpha(x) \geq \frac{\min\{1, \alpha\}^2}{2\alpha} \|R_1(x)\|^2$$

for all $x \in \Re^m$ with $G(x) \in \mathcal{K}$. Moreover, $\hat{m}_\alpha(x)$ is nonnegative for all $x \in \Re^m$ with $G(x) \in \mathcal{K}$ and equals 0 if and only if x is a solution to GNCP. If in addition G and H are differentiable, then so is $\hat{m}_\alpha$ and

$$\nabla \hat{m}_\alpha(x) = \nabla G(x) \left(S_\alpha(x) - H(x) \right) + \frac{1}{\alpha} \nabla H(x) S_\alpha(x)$$

for all $x \in \Re^m$.

In the special case in which $m = n$, $G(x) = x$ for all x and $H \equiv -F$ for some $F : \Re^m \mapsto \Re^n$, it is readily seen from the alternative formula (4) for $\hat{\pi}_\alpha$ that the function $\hat{m}_\alpha$ reduces to Fukushima's regularized gap function f defined in [Fuk92] (assuming, without loss of generality, that $G = I$ therein). On the other hand, the latter function is defined not only for convex cones but also more generally for convex sets.

As a consequence of Proposition 2.2, we have the second main result showing that $\tilde{m}_\alpha$ has nice symmetry with respect to α, is a merit function for GNCP on $\Re^m$, is in the order of $\|R_1\|$ squared, and is differentiable whenever G and H are.

THEOREM 3.2. *For any $\alpha \in (0,\infty)$, the function $\tilde{m}_\alpha$ satisfies*

$$(1-\alpha)\|R_1(x)\|^2 \le \tilde{m}_\alpha(x) = -\tilde{m}_{1/\alpha}(x) \le \left(\frac{1}{\alpha-1}\right)\|R_1(x)\|^2$$

for all $x \in \Re^m$. Moreover, for $\alpha \in (0,1)$ (respectively, $\alpha \in (1,\infty)$), we have that $\tilde{m}_\alpha(x)$ is nonnegative (respectively, nonpositive) for all $x \in \Re^m$ and equals 0 if and only if x is a solution to GNCP. If in addition G and H are differentiable, then so is $\tilde{m}_\alpha$ and

$$\nabla\tilde{m}_\alpha(x) = \nabla G(x)\left(S_\alpha(x) + \frac{1}{\alpha}R_\alpha(x)\right) + \nabla H(x)\left(R_\alpha(x) + \frac{1}{\alpha}S_\alpha(x)\right)$$

for all $x \in \Re^m$.

We claim that, in the special case of NCP (corresponding to GNCP with $m = n$, $\mathcal{K} = \Re^n_+$, $G(x) = x$ for all x, and $H \equiv -F$ for some $F : \Re^m \mapsto \Re^n$), the function $-\tilde{m}_\alpha$ reduces to Mangasarian and Solodov's implicit Lagrangian [MaS93]. This follows from the observation that, when $\mathcal{K} = \Re^n_+$, the function $\tilde{\pi}_\alpha$ simplifies to

$$\tilde{\pi}_\alpha(u,v) = \langle u,v\rangle + \frac{1}{2\alpha}\left(\|u\|^2 - \|[u+\alpha v]_+\|^2 + \|v\|^2 - \|[v+\alpha u]_+\|^2\right),$$

where $[\cdot]_+$ denotes the positive part of a vector. It was shown in [LMRS94] (see Theorem 3.1 therein) that the implicit Lagrangian with $\alpha \in (1,\infty)$ is between $(1 - 1/\alpha)\|x - [x - F(x)]_+\|^2$ and $(\alpha - 1)\|x - [x - F(x)]_+\|^2$ for all $x \in \Re^m$. By using Theorem 3.2, we conclude that this is still true with $\alpha \in (0,1]$.

It follows from Theorem 3.1 that, for any $\alpha \in (0,\infty)$, $\sqrt{\hat{m}_\alpha}$ provides an error bound for GNCP on $G^{-1}(\mathcal{K})$ (i.e., the Euclidean distance from any $x \in \Re^m$ satisfying $G(x) \in \mathcal{K}$ to the solution set of GNCP is bounded above by some constant times $\sqrt{\hat{m}_\alpha(x)}$) if $\|R_1\|$ does. Similarly, it follows from Theorem 3.2 that, for any $\alpha \in (0,\infty)$ with $\alpha \ne 1$, $\sqrt{|\tilde{m}_\alpha|}$ provides an error bound for GNCP on $\Re^m$ if and only if $\|R_1\|$ does. Moreover, $\hat{m}_\alpha$ has bounded level sets on $G^{-1}(\mathcal{K})$ if the solution set of GNCP is bounded and $\|R_1\|$ provides a global error bound for GNCP on $G^{-1}(\mathcal{K})$; $|\tilde{m}_\alpha|$ has bounded level sets on $\Re^m$ if the solution set of GNCP is bounded and $\|R_1\|$ provides a global error bound for GNCP on $\Re^m$. In the case $G(x) = x$ for all x, necessary and sufficient conditions for $\|R_1\|$ to provide a (local or global) error bound for GNCP have been much studied and are discussed in [LMRS94], [LuT92], [LuT93], [Man90], [Pan87], [MaP90]. For example, it is known (see [Pan87, Thm. 3.1]) that the solution set of GNCP is bounded (in fact, unique) and $\|R_1\|$ provides a global error bound for GNCP on $\Re^m$ if $G(x) = x$ for all x and $-H$ is strongly monotone and Lipschitz continuous on $\Re^m$. It follows that, in this case, $\hat{m}_\alpha$ has bounded level sets on $G^{-1}(\mathcal{K})$ and $|\tilde{m}_\alpha|$ has bounded level sets on $\Re^m$. The latter result improves [YaF95, Thm. 2.4], which considers only the case $\mathcal{K} = \Re^n_+$ and further assumes α is sufficiently large.

Although $\tilde{m}_\alpha$ is more complicated than $\hat{m}_\alpha$, $\tilde{m}_\alpha$ has the advantage that it works on all of $\Re^m$, not only on $G^{-1}(\mathcal{K})$, and that it grows no faster than $\|R_1\|$ squared on $\Re^m$ (whereas $\hat{m}_\alpha$ can grow faster than $\|R_1\|$ raised to any power on $G^{-1}(\mathcal{K})$, as is suggested by the discussion following Proposition 2.1). Finally, it can be seen from the formula for $\nabla\tilde{m}_\alpha$ given in Theorem 3.2 that, in the case in which ∇G and ∇H are bounded on $\Re^m$, the norm of $\nabla\tilde{m}_\alpha$ is bounded above by a constant times $\|R_1\|$ on $\Re^m$. Thus, $\tilde{m}_\alpha$ is additionally useful because its gradient does not grow too fast in norm.

Finally, it was shown in [LMRS94, Thm. 5.1] that, in the case of NCP, the square root of the partial derivative of 2α times the implicit Lagrangian (with respect to $\alpha \in (1, \infty)$) is also in the order of the natural residual squared. We can extend this result to $\hat{m}_\alpha$ and $\tilde{m}_\alpha$ by using Danskin's result to compute the partial derivative of $\hat{m}_\alpha$ and $\tilde{m}_\alpha$ with respect to $\alpha \in (0, \infty)$, yielding

$$\frac{\partial}{\partial \alpha}\hat{m}_\alpha(x) = -\frac{1}{2\alpha^2}\|S_\alpha(x)\|^2,$$
$$\frac{\partial}{\partial \alpha}\tilde{m}_\alpha(x) = -\frac{1}{2\alpha^2}(\|R_\alpha(x)\|^2 + \|S_\alpha(x)\|^2)$$

for all $x \in \Re^m$ and all $\alpha \in (0, \infty)$. Using (2), it readily follows that $\sqrt{-\frac{\partial}{\partial \alpha}\hat{m}_\alpha}$ and $\sqrt{-\frac{\partial}{\partial \alpha}\tilde{m}_\alpha}$ are in the order of $\|R_1\|^2$ on $\Re^m$. By using Theorem 3.2 and the simple fact that

$$\frac{\partial}{\partial \alpha}[-2\alpha\tilde{m}_\alpha(x)] = -2\tilde{m}_\alpha(x) - 2\alpha\frac{\partial}{\partial \alpha}\tilde{m}_\alpha(x)$$

for all $x \in \Re^m$, we can further conclude that $\sqrt{\frac{\partial}{\partial \alpha}[-2\alpha\tilde{m}_\alpha]}$, with $\alpha \in (1, \infty)$, is in the order of $\|R_1\|^2$ on $\Re^m$. This result extends [LMRS94, Thm. 5.1].

4. Equivalence of stationary points of $\tilde{m}_\alpha$ to solutions to GNCP. It was shown in [Fuk92] that the regularized gap function has the useful property that, when the function in the variational inequality problem is strongly monotone, all its stationary points are solutions to the variational inequality problem. It was shown in [YaF95] that the implicit Lagrangian has the analogous property that, when the function F in NCP is strongly monotone, all its stationary points are solutions to NCP. Our third main result below extends [YaF95, Thm. 2.2] by giving conditions on G and H under which the stationary points of $\tilde{m}_\alpha$ are the solutions to GNCP. (It is desirable that the stationary points of $\tilde{m}_\alpha$ be solutions to GNCP since many methods for unconstrained minimization find only stationary points.)

THEOREM 4.1. *Assume G and H are differentiable. Then every solution to GNCP is a stationary point of $\tilde{m}_\alpha$ for $\alpha \in (0, \infty)$. Assume in addition that $m = n$ and that G and H satisfy the following condition.*

A1: *For every $x \in \Re^m$, the Jacobian $\nabla G(x)$ is nonsingular and there exists a real $n \times n$ matrix $\Lambda(x)$ such that $-\Lambda(x)[\nabla G(x)]^{-1}\nabla H(x)$ is positive definite and $\Lambda(x)$ has a block diagonal structure matching the Cartesian product structure of $\mathcal{K}$, with every block being some positive multiple of the identity matrix (see (9)).*

Then every stationary point of $\tilde{m}_\alpha$, for $\alpha \in (0, \infty)$ and $\alpha \neq 1$, is a solution to GNCP.

Proof. Since G and H are differentiable, by Theorem 3.2, $\tilde{m}_\alpha$ is differentiable and an $x \in \Re^m$ is a stationary point of $\tilde{m}_\alpha$ if and only if

$$0 = \nabla G(x)\left(S_\alpha(x) + \frac{1}{\alpha}R_\alpha(x)\right) + \nabla H(x)\left(R_\alpha(x) + \frac{1}{\alpha}S_\alpha(x)\right). \tag{7}$$

Thus, if x is a solution to GNCP so that $R_\alpha(x) = S_\alpha(x) = 0$, then x clearly satisfies (7) and hence is a stationary point of $\tilde{m}_\alpha$. Now assume in addition that $m = n$ and G and H satisfy condition A1. If x satisfies (7), taking the inner product of both sides of (7) with $(\Lambda(x)[\nabla G(x)]^{-1})^*(R_\alpha(x) + \frac{1}{\alpha}S_\alpha(x))$ yields (for brevity, we leave out the

"x" argument in what follows)

$$0 = \left\langle R_\alpha + \frac{1}{\alpha}S_\alpha, \Lambda\left(S_\alpha + \frac{1}{\alpha}R_\alpha\right)\right\rangle + \left\langle R_\alpha + \frac{1}{\alpha}S_\alpha, \Lambda[\nabla G]^{-1}\nabla H\left(R_\alpha + \frac{1}{\alpha}S_\alpha\right)\right\rangle. \tag{8}$$

To begin, we show that the first right-hand term in (8) is nonpositive. We have that $G+\alpha H = [G+\alpha H]^+_{\mathcal{K}} + [G+\alpha H]^+_{\mathcal{K}^\circ}$ and $H+\alpha G = [H+\alpha G]^+_{\mathcal{K}} + [H+\alpha G]^+_{\mathcal{K}^\circ}$. We also have, by the assumption on Λ, that (by rearranging the coordinates if necessary)

$$\Lambda = \begin{bmatrix} \lambda_1 I & & \\ & \ddots & \\ & & \lambda_l I \end{bmatrix}, \qquad \mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_l \tag{9}$$

for some positive integer l, some positive scalars $\lambda_1, \ldots, \lambda_l$, and some closed convex cones $\mathcal{K}_1, \ldots, \mathcal{K}_l$. Using these two facts, we see that

$$\begin{aligned}
&\left\langle R_\alpha + \frac{1}{\alpha}S_\alpha, \Lambda\left(S_\alpha + \frac{1}{\alpha}R_\alpha\right)\right\rangle \\
&= \left\langle -[G+\alpha H]^+_{\mathcal{K}} + \frac{1}{\alpha}[H+\alpha G]^+_{\mathcal{K}}, \Lambda\left(-[H+\alpha G]^+_{\mathcal{K}^\circ} + \frac{1}{\alpha}[G+\alpha H]^+_{\mathcal{K}^\circ}\right)\right\rangle \\
&= \sum_{j=1}^{l} \lambda_j \left[\left\langle -[G_j+\alpha H_j]^+_{\mathcal{K}_j} + \frac{1}{\alpha}[H_j+\alpha G_j]^+_{\mathcal{K}_j}, -[H_j+\alpha G_j]^+_{(\mathcal{K}_j)^\circ} + \frac{1}{\alpha}[G_j+\alpha H_j]^+_{(\mathcal{K}_j)^\circ}\right\rangle\right] \\
&= \sum_{j=1}^{l} \lambda_j \left[\langle [G_j+\alpha H_j]^+_{\mathcal{K}_j}, [H_j+\alpha G_j]^+_{(\mathcal{K}_j)^\circ}\rangle + \frac{1}{\alpha^2}\langle [H_j+\alpha G_j]^+_{\mathcal{K}_j}, [G_j+\alpha H_j]^+_{(\mathcal{K}_j)^\circ}\rangle\right] \\
&\leq 0,
\end{aligned}$$

where the second equality follows from (9) with $G_1, \ldots, G_l$ and $H_1, \ldots, H_l$ denoting the subvectors of G and H, respectively, compatible with the Cartesian product structure of $\mathcal{K}$. The third equality and the inequality follow from properties of projection onto a convex cone and its polar.

Second, we have from the positive definiteness assumption on $-\Lambda[\nabla G]^{-1}\nabla H$ that the second right-hand term in (8) is less than or equal to some negative scalar times $\|R_\alpha + \frac{1}{\alpha}S_\alpha\|^2$. Thus (8) implies $R_\alpha + \frac{1}{\alpha}S_\alpha = 0$, which together with (7) and the nonsingularity of ∇G yields $S_\alpha + \frac{1}{\alpha}R_\alpha = 0$. These two equations, together with the assumption $\alpha \neq 1$, show that $R_\alpha = S_\alpha = 0$. □

Condition A1, very roughly speaking, requires that one of ∇G and ∇H be positive definite and the other be negative definite. Note that condition A1 is unchanged in nature when the terms ∇G and ∇H are interchanged in its statement. Also, the assumption that $m = n$ is not essential and can be removed by replacing "$\nabla G(x)$" and "$\nabla H(x)$" in condition A1 with, respectively, "$\nabla G(x)^*\nabla G(x)$" and "$\nabla G(x)^*\nabla H(x)$." Our fourth main result below relates condition A1 to some alternative conditions.

THEOREM 4.2. *Assume G and H are differentiable and $m = n$. Condition* A1 *is equivalent to the following condition.*

A2: *For every $x \in \Re^m$, the Jacobians $\nabla G(x)$ and $\nabla H(x)$ are nonsingular and there exist real $n \times n$ matrices $U(x)$ and $V(x)$ such that both $U(x)\nabla G(x)$ and $V(x)\nabla H(x)$ are positive semidefinite, with at least one of them being positive definite, and $V(x)\nabla G(x) + \nabla H(x)^*U(x)^*$ has a block diagonal structure matching the Cartesian product structure of $\mathcal{K}$, with every block being some negative multiple of the identity matrix.*

Condition A1 *is also implied by any of the following three conditions.*

B1: $\mathcal{K} = \Re^n_+$ *and, for every* $x \in \Re^m$, *the Jacobians* $\nabla G(x)$ *and* $-\nabla H(x)$ *are positive definite and there exist scalars* $\psi(x) \geq 0$ *and* $\zeta(x) \leq 0$ *such that* $\zeta(x)\nabla G(x) + \psi(x)\nabla H(x)^*$ *is negative definite diagonal.*

B2: *For every* $x \in \Re^m$, *the Jacobian* $\nabla G(x)$ *is nonsingular and* $-[\nabla G(x)]^{-1}\nabla H(x)$ *is positive definite.*

B3: *For every* $x \in \Re^m$, *the Jacobians* $\nabla G(x)$ *and* $-\nabla H(x)$ *are positive definite symmetric and commutative.*

Proof. If A1 holds, then clearly A2 holds with $U(x) = 0$ and $V(x) = -\Lambda(x)[\nabla G(x)]^{-1}$. If A2 holds, we claim that A1 holds with

$$\Lambda(x) = -V(x)\nabla G(x) - \nabla H(x)^* U(x)^*.$$

Certainly $\Lambda(x)$ has a block diagonal structure matching the Cartesian product structure of $\mathcal{K}$, with every block being some positive multiple of the identity matrix. Thus, it suffices to show that $-\Lambda(x)[\nabla G(x)]^{-1}\nabla H(x)$ is positive definite. To show this, we multiply both sides of the preceding equation, on the left by $-[\nabla H(x)^*]^{-1}$ and on the right by $[\nabla G(x)]^{-1}$, yielding

$$-[\nabla H(x)^*]^{-1}\Lambda(x)[\nabla G(x)]^{-1} = [\nabla H(x)^*]^{-1}V(x) + U(x)^*[\nabla G(x)]^{-1}.$$

For any two square matrices A and B with B nonsingular, we have that AB is positive semidefinite (respectively, positive definite) if and only if $[B^*]^{-1}A$ is positive semidefinite (respectively, positive definite). Thus, the two matrices on the right-hand side are both positive semidefinite with at least one of them being positive definite. This implies that the matrix on the left-hand side is positive definite and, by the above fact again, the matrix $-\Lambda(x)[\nabla G(x)]^{-1}\nabla H(x)$ is positive definite.

If B1 holds, clearly A2 holds with $U(x) = \psi(x)I$ and $V(x) = \zeta(x)I$.

If B2 holds, clearly A1 holds with $\Lambda(x) = I$.

If B3 holds, we claim that B2 holds (and hence A1 holds). Since $[\nabla G(x)]^{-1}$ is positive definite symmetric, we can show this by proving that the product of two positive definite symmetric and commutative matrices A and B is positive definite. Since B is positive definite symmetric, it may be expressed as $B = B^{\frac{1}{2}}B^{\frac{1}{2}}$ for some positive definite symmetric matrix $B^{\frac{1}{2}}$. Since AB is symmetric (because A and B are symmetric and commutative) and the eigenvalues of AB are the same as those of $B^{\frac{1}{2}}AB^{\frac{1}{2}}$, it suffices to show that $B^{\frac{1}{2}}AB^{\frac{1}{2}}$ is positive definite. To show this, note that for any $u \in \Re^n$ such that $u \neq 0$, the symmetry of $B^{\frac{1}{2}}$ implies that

$$\langle u, B^{\frac{1}{2}}AB^{\frac{1}{2}}u\rangle = \langle B^{\frac{1}{2}}u, AB^{\frac{1}{2}}u\rangle > 0,$$

where the strict inequality follows from A being positive definite and $B^{\frac{1}{2}}$ being nonsingular. □

It follows from Theorems 4.1 and 4.2 that if G and H satisfy any of the conditions A1, A2, B1, B2, or B3, then GNCP is equivalent to finding a stationary point of $\tilde{m}_\alpha$ for any $\alpha \in (0, \infty)$ with $\alpha \neq 1$. This result extends [YaF95, Thm. 2.2], which considers the special case of NCP and further assumes that $\alpha > 1$.

We note that condition B3 cannot be replaced by the weaker condition that $\nabla G(x)$ and $-\nabla H(x)$ are positive definite and either $\nabla G(x)$ or $\nabla H(x)$ is symmetric. To see this, consider the following example with $m = n = 2$, where

$$\mathcal{K} = \Re^2_+, \qquad G(x_1, x_2) = \begin{bmatrix} 2x_1 - x_2 + \frac{1}{3} \\ -x_1 + x_2 \end{bmatrix}, \qquad H(x_1, x_2) = \begin{bmatrix} -x_1 - 2x_2 \\ 2x_1 - x_2 - \frac{1}{3} \end{bmatrix}.$$

It can be seen that, for every $x \in \Re^2$, $\nabla G(x)$ and $-\nabla H(x)$ are positive definite and $\nabla G(x)$ is symmetric. However, we have $\nabla \tilde{m}_2(x) = 0$ when $x = (1,2)$, but $(1,2)$ is not a solution to GNCP.

5. Isolated global minima/maxima of $\tilde{m}_\alpha$ when $\mathcal{K} = \Re^n_+$. Our next result states that, in the case $m = n$ and $\mathcal{K} = \Re^n_+$, the global minima (respectively, global maxima) of $\tilde{m}_\alpha$, for $\alpha \in (0,1)$ (respectively, $\alpha \in (1,\infty)$), are isolated under a certain linear independence assumption on the columns of ∇G and ∇H. This result is a fairly direct extension of [MaS93, Thm. 2.3] for the case in which $G(x) = x$ for all x and $\alpha \in (1,\infty)$. Because the proof is short, we have included it for completeness. Isolated global minima/maxima are of interest since many methods for unconstrained minimization/maximization attain fast local convergence only when the minima/maxima are isolated.

THEOREM 5.1. *Assume $m = n$ and $\mathcal{K} = \Re^n_+$. If $\bar{x}$ is a solution to GNCP such that G and H are twice differentiable at $\bar{x}$ and satisfy the strict complementarity condition*

$$\begin{aligned} G_j(\bar{x}) &= 0 \quad \forall j \in J, & G_j(\bar{x}) &> 0 \quad \forall j \in K, \\ H_j(\bar{x}) &< 0 \quad \forall j \in J, & H_j(\bar{x}) &= 0 \quad \forall j \in K \end{aligned}$$

for some partition J and K of $\{1,\ldots,n\}$ (i.e., $J \cap K = \emptyset$ and $J \cup K = \{1,\ldots,n\}$), and if the gradient vectors $\nabla G_j(\bar{x})$, $j \in J$ and $\nabla H_k(\bar{x})$, $k \in K$ are linearly independent, then $\bar{x}$ is an isolated global minimum (respectively, global maximum) of $\tilde{m}_\alpha$ for any $\alpha \in (0,1)$ (respectively, $\alpha \in (1,\infty)$).

Proof. To prove the result, it suffices to show that $\nabla^2 \tilde{m}_\alpha(\bar{x})$ is positive definite for $\alpha \in (0,1)$ and is negative definite for $\alpha \in (1,\infty)$. By using Theorem 3.2 and the fact that $R_\alpha(\bar{x}) = S_\alpha(\bar{x}) = 0$, we have

$$\alpha \nabla^2 \tilde{m}_\alpha(\bar{x}) = (\alpha \nabla G(\bar{x}) + \nabla H(\bar{x})) \nabla S_\alpha(\bar{x})^* + (\alpha \nabla H(\bar{x}) + \nabla G(\bar{x})) \nabla R_\alpha(\bar{x})^*.$$

We also have from the strict complementarity condition (and the rearrangement of coordinates if necessary) that

$$\begin{aligned} \nabla R_\alpha(\bar{x}) &= \nabla G(\bar{x}) - \left[\begin{array}{cc} 0_J & \nabla G_K(\bar{x}) + \alpha \nabla H_K(\bar{x}) \end{array} \right], \\ \nabla S_\alpha(\bar{x}) &= \nabla H(\bar{x}) - \left[\begin{array}{cc} \nabla H_J(\bar{x}) + \alpha \nabla G_J(\bar{x}) & 0_K \end{array} \right], \end{aligned}$$

where subscript J (respectively, K) denotes the columns whose indexes are in J (respectively, K). Thus,

$$\begin{aligned} \alpha \nabla^2 \tilde{m}_\alpha(\bar{x}) &= \alpha \nabla G(\bar{x}) \nabla H(\bar{x})^* + \nabla H(\bar{x}) \nabla H(\bar{x})^* \\ &\quad - (\alpha \nabla G_J(\bar{x}) + \nabla H_J(\bar{x}))(\alpha \nabla G_J(\bar{x})^* + \nabla H_J(\bar{x})^*) \\ &\quad + \alpha \nabla H(\bar{x}) \nabla G(\bar{x})^* + \nabla G(\bar{x}) \nabla G(\bar{x})^* \\ &\quad - (\alpha \nabla H_K(\bar{x}) + \nabla G_K(\bar{x}))(\alpha \nabla H_K(\bar{x})^* + \nabla G_K(\bar{x})^*) \\ &= (1 - \alpha^2)(\nabla G_J(\bar{x}) \; \nabla H_K(\bar{x})) \left(\begin{array}{c} \nabla G_J(\bar{x})^* \\ \nabla H_K(\bar{x})^* \end{array} \right). \end{aligned}$$

Since $m = n$ and the columns of $\nabla G_J(\bar{x})$ and $\nabla H_K(\bar{x})$ are linearly independent, this equation shows that $\nabla^2 \tilde{m}_\alpha(\bar{x})$ is positive definite for $\alpha \in (0,1)$ and is negative definite for $\alpha \in (1,\infty)$. □

We note that the condition of Theorem 5.1 is a special case of the standard second-order sufficient condition for $\bar{x}$ to be an isolated local minima of the nonlinear programming problem $\min\{ -\langle G(x), H(x) \rangle \mid -G(x) \leq 0, H(x) \leq 0 \}$ (see [FiM90, Thm. 4]). This can be seen by observing that $-H(\bar{x})$ and $G(\bar{x})$ form a Karush–Kuhn–Tucker vector for this problem.

6. Reformulation of GNCP with $\mathcal{K} = \Re^n_+$ as NCP. In the special case of GNCP in which $m = n$, $\mathcal{K} = \Re^n_+$, and G has an inverse, it is tempting to make the substitution $z = G(x)$ to reduce GNCP to the following NCP (RNCP):

$$\text{(RNCP)} \qquad \text{Find } z \in \Re^n \text{ satisfying } \quad \langle z, Q(z)\rangle = 0,\ z \geq 0,\ Q(z) \geq 0,$$

where we let

$$Q(z) = -H(G^{-1}(z))$$

for all $z \in \Re^n$. Then we can apply the implicit Lagrangian to RNCP and obtain as a merit function for RNCP on $\Re^n$ the function $h_\alpha : \Re^n \to \Re$ given by

$$h_\alpha(z) = \langle z, Q(z)\rangle + \frac{1}{2\alpha}\left(\|[z - \alpha Q(z)]_+\|^2 - \|z\|^2 + \|[Q(z) - \alpha z]_+\|^2 - \|Q(z)\|^2\right),$$

where $\alpha \in (1, \infty)$. By [YaF95, Thm. 2.1] (or, more generally, by Theorem 4.1), if Q is differentiable and ∇Q is positive definite everywhere, the stationary points of h_α coincide with the solutions to RNCP.

Unfortunately, the preceding reduction does not buy us anything new. First, it is easily seen using the connection between the implicit Lagrangian and $\tilde{m}_\alpha$ (see the discussion following Theorem 3.2) that

$$h_\alpha(G(x)) = -\tilde{m}_\alpha(x)$$

for all $x \in \Re^m$. Second, it is straightforward to check that if G and H are differentiable, we have

$$\nabla Q(G(x)) = -[\nabla G(x)]^{-1}\nabla H(x)$$

for all $x \in \Re^m$, so the preceding sufficient condition for the stationary points of h_α to coincide with the solutions to RNCP (i.e., ∇Q be positive definite everywhere) is no better than the sufficient condition for the stationary points of $\tilde{m}_\alpha$ to coincide with the solutions to GNCP (see condition B2 in Theorem 4.2).

7. Conclusion and extensions. We have shown that quadratic regularization of the linearized gap function provides a unified (and simplified) approach to generating merit functions for GNCP that are differentiable whenever G and H are. We have also studied the growth properties of these merit functions and have given conditions under which the stationary points of one such merit function coincide with the solutions to GNCP and, in the case $m = n$ and $\mathcal{K} = \Re^n_+$, conditions under which the global minima/maxima of this merit function are isolated. Much of our results and analysis also extend to a Hilbert space setting.

There are a number of directions in which our results and analysis can be extended. For instance, we can study merit functions for GNCP based on nonquadratic regularization. Also, we can study conditions under which the natural residual $\|R_1\|$ provides local/global error bound for GNCP on $\Re^m$. We can also seek to derive an analog of Theorem 4.1 for $\hat{m}_\alpha$. Finally, we can study descent methods for minimizing (respectively, maximizing) $\tilde{m}_\alpha$ for $\alpha \in (0, 1)$ (respectively, for $\alpha \in (1, \infty)$), as is done in [YaF95, §3] for the case of NCP. For example, it follows from the proof of Theorem 4.1 that, in the case in which G and H satisfy condition A1, the vector

$$d(x) = (\Lambda(x)[\nabla G(x)]^{-1})^* \left(R_\alpha(x) + \frac{1}{\alpha} S_\alpha(x)\right)$$

is a descent direction for $\tilde{m}_\alpha$ at x. Then, by using $d(x)$, we can develop a descent method for minimizing $\tilde{m}_\alpha$ and study the convergence properties of this method.

An interesting question, raised by Mangasarian, concerns whether $\tilde{m}_\alpha$ can be alternatively derived using the approach he and Solodov used in [MaS93] to derive the implicit Lagrangian, namely, taking the augmented Lagrangian associated with the "natural" nonlinear programming formulation of the complementarity problem and replacing the Lagrange multipliers by the complementary quantities. As we show below, the answer is indeed a "yes." Consider the following nonlinear programming formulation of GNCP:

$$\min\{\ -\langle G(x), H(x)\rangle \mid G(x) \in \mathcal{K},\ H(x) \in \mathcal{K}^\circ\ \},$$

whose associated augmented Lagrangian, with v and u denoting the Lagrange multiplier vectors attached to the constraints $G(x) \in \mathcal{K}$ and $H(x) \in \mathcal{K}^\circ$, respectively, is

$$L_\alpha(x,u,v) = -\langle G(x), H(x)\rangle + \min_{\substack{\psi\in\mathcal{K}\\ \zeta\in\mathcal{K}^\circ}} \Big\{\langle G(x)-\psi, v\rangle + \langle H(x)-\zeta, u\rangle + \frac{\alpha}{2}\|G(x)-\psi\|^2 + \frac{\alpha}{2}\|H(x)-\zeta\|^2\Big\},$$

with $\alpha \in (0,\infty)$. It is readily verified using the formula for $\tilde{\pi}_{1/\alpha}$ that

$$L_\alpha(x, G(x), H(x)) = \tilde{\pi}_{1/\alpha}(G(x), H(x)) = \tilde{m}_{1/\alpha}(x) = -\tilde{m}_\alpha(x)$$

for all $x \in \Re^m$, where the last equality follows from Theorem 3.2.

Acknowledgments. We thank Dr. C. Kanzow and two anonymous referees for their comments which led to improvements on the original version of this paper.

REFERENCES

[Auc89] G. AUCHMUTY, *Variational principles for variational inequalities*, Numer. Funct. Anal. Optim., 10 (1989), pp. 863–874.

[Aus76] A. AUSLENDER, *Optimisation: Méthodes Numériques*, Masson, Paris, 1976.

[BeT89] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[DoM80] I. C. DOLCETTA AND U. MOSCO, *Implicit complementarity problems and quasi-variational inequalities*, in Variational Inequalities and Complementarity Problems: Theory and Applications, R. W. Cottle, F. Giannessi, and J.-L. Lions, eds., Wiley, New York, 1980, pp. 75–87.

[FiM90] A. V. FIACCO, AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[Fuk92] M. FUKUSHIMA, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.

[GaB84] E. M. GAFNI AND D. P. BERTSEKAS, *Two-metric projection methods for constrained optimization*, SIAM J. Control Optim., 22 (1984), pp. 936–964.

[Kan96] C. KANZOW, *Nonlinear complementarity as unconstrained optimization*, J. Optim. Theory Appl., 88 (1996), to appear.

[LuT92] Z.-Q. LUO AND P. TSENG, *On a global error bound for a class of monotone affine variational inequality problems*, Oper. Res. Lett., 11 (1992), pp. 159–165.

[LuT93] ———, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res., 46 (1993), pp. 157–178.

[LMRS94] Z.-Q. LUO, O. L. MANGASARIAN, J. REN, AND M. V. SOLODOV, *New error bounds for the linear complementarity problem*, Math. Oper. Res., 19 (1994), pp. 880–892.

[Man90] O. L. MANGASARIAN, *Error bounds for nondegenerate monotone linear complementarity problems*, Math. Programming, 48 (1990), pp. 437–445.

[MaS93] O. L. MANGASARIAN AND M. V. SOLODOV, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming, 62 (1993), pp. 277–297.

[MaP90] R. MATHIAS AND J.-S. PANG, *Error bounds for the linear complementarity problem with a P-matrix*, Linear Algebra Appl., 132 (1990), pp. 123–136.

[Noo88] M. A. NOOR, *Quasi-complementarity problem*, J. Math. Anal. Appl., 130 (1988), pp. 344–353.

[Pan81] J.-S. PANG, *The implicit complementarity problem*, in Nonlinear Programming 4, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1981, pp. 487–518.

[Pan87] ———, *A posteriori error bounds for the linearly-constrained variational inequality problem*, Math. Oper. Res., 12 (1987), pp. 474–484.

[Pan95] ———, *Complementarity problems*, in Handbook on Global Optimization, R. Horst and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1995, pp. 271–338.

[TFI93] K. TAJI, M. FUKUSHIMA, AND T. IBARAKI, *A globally convergent Newton method for solving strongly monotone variational inequalities*, Math. Programming, 58 (1993), pp. 369–383.

[YaF95] N. YAMASHITA AND M. FUKUSHIMA, *On stationary points of the implicit Lagrangian for nonlinear complementarity problems*, J. Optim. Theory Appl., 84 (1995), pp. 653–663.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 461–487, May 1996
© 1996 Society for Industrial and Applied Mathematics
011

AN SQP ALGORITHM FOR FINELY DISCRETIZED CONTINUOUS MINIMAX PROBLEMS AND OTHER MINIMAX PROBLEMS WITH MANY OBJECTIVE FUNCTIONS*

JIAN L. ZHOU† AND ANDRÉ L. TITS‡

Abstract. A common strategy for achieving global convergence in the solution of semi-infinite programming (SIP) problems, and in particular continuous minimax problems, is to (approximately) solve a sequence of discretized problems, with progressively finer discretization meshes. Finely discretized minimax and SIP problems, as well as other problems with many more objectives/constraints than variables, call for algorithms in which successive search directions are computed based on a small but significant subset of the objectives/constraints, with ensuing reduced computing cost per iteration and decreased risk of numerical difficulties. In this paper, a sequential quadratic programming (SQP)-type algorithm is proposed that incorporates this idea in the particular case of minimax problems. The general case will be considered in a separate paper. The quadratic programming subproblem that yields the search direction involves only a small subset of the objective functions. This subset is updated at each iteration in such a way that global convergence is ensured. Heuristics are suggested that take advantage of a possible close relationship between "adjacent" objective functions. Numerical results demonstrate the efficiency of the proposed algorithm.

Key words. continuous minimax, semi-infinite programming, many constraints, sequential quadratic programming, discretization, global convergence

AMS subject classifications. 49M07, 49M37, 49M39, 65K05, 90C06, 90C30, 90C34

1. Introduction.

Optimization problems that arise in engineering design often belong to the class of semi-infinite programming (SIP) problems, i.e., they involve specifications that are to be satisfied over an interval of values of an independent parameter such as time, frequency, temperature, or modeling error (see, e.g., [2], [3], [30], [33]). A simple example is given by the problem

$$\text{(SI)} \qquad \text{minimize } f(x) \quad \text{s.t.} \quad \Phi_{[0,1]}(x) \leq 0, \ x \in R^n,$$

with

$$\Phi_{[0,1]}(x) := \sup_{\omega \in [0,1]} \phi(x, \omega).$$

An important class of SIP problems is that of "continuous" minimax problems such as

$$\text{(CMM)} \qquad \text{minimize } \sup_{\omega \in [0,1]} \phi(x, \omega).$$

Note that (CMM) can be equivalently formulated in the form of (SI) as

$$\text{minimize } x^0 \quad \text{s.t.} \sup_{\omega \in [0,1]} \phi(x, \omega) \leq x^0, (x, x^0) \in R^{n+1}.$$

*Received by the editors January 11, 1993; accepted for publication (in revised form) September 7, 1994. This research was supported in part by the National Science Foundation's Engineering Research Centers Program NSFD-CDR-88-03012, by National Science Foundation grant DMC-88-15996, and by a grant from the Westinghouse Corporation.

†Institute for Systems Research, University of Maryland, College Park, MD 20742.

‡Institute for Systems Research and Electrical Engineering Department, University of Maryland, College Park, MD 20742; all correspondence should be addressed to this author.

The difficulties in solving (SI), and in particular (CMM), stem mostly from the fact that (i) the accurate evaluation of $\Phi_{[0,1]}$ for each x involves a potentially time-consuming global maximization, and (ii) $\Phi_{[0,1]}$ is nondifferentiable in general, even when ϕ is smooth. Various approaches have been proposed to circumvent these difficulties (see [18] for a recent survey). Some algorithms are based on the characterization of maximizers of $\phi(x,\cdot)$ over $[0,1]$ in the neighborhood of a local solution of (SI) (see, e.g., [12], [17], [20], [34]). Under mild assumptions, the set of such maximizers contains a "small" number of points (for small n). The solution of the original problem can then be reduced to the solution of a problem involving approximations to these maximizers $\omega_i(x)$. The application of Newton's method or a sequential quadratic programming (SQP) method to the reduced problem (with constraints $\phi(x,\omega_i(x)) \leq 0$) brings about a fast local rate of convergence. However, global convergence, when ensured at all, involves a potentially very costly line search [5], [42]. A large class of globally convergent algorithms, on the other hand, is based on approximating $\Phi_{[0,1]}$ by means of progressively finer discretizations of $[0,1]$, i.e., substituting for (SI) the problems

$$\text{(DSI)} \qquad \text{minimize} \quad f(x) \quad \text{s.t.} \quad \phi(x,\omega) \leq 0 \quad \forall \omega \in \Omega$$

with, for instance,

$$\Omega = \left\{0, \frac{1}{q}, \frac{2}{q}, \ldots, \frac{(q-1)}{q}, 1\right\},$$

where q, a positive integer, is progressively increased (see, e.g., [10], [13], [16], [27], [31], [32], [34], [38]). The overall performance of these algorithms depends heavily on the performance at each discretization level, especially when q becomes large.

Problem (DSI) involves finitely many smooth constraints and thus in principle can be solved by classical constrained optimization techniques. Yet typically, if q is large compared to the number n of variables, only a small portion of the constraints are active at the solution. Suitably taking advantage of this situation may lead to substantial computational savings. Similar considerations arise in connection with inequality constrained optimization problems of the form

$$\text{(MC)} \qquad \text{minimize} \quad f(x) \quad \text{s.t.} \quad \phi_i(x) \leq 0, \quad i = 0, \ldots, \ell,$$

in which $\ell \gg n$, i.e., in which constraints far outnumber variables. The minimax problem (here with finitely many objective functions) is an important special case of this problem. Examples of (MC) include mechanical design problems involving trusses (see, e.g., [37], [43] or papers in [6], [25]). Note that there is no essential difference between (DSI) and (MC). Their similarity is particularly strong if the constraints in (MC) are "sequentially related" in the sense that the values taken by ϕ_i are typically close to those taken by ϕ_{i+1}.

In [32], [27], (DSI) is solved by means of first-order (thus slow) methods. In [32], based on ideas of Zoutendijk [46] and Polak [29, §4.3], the construction of the search direction at iteration k makes use of the gradients $\nabla_x \phi(x_k,\omega)$ at all points $\omega \in \Omega$ at which $\phi(x_k,\omega) \geq -\epsilon$ ("ϵ-active" constraints), where $\epsilon > 0$ is appropriately small. When the discretization is fine, however, the set of such points is often unduly large because it contains entire neighborhoods of local maximizers. By proposing a scheme that suitably detects "critical" values of ω and "remembers" them from iteration to iteration in a manner reminiscent of bundle-type methods in nonsmooth optimization (see, e.g., [21], [23]), the authors of [27] show that only a small subset of these points

need be used. Specifically, at iteration k of that algorithm, a first-order direction d_k is computed using a certain subset Ω_k of Ω. After a new iterate x_{k+1} has been obtained, a new set Ω_{k+1} is constructed by including (i) all ω's that globally maximize $\phi(x_{k+1}, \cdot)$ over Ω; (ii) all ω's that globally maximize $\phi(\bar{x}_{k+1}, \cdot)$, where $\bar{x}_{k+1}$ is a trial point that was rejected in the previous line search; and (iii) all ω's in Ω_k that affected direction d_k. This scheme is shown in [27] to induce global convergence. It is efficient because, under mild assumptions, the dimension of the quadratic programming problem that yields d_k is moderate and gradient evaluations are required only at a few grid points. However, at each level of discretization (i.e., for each fixed q), the algorithm proposed in [27] (like that proposed in [32]) exhibits at best a linear rate of convergence.

SQP-type algorithms, while often impractical for problems with large numbers of variables, are particularly suited to various classes of engineering applications where the number of variables is not too large but evaluations of objective/constraint functions and their gradients are highly time consuming. Indeed, because these algorithms use quadratic programs as successive models, progress between (expensive) function evaluations is typically significantly better than it is with algorithms making use of mere linear systems of equations as models. In the context of SQP-type algorithms for the solution of problems with many constraints, Biggs [1] proposed replacing with equality constraints the active inequality constraints and ignoring all other inequality constraints in the computation of the search direction. Much later, Polak and Tits [34] and Mine, Fukushima, and Tanaka [24] adapted the "ϵ-active" idea to the SQP context and Powell [36] proposed a "tolerant" algorithm for linearly constrained problems, which also borrows from the "ϵ-active" concept. Again, however, in the case of finely discretized SIP problems, the number of constraints may be unduly large. Recently, Conn and Li [4] proposed a working set scheme for the minimax problem and obtained promising numerical results. Finally, in [41], Schittkowski proposed modifications of standard SQP methods for the solution of problems with many constraints. However, no convergence analysis was provided; in practice global convergence may or may not take place, depending on the heuristics used to update an active working set of constraints.

In this paper, we propose and analyze an SQP-type algorithm based on the scheme introduced in [27] for the special case of the discretized minimax problem

$$\text{(P)} \qquad \text{minimize} \quad \max_{\omega \in \Omega} \phi(x, \omega),$$

where Ω is again a finite set. The general discretized SIP case involves additional intrinsic difficulties and will be considered in a separate paper. We define

$$\Phi(x) = \max_{\omega \in \Omega} \phi(x, \omega).$$

At iteration k, given an iterate x_k and a subset Ω_k of Ω, a search direction d_k is obtained as the solution of the "quadratic program" $\text{QP}(x_k, H_k, \Omega_k)$.[1] Here, for any $x \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$ symmetric positive definite, and $\hat{\Omega} \subseteq \Omega$, $\text{QP}(x, H, \hat{\Omega})$ is defined by

$$\text{QP}(x, H, \hat{\Omega}) \qquad \text{minimize} \quad \frac{1}{2}\langle d, Hd \rangle + \Phi'_{\hat{\Omega}}(x, d) \quad \text{s.t.} \quad d \in R^n,$$

[1] $\text{QP}(x, H, \hat{\Omega})$ is equivalent to the true quadratic program (over R^{n+1}) minimize $\frac{1}{2}\langle d, Hd \rangle + d^0$ s.t. $\phi(x, \omega) + \langle \nabla_x \phi(x, \omega), d \rangle - \Phi_{\hat{\Omega}}(x) - d^0 \leq 0 \ \ \forall \omega \in \hat{\Omega}$.

where

$$\Phi'_{\hat{\Omega}}(x,d) = \max_{\omega\in\hat{\Omega}}\{\phi(x,\omega) + \langle\nabla_x\phi(x,\omega),d\rangle\} - \Phi_{\hat{\Omega}}(x) \tag{1.1}$$

is a first-order approximation to $\Phi_{\hat{\Omega}}(x+d) - \Phi_{\hat{\Omega}}(x)$, with

$$\Phi_{\hat{\Omega}}(x) = \max_{\omega\in\hat{\Omega}}\phi(x,\omega).$$

A line search (e.g., of Armijo-type such as that suggested by Han [14], [15]) is performed along direction d_k to obtain a next iterate $x_{k+1} = x_k + t_k d_k$, with $t_k \in (0,1]$; H_k is updated to H_{k+1}; and a new subset Ω_{k+1} of Ω is constructed according to a scheme inspired from that used in [27]. In particular, if $t_k < 1$, Ω_{k+1} includes a point $\bar{\omega}_k$ that caused the last trial point to be rejected by the line search. However, in the present context, a difficulty arises. Suppose $\bar{\omega}_k$ was not in Ω_k. The rationale for including it in Ω_{k+1} is that, had it been included in Ω_k, a larger step would likely have been accepted (since $\bar{\omega}_k$ is now preventing a larger step). In the context of [27] where a first-order search direction is used (i.e., $H_k = I$ for all k), it follows that d_{k+1} will likely allow a larger step to be taken. In the current framework, however, it is unclear whether $\bar{\omega}_k$ is of any help in the new metric H_{k+1}, and global convergence may not occur. One remedy is to not update H_k whenever $t_k < 1$ and $\bar{\omega}_k \notin \Omega_k$ is picked by the algorithm. Because it will be proved that eventually $\bar{\omega}_k$ can be picked only from Ω_k (Lemma 3.14), such a scheme will not prevent normal updating from eventually taking place. (Thus it will not jeopardize the anticipated superlinear rate of convergence.) Yet disallowing normal updating of H_k in early iterations can hinder the algorithm's effectiveness. To obviate this effect we will disallow normal updating of H_k only if the additional condition $t_k < \delta$ is satisfied, where δ is a small positive number. Indeed, if t_k stays bounded away from zero, then $\{d_k\}$ must go to zero (Lemma 3.3(iv)) and global convergence takes place in any case (Lemma 3.4(ii)). It is shown that this overall algorithm indeed achieves global convergence and maintains a fast rate of local convergence.

A well-known possible adverse effect is that the line search may truncate the unit step even arbitrarily close to a solution, thus preventing superlinear convergence (Maratos effect). It will be shown that this can be avoided by incorporating in the basic algorithm standard techniques such as a second-order correction (see, e.g., [28], [44]).

Algorithm 2.1 allows additional ω's to be included in Ω_k at each iteration. Clever heuristics may significantly speed up the algorithm, especially in early iterations. In our implementation (discussed in a subsequent section) we paid special attention to finely discretized SIP problems in which $\phi(x,\omega)$ is continuous in ω and to other problems in which "adjacent" objectives are closely related.

The remainder of the paper is organized as follows. The basic algorithm is stated in §2. A complete convergence analysis is presented in §3. In §4, implementation issues are discussed and numerical results are reported. Section 5 is devoted to final remarks. The paper ends with an appendix, §6, containing proofs of results in §3.

2. Preliminaries and algorithm statement. The following assumption is made throughout.

Assumption 1. For every $\omega \in \Omega$, $\phi(\cdot,\omega) : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.

Let x^* be a local minimizer for (P). Then (see, e.g., [15]) it is a *Karush–Kuhn–Tucker (KKT) point* for (P), i.e., there exist *KKT multipliers* μ_ω^*, $\omega \in \Omega$ such that

$$\text{(2.1)} \qquad \begin{cases} \displaystyle\sum_{\omega\in\Omega} \mu_\omega^* \nabla_x \phi(x^*,\omega) = 0, \\ \mu_\omega^* \geq 0 \ \forall \omega \in \Omega \ \text{ and } \ \displaystyle\sum_{\omega\in\Omega} \mu_\omega^* = 1, \\ \mu_\omega^* = 0 \ \forall \omega \in \Omega \ \text{ s.t. } \ \phi(x^*,\omega) < \Phi(x^*). \end{cases}$$

It is readily verified that there is a natural correspondence between the KKT points of (P) and those of the equivalent constrained minimization problem

$$\text{(P}_{\text{eq}}\text{)} \qquad \text{minimize } \ x^0 \ \text{ s.t. } \ \Phi(x) \leq x^0, \ \ x \in R^n, \ \ x^0 \in R.$$

Specifically, the following lemma holds.

LEMMA 2.1. *A point x^* is a KKT point for* (P) *if and only if* $(x^*, \Phi(x^*))$ *is a KKT point for* (P_{eq}). *The associated KKT multipliers are identical.*

Similarly, given $x \in R^n$, $H = H^T > 0$, and $\hat{\Omega} \subseteq \Omega$, if d solves $\text{QP}(x, H, \hat{\Omega})$, then it is a KKT point for $\text{QP}(x, H, \hat{\Omega})$, i.e., there exist μ_ω, $\omega \in \hat{\Omega}$ such that

$$\text{(2.2)} \qquad \begin{cases} Hd + \displaystyle\sum_{\omega\in\hat{\Omega}} \mu_\omega \nabla_x \phi(x,\omega) = 0, \\ \mu_\omega \geq 0 \ \forall \omega \in \hat{\Omega} \ \text{ and } \ \displaystyle\sum_{\omega\in\hat{\Omega}} \mu_\omega = 1, \\ \mu_\omega = 0 \ \forall \omega \in \hat{\Omega} \ \text{ s.t. } \ \phi(x,\omega) + \langle \nabla_x \phi(x,\omega), d\rangle - \Phi_{\hat{\Omega}}(x) < \Phi'_{\hat{\Omega}}(x,d). \end{cases}$$

Moreover, since $\varphi(d) := \frac{1}{2}\langle d, Hd\rangle + \Phi'_{\hat{\Omega}}(x,d)$ is strictly convex in d (sum of a strictly convex function and a convex function), it has a unique minimizer d^*. It follows that the equivalent quadratic program in R^{n+1} has a unique minimizer $(d^*, \varphi(d^*))$ and thus that $\text{QP}(x, H, \hat{\Omega})$ has d^*, its global minimizer, as its only KKT point.

We are now ready to make the rule for updating Ω_k precise. Following [27], Ω_{k+1} contains the union of three sets.[2] Given $x \in R^n$, let

$$\Omega_{\max}(x) = \{\omega \in \Omega : \phi(x,\omega) = \Phi(x)\}$$

be the set of maximizers of $\phi(x,\cdot)$. The first component of Ω_{k+1} is $\Omega_{\max}(x_{k+1})$. Indeed if $\Omega_{\max}(x_{k+1})$ were not included, d_{k+1} might not be a direction of descent for Φ at x_{k+1}. The second component of Ω_{k+1} is obtained from the line search. While the essence of the ideas put forth in this paper is independent of the specifics of this line search, for the sake of exposition we will consider the case of an Armijo-type line search inspired by the line search used by Han [14], [15]. Thus $x_{k+1} = x_k + t_k d_k$, where t_k is the largest number t in $\{1, \beta, \beta^2, \ldots\}$ satisfying

$$\text{(2.3)} \qquad \Phi(x_k + t d_k) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k\rangle,$$

where $\alpha \in (0, \frac{1}{2})$ and $\beta \in (0,1)$ are fixed. Suppose the line search at iteration k results in $t_k < 1$, implying that the line search test (2.3) is violated at $x_k + \frac{t_k}{\beta} d_k$. A next

[2] In [27], Ω_{k+1} is set to be *equal* to this union.

search direction taking this into account is called for. Thus, Ω_{k+1} will include some $\bar{\omega}_k$ such that

$$\phi\left(x_k + \frac{t_k}{\beta} d_k, \bar{\omega}_k\right) > \Phi(x_k) - \alpha \frac{t_k}{\beta} \langle d_k, H_k d_k \rangle.$$

Finally, to avoid zigzagging (which could prevent global convergence; see the example in [27]) it is important that key elements in Ω_k be kept in Ω_{k+1}. A natural choice is to preserve all $\omega \in \Omega_k$ that are binding at the solution of $\mathrm{QP}(x_k, H_k, \Omega_k)$, i.e., those ω for which the corresponding multiplier $\mu_{k,\omega}$ is strictly positive.[3] (Clearly, this is also needed for fast local convergence.) Thus, the third component of Ω_{k+1} is

$$\Omega_k^b = \{\omega \in \Omega_k : \mu_{k,\omega} > 0\}.$$

Thus the overall algorithm for the solution of (P) is as follows.

ALGORITHM 2.1.
Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $0 < \delta \ll 1$.
Data. $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$ with $H_0 = H_0^T > 0$.
Step 0. *Initialization.* Set $k = 0$ and pick $\Omega_0 \supseteq \Omega_{\max}(x_0)$.
Step 1. *Computation of search direction and step length.*
(i) Compute d_k by solving $\mathrm{QP}(x_k, H_k, \Omega_k)$. If $d_k = 0$, stop.
(ii) Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying

$$\Phi(x_k + t d_k) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k \rangle. \tag{2.4}$$

Step 2. *Updates.* Set $x_{k+1} = x_k + t_k d_k$. If $t_k < 1$, pick $\bar{\omega}_k$ such that

$$\phi\left(x_k + \frac{t_k}{\beta} d_k, \bar{\omega}_k\right) > \Phi(x_k) - \alpha \frac{t_k}{\beta} \langle d_k, H_k d_k \rangle.$$

Pick

$$\Omega_{k+1} \supseteq \begin{cases} \Omega_{\max}(x_{k+1}) \cup \Omega_k^b & \text{if } t_k = 1, \\ \Omega_{\max}(x_{k+1}) \cup \Omega_k^b \cup \{\bar{\omega}_k\} & \text{if } t_k < 1. \end{cases} \tag{2.5}$$

If $t_k \leq \delta$ and $\bar{\omega}_k \notin \Omega_k$, set $H_{k+1} = H_k$; otherwise, compute a new positive definite approximation H_{k+1} to the Hessian of the Lagrangian of (P) at the solution. Set $k = k + 1$. Go back to Step 1. □

3. Convergence analysis. Although (P) takes the form of an ordinary minimax problem, the classical convergence analysis for such problems (e.g., [14], [15]) cannot be directly applied to the present situation since, at each iteration, only a subset of the discretized set Ω is employed to construct a search direction.

3.1. Global convergence. The following additional standard assumptions are made.

Assumption 2. For any $x_0 \in \mathbb{R}^n$, the level set $\{x \in \mathbb{R}^n : \Phi(x) \leq \Phi(x_0)\}$ is compact.

Assumption 3. There exist σ_1, $\sigma_2 > 0$ such that

$$\sigma_1 \|d\|^2 \leq \langle d, H_k d \rangle \leq \sigma_2 \|d\|^2 \quad \forall d \in \mathbb{R}^n, \forall k.$$

[3] If the multiplier vector associated with $\mathrm{QP}(x_k, H_k, \Omega_k)$ is not unique, any of the choices is appropriate.

We first show that, because of the fact that Ω_k always contains $\Omega_{\max}(x_k)$, Algorithm 2.1 is well defined.

LEMMA 3.1. *At any iteration k there exists $\bar{t}_k > 0$ such that, for all $t \in [0, \bar{t}_k]$,*

$$\Phi(x_k + td_k) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k \rangle.$$

Proof. Since d_k solves $\mathrm{QP}(x_k, H_k, \Omega_k)$, it yields an objective value no larger than that at $d = 0$ for that problem, and thus

$$\Phi'_{\Omega_k}(x_k, d_k) \leq -\frac{1}{2}\langle d_k, H_k d_k \rangle.$$

In view of (1.1) and Assumption 3, it follows that

$$\langle \nabla \phi(x_k, \omega), d_k \rangle < -\frac{1}{2}\langle d_k, H_k d_k \rangle \quad \forall \omega \in \Omega_k \ .$$

Since $\alpha < \frac{1}{2}$ and since $\Omega_k \supseteq \Omega_{\max}(x_k)$, it then follows that there exists $\hat{t}_k > 0$ such that, for all $t \in [0, \hat{t}_k]$,

$$\Phi_{\Omega_k}(x_k + td_k) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k \rangle.$$

On the other hand, since $\Omega_{\max}(x_k) \subseteq \Omega_k$, $\phi(x_k, \omega) < \Phi(x_k)$ for all $\omega \notin \Omega_k$. In view of the continuity assumptions, this implies that there exists $\bar{t}_k > 0$ such that, for all $t \in [0, \bar{t}_k]$,

$$\Phi(x_k + td_k) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k \rangle,$$

proving the claim. □

Thus the line search is always well defined and Algorithm 2.1 stops only when $d_k = 0$. The following lemma implies that, if this occurs, the last point x_k must be a KKT point.

LEMMA 3.2. *Let $H > 0$, $x \in \mathbb{R}^n$, and $\hat{\Omega} \subseteq \Omega$ with $\Phi_{\hat{\Omega}}(x) = \Phi(x)$. Then the unique KKT point d of $\mathrm{QP}(x, H, \hat{\Omega})$ is zero if and only if x is a KKT point for* (P).

Proof. Suppose the unique KKT point of $\mathrm{QP}(x, H, \hat{\Omega})$ is $d = 0$ and let $\{\hat{\mu}_\omega : \omega \in \hat{\Omega}\}$ be the associated KKT multipliers. In view of (2.2) and since $\Phi_{\hat{\Omega}}(x) = \Phi(x)$, the KKT condition (2.1) for (P) holds at x with multipliers $\mu_\omega = \hat{\mu}_\omega$ for $\omega \in \hat{\Omega}$ and $\mu_\omega = 0$ for $\omega \in \Omega \backslash \hat{\Omega}$. Thus, x is a KKT point for (P). The converse is similarly proved. □

We now assume that an infinite sequence $\{x_k\}$ is generated by Algorithm 2.1. The following facts are direct consequences of Lemma 3.1 and our assumptions.

LEMMA 3.3. (i) *The sequence $\{x_k\}$ is bounded*; (ii) *the sequence $\{d_k\}$ is bounded*; (iii) *the sequence $\{\Phi(x_k)\}$ converges*; *and* (iv) *the sequence $\{t_k d_k\}$ converges to zero.*

Proof. The claims follow directly from Assumptions 1, 2, and 3 and the fact that $\Phi(x_{k+1}) \leq \Phi(x_k) - \alpha t_k \langle d_k, H_k d_k \rangle$. (Since $|t_k| \leq 1$, convergence to zero of $\{t_k \langle d_k, H_k d_k \rangle\}$ implies convergence to zero of $\{\langle (t_k d_k), H_k (t_k d_k) \rangle\}$.) □

Now let v_k denote the optimal value of $\mathrm{QP}(x_k, H_k, \Omega_k)$, i.e.,

$$v_k = \frac{1}{2}\langle d_k, H_k d_k \rangle + \Phi'_{\Omega_k}(x_k, d_k). \tag{3.1}$$

Since $d = 0$ is feasible for $\mathrm{QP}(x_k, H_k, \Omega_k)$, v_k is nonpositive for all k. It turns out that convergence of $\{d_k\}$ to zero is equivalent to convergence of $\{v_k\}$ to zero and implies

that accumulation points of $\{x_k\}$ are KKT points. More generally, the following lemma holds.

LEMMA 3.4. *Let $K \subseteq \mathbb{N}$ be an infinite index set. Then* (i) *$\{d_k\}$ converges to zero on K if and only if $\{v_k\}$ converges to zero on K; and* (ii) *if $\{d_k\}$ converges to zero on K, then all accumulation points of $\{x_k\}_{k\in K}$ are KKT points for* (P).

Proof. Since, for all k, $\Phi_{\Omega_k}(x_k) = \Phi(x_k)$, it follows from (1.1) and (2.2) with $x = x_k$, $H = H_k$, $\hat{\Omega} = \Omega_k$, $d = d_k$ that, for all k and some $\mu_{k,\omega} \geq 0, \omega \in \Omega_k$, with $\sum_{\omega\in\Omega_k} \mu_{k,\omega} = 1$,

$$\begin{aligned}\Phi'_{\Omega_k}(x_k, d_k) &= \max_{\omega\in\Omega_k}\{\phi(x_k,\omega) + \langle\nabla_x\phi(x_k,\omega), d_k\rangle\} - \Phi(x_k)\\ &= \sum_{\omega\in\Omega_k} \mu_{k,\omega}\{\phi(x_k,\omega) + \langle\nabla_x\phi(x_k,\omega), d_k\rangle\} - \Phi(x_k)\\ &\leq \sum_{\omega\in\Omega_k} \mu_{k,\omega}\langle\nabla_x\phi(x_k,\omega), d_k\rangle,\end{aligned}$$

yielding, again from (2.2),

$$\Phi'_{\Omega_k}(x_k, d_k) \leq -\langle d_k, H_k d_k\rangle. \tag{3.2}$$

In view of (3.1), it follows that

$$v_k \leq -\frac{1}{2}\langle d_k, H_k d_k\rangle.$$

Thus, the "if" part of Lemma 3.4(i) follows directly from Assumption 3. On the other hand, if $\{d_k\}$ goes to zero on K, since $\{x_k\}$ is bounded, it follows from (1.1) that

$$\lim_{k\in K, k\to\infty} \Phi'_{\Omega_k}(x_k, d_k) = 0.$$

The "only if" part of Lemma 3.4(i) then follows from (3.1).

To prove Lemma 3.4(ii), suppose $\{d_k\}$ goes to zero on K and let $K' \subseteq K$ be any infinite index set such that $\{x_k\}$ converges to some $\hat{x}$ on K'. Without loss of generality, assume that, for some $\hat{\Omega} \subseteq \Omega$, $\Omega_k = \hat{\Omega}$ for all $k \in K'$. Then $\Phi_{\hat{\Omega}}(x_k) = \Phi(x_k)$ for all $k \in K'$ and thus $\Phi_{\hat{\Omega}}(x^*) = \Phi(x^*)$. In view of Assumption 3 and the boundedness of $\{\mu_{k,\omega}\}$ for all $\omega \in \hat{\Omega}$, there exists $K'' \subseteq K'$ such that $\{H_k\}$ converges to some H^* on K'' and, for each $\omega \in \hat{\Omega}$, there exists $\hat{\mu}_\omega$ such that $\{\mu_{k,\omega}\}$ converges to $\hat{\mu}_\omega$ on K''. Letting $\hat{\mu}_\omega = 0$ for $\omega \in \Omega\backslash\hat{\Omega}$, taking limits for $k \in K''$ in the optimality condition (2.2) associated with $\mathrm{QP}(x_k, H_k, \hat{\Omega})$, and comparing the resulting conditions with (2.1) shows that $\hat{x}$ is a KKT point for (P). □

The next lemma, which is the same as Lemma 4.7 in [21, Chap. 3], is central to the proof of global convergence.

LEMMA 3.5. *Let $x^* \in \mathbb{R}^n$ be such that*

$$\liminf_{k\to\infty} \max\{|v_k|, \|x_k - x^*\|\} = 0.$$

Then x^* is a KKT point for (P).

Proof. The assumption implies that there exists an infinite index set K such that $\{x_k\}$ converges to x^* and $\{v_k\}$ converges to zero, both on K. Thus, the conclusion follows from Lemma 3.4. □

The establishment of the global convergence of Algorithm 2.1 employs a contradiction argument inspired by [21, Chap. 3]. If $\{x_k\}$ has a limit point x^* that is not a KKT point, v_k is bounded away from zero on the corresponding subsequence (Lemma 3.5), with a uniform lower bound $\epsilon^* > 0$ for *all* subsequences over which $\{x_k\}$ converges to x^*. It is shown (Lemma 3.6) that in such a case $|v_{k+1}|$ is significantly smaller than $|v_k|$ on any such subsequence K. Since, in view of Lemma 3.3(iv), $\{x_{k+1}\}$ also converges to x^*, $|v_{k+2}|$ is also significantly smaller than $|v_{k+1}|$. A careful repeated application of this argument shows that $|v_k|$ becomes smaller than ϵ^* on a sequence at "finite distance" from K, which is a contradiction.

The proof of the following lemma is inspired by that of Lemma 4.11 in [21, Chap. 3] (see also the proof of Lemma 3.15 in [44]) and is given in the appendix. It relies crucially on the assumption that $\{\|H_{k+1} - H_k\|\} \to 0$ whenever $\{t_k\} \to 0$, which is ensured in Algorithm 2.1 by setting $H_{k+1} = H_k$ when t_k is small and $\bar{\omega}_k \notin \Omega_k$; it also relies on the inclusion in Ω_{k+1} of the second and third subsets in (2.5).

LEMMA 3.6. *There exists $c > 0$ such that, if K is an infinite index set on which $\{d_k\}$ is bounded away from zero, then there exists an integer N such that*

$$|v_{k+1}| \leq |v_k| - c|v_k|^2 \quad \forall k \geq N, k \in K. \tag{3.3}$$

The repeated application of this result yields the following lemma.

LEMMA 3.7. *There exists $c > 0$ such that, if K is an infinite index set on which $\{x_k\}$ is bounded away from KKT points, then, given any positive integer i_0, there exists an integer N such that*

$$|v_{k+i+1}| \leq |v_{k+i}| - c|v_{k+i}|^2 \quad \forall k \geq N, k \in K, \forall i \in [0, i_0]. \qquad \square$$

Proof. For any integer i, in view of Lemma 3.3(iv), $\{x_{k+i}\}$ is bounded away from KKT points for $k \in K$ and this in turn implies, in view of Lemma 3.4, that $\{d_{k+i}\}$ is bounded away from zero for $k \in K$. Therefore, in view of Lemma 3.6, for each i there exists N_i such that

$$|v_{k+i+1}| \leq |v_{k+i}| - c|v_{k+i}|^2 \quad \forall k \geq N_i, k \in K.$$

Choosing $N = \max_{0 \leq i \leq i_0}\{N_i\}$ proves the claim. $\square$

LEMMA 3.8. *Given $\eta > 0$ and $\epsilon > 0$, there exists an integer i_0 depending only on η and ϵ such that, for any sequence $\{z_i\}$ of real numbers satisfying*

$$0 \leq z_{i+1} \leq z_i - \eta z_i^2 \quad \forall i \in \mathbb{N},$$

$z_i < \epsilon$ for all $i \geq i_0$.

Proof. See the appendix. $\square$

We are now ready to establish the global convergence of Algorithm 2.1. The fact that c in Lemma 3.7 is independent of K and i_0 in Lemma 3.8 is independent of z_0 plays a crucial role in the proof.

THEOREM 3.9. *Let $\{x_k\}$ be the sequence generated by Algorithm 2.1. Then every accumulation point of $\{x_k\}$ is a KKT point.*

Proof. Let x^* and K be such that $\{x_k\}$ converges to x^* on K, an infinite index set. Proceeding by contradiction, we assume x^* is not a KKT point. It follows from Lemma 3.5 that there exists $\epsilon^* > 0$ such that

$$\liminf_{k \to \infty} \max\{|v_k|, \|x_k - x^*\|\} > \epsilon^*. \tag{3.4}$$

Thus, $\{v_k\}$ is bounded away from zero on K. In view of Lemma 3.4, $\{d_k\}$ is also bounded away from zero on K and, in view of Lemma 3.3(iv), $\{t_k\}$ converges to zero on K. Let c be as given by Lemma 3.7. Let i_0 be as given in Lemma 3.8 with $\epsilon = \epsilon^*$ and $\eta = c$. In view of Lemma 3.7, there exists an integer N such that

$$|v_{k+i+1}| \leq |v_{k+i}| - c|v_{k+i}|^2 \quad \forall k \geq N, k \in K, \forall i \in [0, i_0].$$

From the definition of i_0, it follows from Lemma 3.8 with $z_i = |v_{k+i}|$, $k \in K$, $i = 0, \ldots, i_0$ and $z_i = 0$, $k \in K$, $i > i_0$ that

$$|v_{k+i_0}| < \epsilon^* \quad \forall k \geq N, k \in K.$$

On the other hand, since by assumption x^* is not a KKT point and since in view of Lemma 3.3(iv) $\{x_{k+i_0}\}$ also converges to x^* on K, it follows from (3.4) that

$$\liminf_{k \in K, k \to \infty} |v_{k+i_0}| > \epsilon^*,$$

which is a contradiction. □

3.2. Local convergence. Under additional regularity conditions, it is shown that, close to a strong local minimizer x^*, the right-hand side of (2.5) becomes equal to $\Omega_{\max}(x^*)$ for all k and no $\omega \in \Omega \backslash \Omega_{\max}(x^*)$ hinders the line search, so that Algorithm 2.1 behaves as if solving (P) with Ω replaced with $\Omega_{\max}(x^*)$. Further, it is shown that H_k will be updated normally and thus will not be prevented from asymptotically suitably approximating $\nabla^2 L(x^*, \mu^*)$, the Hessian of the Lagrangian at the limit KKT pair. If H_k does become a suitable approximation to $\nabla^2 L(x^*, \mu^*)$ and if the full step of one is eventually accepted by the line search, 2-step superlinear convergence will result.

Assumption 1 is replaced by the following.

Assumption 1′. For every $\omega \in \Omega$, the function $\phi(\cdot, \omega) : \mathbb{R}^n \to \mathbb{R}$ is three times continuously differentiable.

Let x^* be an accumulation point of $\{x_k\}$ (thus a KKT point for (P)).

Assumption 4. Any scalars λ_ω, $\omega \in \Omega_{\max}(x^*)$ satisfying

$$\sum_{\omega \in \Omega_{\max}(x^*)} \lambda_\omega \nabla_x \phi(x^*, \omega) = 0 \quad \text{and} \quad \sum_{\omega \in \Omega_{\max}(x^*)} \lambda_\omega = 0$$

must be all zero.

Thus the KKT multipliers μ^*_ω, $\omega \in \Omega$ corresponding to x^* for problem (P) are unique.

Assumption 5. The second-order sufficiency conditions with strict complementary slackness are satisfied at x^*, i.e. (see, e.g., [14]), $\mu^*_\omega > 0$ for all $\omega \in \Omega_{\max}(x^*)$ and

$$\langle d, \nabla^2_{xx} L(x^*, \mu^*) d \rangle > 0 \quad \forall d \in \mathcal{S}^*, d \neq 0,$$

with

$$\nabla^2_{xx} L(x^*, \mu^*) = \sum_{\omega \in \Omega_{\max}(x^*)} \mu^*_\omega \nabla^2_{xx} \phi(x^*, \omega)$$

and

$$\mathcal{S}^* = \{d : \langle d, \nabla_x \phi(x^*, \omega) \rangle = 0 \ \ \forall \omega \in \Omega_{\max}(x^*)\}.$$

The following result is standard for ordinary constrained problems (see, e.g., [8, Thm. 2.3.2]). A proof in the minimax case is given in the appendix for the sake of completeness.

LEMMA 3.10. *The point x^* is an isolated KKT point for* (P).

PROPOSITION 3.11. *The entire sequence $\{x_k\}$ converges to x^*.*

Proof. The claim follows from Theorem 3.9 and Lemmas 3.10 and 3.3(iv). □

Much of the remainder of this section is devoted to showing that, close to x^*, the right-hand side of (2.5) remains constant and equal to $\Omega_{\max}(x^*)$ and no $\omega \notin \Omega\backslash\Omega_{\max}(x^*)$ hinders the line search, i.e., Algorithm 2.1 eventually behaves as if solving a minimax problem of the type

$$\text{minimize} \quad \max_{\omega\in\hat{\Omega}} \phi(x,\omega)$$

(with $\hat{\Omega} = \Omega_{\max}(x^*)$) with Ω_k always set to be equal to $\hat{\Omega}$ instead of updated according to the rule in Step 2. A consequence of this is that rate of convergence results obtained for such standard "constant Ω_k" algorithms hold for Algorithm 2.1. The first step is to show that, for k large enough, $\Omega_{\max}(x^*) \subseteq \Omega_k$ (Lemma 3.14). This is first proved on a subsequence using Lemma 3.6.

LEMMA 3.12. *There exists an infinite index set K such that*

$$\Omega_{\max}(x^*) \subseteq \Omega_k^b \quad \forall k \in K. \tag{3.5}$$

Proof. First, there exists an infinite index set K such that $\{d_k\}$ converges to zero on K. Indeed, if $\{d_k\}$ were bounded away from zero, it would follow from Lemma 3.6 that

$$|v_{k+1}| \leq |v_k| - c|v_k|^2 \quad \forall k \geq N$$

for some $c > 0$ and some integer N, implying that $\{v_k\}$ converges to zero, in violation of Lemma 3.4(i). Next, in view of the finite cardinality of Ω, without loss of generality, we may assume that $\Omega_k^b = \hat{\Omega}$ for all $k \in K$ for some constant set $\hat{\Omega}$. Let $\mu_k \in R^{|\Omega|}$ be a vector with components $\{\mu_{k,\omega}\}$ such that $\mu_{k,\omega}$, $\omega \in \hat{\Omega}$ are the KKT multipliers associated with $\mathrm{QP}(x_k, H_k, \Omega_k)$ and $\mu_{k,\omega} = 0$, $\omega \in \Omega\backslash\Omega_k$. Without loss of generality, $\{\mu_k\} \to \hat{\mu}$ as $k \to \infty$, $k \in K$ for some $\hat{\mu}$. We show that $\hat{\mu}$ together with x^* satisfies the KKT conditions (2.1) of the original problem. In view of Proposition 3.11, since $\{d_k\}$ converges to zero on K, taking limits in the optimality condition (2.2) associated with $\mathrm{QP}(x_k, H_k, \Omega_k)$, $k \in K$ yields, since $\hat{\mu}_\omega = 0$ for all $\omega \in \Omega\backslash\Omega_k$,

$$\sum_{\omega\in\Omega} \hat{\mu}_\omega \nabla_x \phi(x^*,\omega) = 0,$$

$$\hat{\mu}_\omega \geq 0 \ \forall \omega \in \Omega \quad \text{and} \quad \sum_{\omega\in\Omega} \hat{\mu}_\omega = 1,$$

$$\hat{\mu}_\omega = 0 \quad \forall \omega \in \Omega \text{ s.t. } \phi(x^*,\omega) < \Phi(x^*).$$

Therefore, x^* with $\{\hat{\mu}_\omega, \omega \in \hat{\Omega}; \hat{\mu}_\omega = 0, \omega \in \Omega\backslash\hat{\Omega}\}$ satisfies (2.1). Uniqueness of the multipliers for (P) at x^* and strict complementarity (Assumptions 4 and 5) imply that $\omega \in \hat{\Omega}$ for all ω such that $\phi(x^*,\omega) = \Phi(x^*)$, i.e., (3.5) holds. □

The following lemma, on the other hand, establishes that d_k is small whenever (3.5) holds.

LEMMA 3.13. *Let K be an infinite index set such that $\Omega_{\max}(x^*) \subseteq \Omega_k$ for all $k \in K$. Then $\{d_k\}$ converges to zero on K.*

Proof. Given $\hat{\Omega} \subseteq \Omega$, let $K_{\hat{\Omega}} = \{k \in K : \Omega_k = \hat{\Omega}\}$. For any $\hat{\Omega} \subseteq \Omega$ such that $K_{\hat{\Omega}}$ is an infinite set, we prove by contradiction that $\{d_k\}$ converges to zero on $K_{\hat{\Omega}}$. Since Ω has only finitely many subsets, the lemma will follow. Thus suppose that for some infinite index set $K' \subseteq K_{\hat{\Omega}}$, $\{d_k\}$ is bounded away from zero on K' and let $K'' \subseteq K'$ be such that $\{H_k\}$ converges to H^* on K'' for some $H^* > 0$ (such K'' exists in view of Assumption 3). Since $\Omega_{\max}(x^*) \subseteq \hat{\Omega}$, $\mathrm{QP}(x^*, H^*, \hat{\Omega})$ has $d = 0$ as its unique solution (Lemma 3.2). It follows from [39, Thm. 2.1] that $\{d_k\} \to 0$ as $k \to \infty$, $k \in K''$, contradicting the fact that $\{d_k\}$ is bounded away from zero on K'. ∎

LEMMA 3.14. *For k large enough, $\Omega_{\max}(x^*) \subseteq \Omega_k^b$.*

Proof. In view of Lemma 3.12, the claim holds on an infinite subsequence. To complete the proof we show that, given any infinite index set K such that $\Omega_{\max}(x^*) \subseteq \Omega_k^b$ for all $k \in K$, it holds that $\Omega_{\max}(x^*) \subseteq \Omega_{k+1}^b$ for all $k \in K$, k large enough. In view of the construction of Ω_{k+1}, it is enough to show that $\mu_{k+1,\omega} > 0$ for all $\omega \in \Omega_{\max}(x^*)$, $k \in K$, k large enough, where $\mu_{k+1,\omega}$, $\omega \in \Omega_{k+1}$ are the KKT mutipliers associated with $\mathrm{QP}(x_{k+1}, H_{k+1}, \Omega_{k+1})$. Thus let K be an infinite index set such that $\Omega_{\max}(x^*) \subseteq \Omega_k^b$ for all $k \in K$ (so that $\Omega_{\max}(x^*) \subseteq \Omega_{k+1}$ for all $k \in K$). Lemma 3.13 implies that $\{d_{k+1}\}$ converges to zero on K. Suppose by contradiction that there exists $\omega^* \in \Omega_{\max}(x^*)$ and an infinite index set $K' \subseteq K$ such that $\mu_{k+1,\omega^*} = 0$ for all $k \in K'$ (note that $\Omega_{\max}(x^*)$ is a finite set). An argument similar to that used in the proof of Lemma 3.12 shows that, in view of Assumption 4, $\mu_{\omega^*}^* = 0$, contradicting the strict complementarity (Assumption 5). ∎

The following result follows directly from Lemmas 3.13 and 3.14.

LEMMA 3.15. *The entire sequence $\{d_k\}$ converges to zero.* ∎

This leads to the main result of this section.

PROPOSITION 3.16. *For k large enough,*

$$\Omega_k^b = \Omega_{\max}(x^*),$$

$$\phi(x_k + td_k, \omega) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k \rangle \quad \forall t \in [0,1],\ \omega \in \Omega \backslash \Omega_{\max}(x^*), \tag{3.6}$$

and

$$\phi(x_{k+1}, \omega) < \Phi(x_{k+1}) \quad \forall \omega \in \Omega \backslash \Omega_{\max}(x^*).$$

Proof. To prove the first claim, in view of Lemma 3.14, it suffices to show that, for k large enough, $\Omega_k^b \subseteq \Omega_{\max}(x^*)$. To this end, let $\hat{\omega} \in \Omega \backslash \Omega_{\max}(x^*)$, i.e., suppose that $\phi(x^*, \hat{\omega}) < \Phi(x^*)$. Our continuity assumption and Proposition 3.11 then imply that, for k large enough, $\phi(x_k, \hat{\omega}) < \Phi(x_k)$, or equivalently $\phi(x_k, \hat{\omega}) < \max_{\omega \in \Omega_k} \phi(x_k, \omega)$, since by construction $\Omega_{\max}(x^*) \subseteq \Omega_k$. This, together with Proposition 3.11, Lemma 3.15, and the continuity assumption, implies that, for k large enough,

$$\phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \hat{\omega}), d_k \rangle < \max_{\omega \in \Omega_k} \{\phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k \rangle\},$$

so that $\mu_{k,\hat{\omega}} = 0$ and $\hat{\omega} \notin \Omega_k^b$, which thus proves the first claim. The second and third claims follow directly from Proposition 3.11, Lemma 3.15, and the continuity assumption. ∎

In view of Lemma 3.15, it follows from Proposition 3.16 that, if Ω_{k+1} is always picked to be equal to the right-hand side of (2.4) (rather than to merely contain it), then, for k large enough, $\Omega_k = \Omega_{\max}(x^*)$. Whether or not this is the case, for k large enough, Algorithm 2.1 will behave exactly as if solving the problem

$$\text{minimize} \quad \max_{\omega \in \Omega_{\max}(x^*)} \phi(x, \omega) \tag{3.7}$$

with $\Omega_k = \Omega_{\max}(x^*)$ selected at each iteration. (Indeed any ω not in Ω_k^b does not affect direction d_k and, when d_k is small enough, any ω not in $\Omega_{\max}(x^*)$ does not affect the line search.) In particular, for k large enough, if some $\bar{\omega}_k$ is picked by Algorithm 2.1, it must already be in Ω_k. Thus H_k is eventually updated at every iteration and the local behavior of Algorithm 2.1 becomes identical to that of the algorithm proposed by Han [14], [15] (except for a different rule for selecting t_k satisfying (2.4)).

Suppose that, as a result of the updating rule, H_k approaches the Hessian of the Lagrangian in the sense that

$$\lim_{k\to\infty} \frac{\|P_k\{H_k - \nabla^2_{xx}L(x^*,\mu^*)\}P_k d_k\|}{\|d_k\|} = 0, \tag{3.8}$$

where the matrices P_k are defined by

$$P_k = I - R_k(R_k^T R_k)^{-1}R_k^T$$

with $R_k = [\nabla_x\phi(x,\omega_i) - \nabla_x\phi(x,\omega_1) : i = 2,\dots,s]$,[4] $\omega_1,\dots,\omega_s$ being the elements of $\Omega_{\max}(x^*)$. Suppose moreover that $t_k = 1$ for k large enough. Then (see [14]), the convergence rate is 2-step superlinear, i.e.,

$$\lim_{k\to\infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

To achieve $t_k = 1$ for k large enough, it is necessary to introduce a scheme to avoid the Maratos effect. One option is to adopt a second-order correction such as that used in [28] and [45]. (In the latter, it is combined with a "nonmonotone line search"; using such a line search here would entail a more complicated analysis.) Specifically, Step 1(ii) in Algorithm 2.1 is replaced with the following.

Step 1(ii$_1$). If $\Phi(x_k + d_k) \le \Phi(x_k) - \alpha\langle d_k, H_k d_k\rangle$, set $\tilde{d}_k = 0$. Otherwise, compute a correction $\tilde{d}_k$ solution of the problem[5] in $\tilde{d}$:

$$\text{minimize } \frac{1}{2}\langle d_k + \tilde{d}, H_k(d_k + \tilde{d})\rangle + \tilde{\Phi}'(x_k + d_k, x_k, \tilde{d}),$$

where

$$\tilde{\Phi}'(x_k + d_k, x_k, \tilde{d}) = \max_{\omega\in\Omega_k}\{\phi(x_k + d_k,\omega) + \langle\nabla\phi(x_k,\omega), \tilde{d}\rangle\} - \Phi(x_k + d_k). \tag{3.9}$$

If $\|\tilde{d}_k\| > \|d_k\|$, set $\tilde{d}_k = 0$.

Step 1(ii$_2$). Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$\Phi(x_k + td_k + t^2\tilde{d}_k) \le \Phi(x_k) - \alpha t\langle d_k, H_k d_k\rangle.$$

Also, in Step 2 of Algorithm 2.1, x_{k+1} is set to $x_k + t_k d_k + t_k^2\tilde{d}_k$ and, if $t_k < 1$, $\bar{\omega}_k$ is any ω satisfying

$$\phi\left(x_k + \frac{t_k}{\beta}d_k + \left(\frac{t_k}{\beta}\right)^2\tilde{d}_k, \omega\right) > \Phi(x_k) - \alpha\frac{t_k}{\beta}\langle d_k, H_k d_k\rangle.$$

[4] Note that P_k remains invariant if in the definition of R_k the role of ω_1 is played by any other ω_i.

[5] Alternatively, $\tilde{d}$ could be selected as the solution of a linear least squares problem; see [28].

It is readily checked that such modification does not affect the analysis carried out in this section, the only necessary changes being to substitute, in the statements of Lemmas 3.1 and 6.1 and at various places in the proofs, all instances of $x_k + td_k$ with $x_k + td_k + t^2\tilde{d}_k$ (and similarly, *mutatis mutandis*, when t_k or $\frac{t_k}{\beta}$ is present instead of t; the fact that $\|\tilde{d}_k\| \leq \|d_k\|$ is enforced in the modified algorithm is a key to the validity of the modified proofs). Thus the modified algorithm will eventually behave as if solving (3.7) with, at each iteration, Ω_k selected to be equal to $\Omega_{\max}(x^*)$ and H_k normally updated. It is shown in [45] that, if (3.8) holds, the step $t_k = 1$ will always be accepted for k large enough. In fact the proof in [45] makes use of the weaker assumption

$$\lim_{k\to\infty} \frac{\|d_k^T P_k\{H_k - \nabla^2_{xx} L(x^*, \mu^*)\}P_k d_k\|}{\|d_k\|^2} = 0.$$

It is also shown in [45] that $\tilde{d}_k = O(\|d_k\|^2)$. It follows that 2-step superlinear convergence is preserved when (3.8) holds.

4. Implementation and numerical results. An efficient implementation of Algorithm 2.1, including the Maratos effect avoidance scheme described near the end of §3.2, has been developed as part of a C code dubbed CFSQP [22].[6] Version 2.0 of CFSQP was used to perform the numerical tests described next.

The specifics of the CFSQP implementation are as follows. In Algorithm 2.1, the rule for updating Ω_k specifies only that it must contain a certain subset of "critical" points of Ω. In practice, initial convergence is often sped up if additional "potentially critical" elements of Ω are also included. On the other hand, it is clear that increasing the size of Ω_k increases the number of gradient evaluations per iteration and makes $\mathrm{QP}(x_k, H_k, \Omega_k)$ more complex to solve. Thus a compromise must be struck. Various heuristics come to mind (see, e.g., [41]). The current CFSQP implementation focuses on the frequent case where "adjacent" objectives are closely related. (Objectives are "sequentially related.") It follows the idea used in [11], [27] of including in Ω_k the set $\Omega_\epsilon^{\ell\ell m}(x_k)$ of "ϵ-active left local maximizers" at x_k for some $\epsilon > 0$. A point $\omega_i \in \Omega := \{\omega_0, \ldots, \omega_q\}$ is ϵ-active if it belongs to

$$\Omega_\epsilon(x) = \{\omega_i : \phi(x, \omega_i) > \Phi(x) - \epsilon\}.$$

It is a left local maximizer of ϕ over Ω at x if one of the following three conditions holds: (i) $i \in \{1, \ldots, q-1\}$ and

$$\phi(x, \omega_i) > \phi(x, \omega_{i-1}) \tag{4.1}$$

and

$$\phi(x, \omega_i) \geq \phi(x, \omega_{i+1}); \tag{4.2}$$

(ii) $i = 0$ and (4.2); (iii) $i = q$ and (4.1). We also found that using $\Omega_0 = \Omega_{\max}(x_0)$ often gave a poor initial search direction and performance could be improved if additional points were heuristically selected for the first iteration. For many problems, the performance was improved if the end points ω_0 and ω_q were included in Ω_0. Thus, for Ω_0 and Ω_{k+1} (in Steps 0 and 2 of Algorithm 2.1), CFSQP selects, respectively,

$$\Omega_0 = \Omega_{\max}(x_0) \cup \Omega_\epsilon^{\ell\ell m}(x_0) \cup \{\omega_0\} \cup \{\omega_q\}$$

[6] CFSQP is available from the authors.

and

$$\Omega_{k+1} = \begin{cases} \Omega_{\max}(x_{k+1}) \cup \Omega_k^b \cup \Omega_\epsilon^{\ell\ell m}(x_{k+1}) & \text{if } t_k = 1, \\ \Omega_{\max}(x_{k+1}) \cup \Omega_k^b \cup \{\bar{\omega}_k\} \cup \Omega_\epsilon^{\ell\ell m}(x_{k+1}) & \text{if } t_k < 1. \end{cases}$$

Many problems encountered in practice involve more than one set of "sequentially related" objectives, e.g., a finely discretized version of the problem

$$\text{minimize } \max\left\{\sup_\omega \phi^1(x,\omega), \ldots, \sup_\omega \phi^\ell(x,\omega)\right\}, \tag{4.3}$$

where ω ranges over some interval. An important example of this type of problem is Chebyshev approximation, which has the form

$$\text{minimize } \sup_\omega |\phi(x,\omega)|$$

or, equivalently,

$$\text{minimize } \max\left\{\sup_\omega \phi(x,\omega), \sup_\omega -\phi(x,\omega)\right\}.$$

All but the last problem in the numerical tests discussed next are of this type. Note that Algorithm 2.1 and the analysis of §3 apply without modification to (4.3) by "linearly ordering" the discrete objectives as, say, $\phi^1(\cdot,\omega_0^1), \ldots, \phi^1(\cdot,\omega_{q_1}^1), \phi^2(\cdot,\omega_0^2), \ldots, \phi^\ell(\cdot,\omega_{q_\ell}^\ell)$, where it is assumed that the ith set contains q_i objective functions. However, in its selection of Ω_k, the CFSQP implementation takes into account the grouping of the objectives into subsets. Specifically, Ω_0 and Ω_k include the global maximizers and the ϵ-active left local maximizers for each of the ϕ^i's considered independently, and Ω_0 includes the "end points" for each of the ϕ^i's.

The following parameter values are used in CFSQP: $\alpha = 0.1$, $\beta = 0.5$, δ is the square root of the machine precision, and $\epsilon = 1$ (in $\Omega_\epsilon^{\ell\ell m}(x)$). For the solution of the QP subproblems, CFSQP invokes QLD, a code due to Powell and Schittkowski [40]. H_k is updated using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula with Powell's modification [35] with the following stipulations: The evaluation of the gradient of the Lagrangian function is based on the KKT multipliers corresponding to the QP subproblem and multipliers associated with values of ω not used in the QP are set to 0. Assigning the value 0 to multipliers associated with constraints not considered in the current QP subproblem is equivalent to considering them inactive, which is consistent with the intuition underlying the selection of Ω_k.

The numerical results reported next were obtained on discretized versions of nine test problems borrowed from the literature. Problems `OET 1` through `OET 7` are taken from [26], `HET-Z` from [19], and `PT` from [34]. Problems `OET 1` through `OET 7` and `HET-Z` are of the form

$$\text{minimize } \max_{\omega \in I} |\phi(x,\omega)|,$$

with ϕ and I as follows (the ξ's are the components of x):

`OET 1`: $\phi(x,\omega) = \omega^2 - (\xi_1\omega + \xi_2\exp(\omega))$, $I = [0,2]$, $x_0 = (1,1)$.
`OET 2`: $\phi(x,\omega) = \frac{1}{1+\omega} - \xi_1\exp(\xi_2\omega)$, $I = [-0.5,0.5]$, $x_0 = (1,-1)$.
`OET 3`: $\phi(x,\omega) = \sin(\omega) - (\xi_1 + \xi_2\omega + \xi_3\omega^2)$, $I = [0,1]$, $x_0 = (1,1,1)$.
`OET 4`: $\phi(x,\omega) = \exp(\omega) - \frac{\xi_1+\xi_2\omega}{1+\xi_3\omega}$, $I = [0,1]$, $x_0 = (1,1,1)$.
`OET 5`: $\phi(x,\omega) = \sqrt{\omega} - (\xi_4 - (\xi_1\omega^2 + \xi_2\omega + \xi_3)^2)$, $I = [0.25,1]$, $x_0 = (1,1,1,1)$.
`OET 6`: $\phi(x,\omega) = \frac{1}{1+\omega} - (\xi_1\exp(\xi_3\omega) + \xi_2\exp(\xi_4\omega))$, $I = [-0.5,0.5]$, $x_0 = (1,1,-3,-1)$.

OET 7: $\phi(x,\omega) = \frac{1}{1+\omega} - (\xi_1 \exp(\xi_4\omega) + \xi_2 \exp(\xi_5\omega) + \xi_3 \exp(\xi_6\omega))$, $I = [-0.5, 0.5]$, $x_0 = (1,1,1,-7,-3,-1)$.

HET-Z: $\phi(x,\omega) = (1-\omega^2) - (0.5x^2 - 2x\omega)$, $I = [-1,1]$, $x_0 = 1$.

Problem PT is of the form

$$\text{minimize } \max_{\omega \in I} \phi(x,\omega),$$

with $\phi(x,\omega) = (2\omega^2 - 1)x + \omega(1-\omega)(1-x)$, $I = [0,1]$, and $x_0 = 5$.

To assess the efficiency of the scheme proposed in this paper, we compared the CFSQP implementation of Algorithm 2.1 with two algorithms differing from it only in the selection of Ω_k at each iteration. In algorithm FULL, $\Omega_k = \Omega$ at each iteration, which essentially corresponds to Han's algorithm [14], [15]. In algorithm ϵ-ACT, a simple "ϵ-active" scheme is used, specifically, $\Omega_k = \Omega_\epsilon(x_k)$ for all k, with $\epsilon = 0.1$ (both for $+\phi$ and $-\phi$ in the case of the first eight problems). For all three algorithms, the optimization process was terminated whenever $\|d_k\| \leq$ 1.E-4 was achieved.

In Tables 1 and 2, results are reported for 101 and 501 uniformly spaced mesh points, respectively, (for a total of, respectively, 202 and 1002 "discrete objectives" in the case of the first eight problems); specifically,

$$\Omega = \left\{ a, a + \frac{b-a}{q}, a + \frac{2(b-a)}{q}, \ldots, b \right\},$$

with $q = 100$ and 500, respectively, where a and b are the end points of the interval of variation of ω for the problem under consideration. In the tables, NF is the number of evaluations of objective function ϕ,[7,8] IT is the total number of iterations, $\sum |\Omega_k|$ is the sum over all iterations k of the cardinality of Ω_k (in the case of NEW and FULL, it is equal to the total number of gradient evaluations), and $|\Omega^*|$ is the number of points in $\Omega_k = \Omega^*$ at the stopping point $x_k = x^*$. TIME indicates the execution time in seconds and OBJECTIVE the value of the objective function at x^*. All tests were conducted on a SUN/SPARC 1 workstation.

The following observations may be made. In most cases, the number of iterations and the total number of function evaluations are lowest for FULL and highest for NEW. This is expected, though, since the search directions in NEW are computed based on a much simpler QP model. Note, however, that the increase in the number of iterations and function evaluations when using NEW instead of FULL is typically moderate. In contrast, NEW provides dramatic savings in terms of the number of gradient evaluations and the size of the QP subproblems (whereas the savings achieved by ϵ-ACT are modest). Note, in particular, that $|\Omega^*|$ remains essentially unchanged when the number of mesh points is increased from 101 to 501. The decrease in computational effort achieved by NEW is clearly evident in the dramatically lower TIME of execution.

Finally note that, in the implementation used for these numerical tests, the QP solver does not take into account information from the solution of the previous QP subproblem when starting a new one. (QLD does not allow for such a "crash start.") One could argue that a crash start may significantly speed up the solution of QP

[7]For the first eight problems, all numbers in this column are pessimistic by a factor of about 2: the evaluation of $+\phi$ and $-\phi$ at a given point counts as two function evaluations.

[8]Note that NF is in general not a multiple of the cardinality of Ω. Indeed computation of $\tilde{d}_k$ involves evaluating $\phi(x_k + d_k, \omega)$ only for $\omega \in \Omega_k$ (see (3.9)).

TABLE 1
Numerical results with discretization $|\Omega| = 101$.

PROB	n	ALGO	NF	IT	$\sum\|\Omega_k\|$	$\|\Omega^*\|$	TIME	OBJECTIVE	$\|d^*\|$
OET 1	2	NEW	2546	10	56	6	0.88	0.53819574	0.38E-16
		FULL	1445	6	1212	202	1.54	0.53813894	0.97E-14
		ϵ-ACT	2444	8	560	92	1.63	0.53819574	0.38E-16
OET 2	2	NEW	861	4	22	6	0.53	0.08715336	0.12E-05
		FULL	642	3	606	202	1.05	0.08715640	0.47E-05
		ϵ-ACT	842	4	448	202	1.00	0.08716226	0.35E-04
OET 3	3	NEW	1805	7	47	8	0.96	0.00450481	0.51E-15
		FULL	1387	5	1010	202	2.31	0.00450481	0.11E-16
		ϵ-ACT	1905	7	988	202	1.81	0.00450481	0.19E-15
OET 4	3	NEW	2805	10	68	8	1.34	0.00429463	0.27E-11
		FULL	2472	9	1818	202	4.50	0.00429463	0.76E-11
		ϵ-ACT	3525	12	1494	202	4.51	0.00429463	0.37E-10
OET 5	4	NEW	6727	19	152	8	3.26	0.00264951	0.23E-05
		FULL	5533	18	3636	202	12.2	0.00264951	0.29E-06
		ϵ-ACT	7407	22	3360	202	13.0	0.00264951	0.24E-05
OET 6	4	NEW	4314	14	128	10	2.54	0.00206863	0.18E-09
		FULL	3765	12	2424	202	9.47	0.00206878	0.22E-06
		ϵ-ACT	4035	13	2376	202	10.2	0.00206880	0.39E-06
OET 7	6	NEW	38106	97	1186	12	28.6	0.00006644	0.98E-04
		FULL	11887	31	6262	202	55.8	0.00004432	0.18E-12
		ϵ-ACT	25974	68	13274	202	124.	0.00004432	0.32E-12
HET-Z	1	NEW	606	2	7	3	0.29	1.00000000	0.22E-15
		FULL	1437	7	1414	202	1.63	0.99995000	0
		ϵ-ACT	1010	4	194	64	0.56	0.99995000	0
PT	1	NEW	1224	7	18	2	0.35	0.23605381	0
		FULL	602	5	505	101	0.55	0.23605381	0
		ϵ-ACT	986	5	155	66	0.34	0.23605381	0

subproblems in algorithm FULL, its effect being akin to drastically reducing the number of constraints to be dealt with by the QP solver, and that, as a result, if a crash start were used, the computational cost of solving QP subproblems in FULL might be comparable to that of solving QP subproblems in NEW. To investigate this issue, we conducted additional tests with QPSOL [9] (which allows for crash starts) replacing QLD. It was observed that a crash start is helpful only in the final iterations, when the active set is correctly identified. Our interpretation of this phenomenon is that, in early iterations, although the good behavior of NEW indicates that the crash set is a reasonable approximation for the active set in the sense that there are values of ω in the crash set close to most values of ω in the true active set, there nevertheless may be very little (or none at all) overlap between these two sets. As a result, a crash set may be of no use to an off-the-shelf QP solver. Overall, the QP-solving time in NEW is still significantly lower than the QP solving time in FULL.

5. Conclusion. An SQP-type algorithm has been proposed and analyzed for the solution of minimax optimization problems with many more objective functions than variables, in particular, finely discretized continuous minimax problems. (It has been argued that SQP-type algorithms are particularly suited to certain classes of such problems.) At each iteration, a QP problem involving only a small set of constraints is solved and, correspondingly, only a few gradients are evaluated. Numerical results indicate that the proposed scheme is efficient.

There is no conceptual difficulty in extending the algorithm to handle discretized versions of continuous minimax problems where the maximization is with respect to

TABLE 2
Numerical results with discretization $|\Omega| = 501$.

PROB	n	ALGO	NF	IT	$\sum\|\Omega_k\|$	$\|\Omega^*\|$	TIME	OBJECTIVE	$\|d^*\|$
OET 1	2	NEW	13420	11	62	6	4.17	0.53824312	0.38E-16
		FULL	7153	6	6012	1002	10.6	0.53824312	0.97E-13
		ϵ-ACT	11919	8	2748	448	7.46	0.53824312	0.48E-13
OET 2	2	NEW	4207	4	23	6	1.61	0.08716106	0.15E-05
		FULL	3155	3	3006	1002	4.40	0.08716395	0.47E-05
		ϵ-ACT	4128	4	2212	1002	3.89	0.08716768	0.31E-04
OET 3	3	NEW	8920	7	50	9	3.25	0.00450552	0.25E-05
		FULL	6873	5	5010	1002	9.95	0.00450505	0.46E-17
		ϵ-ACT	9829	7	4904	1002	12.2	0.00450505	0.17E-15
OET 4	3	NEW	13886	10	71	9	4.78	0.00429567	0.12E-06
		FULL	11998	9	9018	1002	21.6	0.00429543	0.76E-11
		ϵ-ACT	17161	12	7404	1002	21.5	0.00429543	0.32E-10
OET 5	4	NEW	33441	19	158	8	12.2	0.00265008	0.27E-5
		FULL	27460	18	18036	1002	59.6	0.00265008	0.30E-06
		ϵ-ACT	42085	25	19694	1002	57.4	0.00265008	0.16E-04
OET 6	4	NEW	21345	14	131	11	8.63	0.00206998	0.16E-05
		FULL	18625	12	12024	1002	50.8	0.00206989	0.22E-06
		ϵ-ACT	19995	13	11856	1002	52.2	0.00206996	0.35E-06
OET 7	6	NEW	54584	30	355	15	27.2	0.00013273	0.32E-04
		FULL	60521	32	32064	1002	273.	0.00004446	0.11E-12
		ϵ-ACT	127096	67	64806	1002	383.	0.00006876	0.77E-13
HET-Z	1	NEW	3006	2	7	3	0.89	1.00000000	0.22E-13
		FULL	10062	10	10020	1002	10.9	0.99999800	0.14E-13
		ϵ-ACT	7092	7	1910	316	3.35	0.99999800	0.14E-13
PT	1	NEW	7337	8	22	2	1.03	0.23606791	0
		FULL	2991	5	2505	501	1.84	0.23606792	0
		ϵ-ACT	3895	5	799	334	0.90	0.23606792	0

several free variables ranging over arbitrary compact sets. The proposed algorithm, with appropriate modifications, has been implemented in an optimization-based design package [7] and has proven very successful in solving various types of engineering design problems.

6. Appendix: Proofs.

6.1. Proof of Lemma 3.6. We denote by $H_k^{\frac{1}{2}}$ the symmetric positive definite square root of H_k, by $H_k^{-\frac{1}{2}}$ its inverse, and we make use of the following notation:

$$\begin{aligned}
\gamma_k(\omega) &= \Phi(x_k) - \phi(x_k, \omega), \\
\pi_k &= \sum_{\omega \in \Omega_k} \mu_{k,\omega} \gamma_k(\omega), \\
\pi_k^+ &= \sum_{\omega \in \Omega_k} \mu_{k,\omega} \gamma_{k+1}(\omega), \\
g_k(\omega) &= H_k^{-\frac{1}{2}} \nabla_x \phi(x_k, \omega), \\
p_k &= \sum_{\omega \in \Omega_k} \mu_{k,\omega} g_k(\omega) = -H_k^{\frac{1}{2}} d_k, \\
p_k^+ &= \sum_{\omega \in \Omega_k} \mu_{k,\omega} g_{k+1}(\omega).
\end{aligned}$$

It follows from (3.1), the boundedness of $\{\mu_k\}$, and Assumptions 1, 2, and 3 that, for

some $M > 1$,

$$\max\{|v_k|, \|p_k^+\|, \|g_{k+1}(\bar{\omega})\|\} \leq M \quad \forall k \in \mathbb{N}, \quad \forall \omega \in \Omega. \tag{6.1}$$

We will show that (3.3) holds with

$$c = \frac{(1-2\tilde{\alpha})^2}{16M^2}, \tag{6.2}$$

where $\tilde{\alpha}$ is any number in $(\alpha, \frac{1}{2})$.

A few more lemmas are first established.

LEMMA 6.1. *Let K be an infinite index set such that $\{\|H_{k+1} - H_k\|\}$ converges to zero on K and $\{d_k\}$ is bounded away from zero on K. Then, given any $\tilde{\alpha} > \alpha$,*

$$\phi(x_{k+1}, \omega) + \langle \nabla_x \phi(x_{k+1}, \omega), H_{k+1}^{-\frac{1}{2}} H_k^{\frac{1}{2}} d_k \rangle - \Phi(x_{k+1}) \geq -\tilde{\alpha} \langle d_k, H_k d_k \rangle$$

whenever $t_k < 1$, for $k \in K$, k large enough, and for all $\omega \in \Omega$ such that

$$\phi\left(x_k + \frac{t_k}{\beta} d_k, \omega\right) > \Phi(x_k) - \alpha \frac{t_k}{\beta} \langle d_k, H_k d_k \rangle.$$

Proof. In view of Lemma 3.3(iv), $\{t_k\}$ converges to zero on K. Proceeding by contradiction, suppose the claim does not hold, i.e., there exists an infinite index set $K' \subseteq K$ such that, for all $k \in K'$, $t_k < 1$ and for some $\bar{\omega}_k \in \Omega$

$$\phi\left(x_k + \frac{t_k}{\beta} d_k, \bar{\omega}_k\right) > \Phi(x_k) - \alpha \frac{t_k}{\beta} \langle d_k, H_k d_k \rangle \tag{6.3}$$

and

$$\phi(x_{k+1}, \bar{\omega}_k) + \langle \nabla_x \phi(x_{k+1}, \bar{\omega}_k), H_{k+1}^{-\frac{1}{2}} H_k^{\frac{1}{2}} d_k \rangle - \Phi(x_{k+1}) < -\tilde{\alpha} \langle d_k, H_k d_k \rangle. \tag{6.4}$$

In view of Lemma 3.3(i) and (ii) and Assumption 3, there exists an infinite index set $K'' \subseteq K'$ such that the sequences $\{x_k\}$, $\{d_k\}$, $\{H_k\}$, and $\{\bar{\omega}_k\}$ converge on K'', respectively, to some x^*, d^*, H^*, and ω^*. In view of Lemma 3.3(iv), $\{x_{k+1}\}$ also converges to x^* on K''. Furthermore, since $\{t_k\}$ goes to zero on K'', it follows from (6.3) and our continuity assumption that $\omega^* \in \Omega_{\max}(x^*)$. Also, a simple contradiction argument using Assumption 3 shows that $\{H_{k+1}^{-\frac{1}{2}} H_k^{\frac{1}{2}}\} \to I$ on K. Thus taking the limit of (6.4) on K'' yields

$$\langle \nabla_x \phi(x^*, \omega^*), d^* \rangle \leq -\tilde{\alpha} \langle d^*, H^* d^* \rangle < -\alpha \langle d^*, H^* d^* \rangle. \tag{6.5}$$

On the other hand, (6.3) implies that

$$\phi\left(x_k + \frac{t_k}{\beta} d_k, \bar{\omega}_k\right) > \phi(x_k, \bar{\omega}_k) - \alpha \frac{t_k}{\beta} \langle d_k, H_k d_k \rangle.$$

Thus,

$$\frac{\phi(x_k + \frac{t_k}{\beta} d_k, \bar{\omega}_k) - \phi(x_k, \bar{\omega}_k)}{\frac{t_k}{\beta}} > -\alpha \langle d_k, H_k d_k \rangle. \tag{6.6}$$

Taking the limit of (6.6) as $k \to \infty$ on K'' yields

$$\langle \nabla_x \phi(x^*, \omega^*), d^* \rangle \geq -\alpha \langle d^*, H^* d^* \rangle,$$

which contradicts (6.5). □

As in [21, Chap. 3], using the dual of $\mathrm{QP}(x_k, H_k, \Omega_k)$ facilitates the analysis.

LEMMA 6.2. *Given any $x \in R^n$, $H = H^T > 0$, and $\hat{\Omega} \subseteq \Omega$, the dual quadratic program $\overline{\mathrm{QP}}(x, H, \hat{\Omega})$ of $\mathrm{QP}(x, H, \hat{\Omega})$ is given by*

$$\overline{\mathrm{QP}}(x, H, \hat{\Omega}) \qquad \textit{maximize} \quad -\left(\frac{1}{2}\left\|\sum_{\omega \in \hat{\Omega}} \mu_\omega g(\omega)\right\|^2 + \sum_{\omega \in \hat{\Omega}} \mu_\omega \gamma(\omega)\right) \textit{s.t.} \quad \mu \in U,$$

where $\gamma(\omega) = \Phi(x) - \phi(x, \omega)$, $g(\omega) = H^{-\frac{1}{2}} \nabla_x \phi(x, \omega)$, and

$$U = \left\{ \mu \in I\!R^{|\hat{\Omega}|} : \sum_{\omega \in \hat{\Omega}} \mu_\omega = 1 \quad \textit{and} \quad \mu_\omega \geq 0 \ \forall \omega \in \hat{\Omega} \right\}.$$

Proof. The dual is given by

$$\text{maximize } \varphi(\mu) \quad \text{s.t. } \mu \in U,$$

where φ is the dual functional, i.e.,

$$\varphi(\mu) = \min_d \left\{ \frac{1}{2}\langle d, Hd \rangle + \sum_{\omega \in \hat{\Omega}} \mu_\omega (\phi(x, \omega) + \langle \nabla_x \phi(x, \omega), d \rangle) - \Phi(x) \right\}. \tag{6.7}$$

In view of Assumption 3, the unique minimizer d^* in (6.7) is given by

$$d^* = -H^{-1} \sum_{\omega \in \hat{\Omega}} \mu_\omega \nabla_x \phi(x, \omega) = -H^{-\frac{1}{2}} \sum_{\omega \in \hat{\Omega}} \mu_\omega g(\omega),$$

yielding

$$\sum_{\omega \in \hat{\Omega}} \mu_\omega \langle \nabla_x \phi(x, \omega), d^* \rangle = -\langle d^*, Hd^* \rangle.$$

Therefore,

$$\begin{aligned} \varphi(\mu) &= -\frac{1}{2}\langle d^*, Hd^* \rangle - \sum_{\omega \in \hat{\Omega}} \mu_\omega \{\Phi(x) - \phi(x, \omega)\} \\ &= -\left(\frac{1}{2}\left\|\sum_{\omega \in \hat{\Omega}} \mu_\omega g(\omega)\right\|^2 + \sum_{\omega \in \hat{\Omega}} \mu_\omega \gamma(\omega)\right) \end{aligned}$$

and the result follows. □

LEMMA 6.3. *There exists $\underline{t} > 0$ such that*

$$\phi(x_k + t d_k, \omega) \leq \Phi(x_k) - \alpha t \langle d_k, H_k d_k \rangle$$

for all k, all $t \in [0, \underline{t}]$, and all $\omega \in \Omega_k$.

Proof. In view of Assumption 1′ and the boundedness of $\{d_k\}$, there exist $c_1 > 0$ and $c_2 > 0$ such that, for all $\omega \in \Omega$, all $t \in [0,1]$, and all k,

$$\phi(x_k + td_k, \omega) \le \phi(x_k, \omega) + c_1 t \|d_k\|$$

and

$$\phi(x_k + td_k, \omega) \le \phi(x_k, \omega) + t\langle \nabla_x \phi(x_k, \omega), d_k\rangle + c_2 t^2 \|d_k\|^2.$$

Thus, it follows from (2.2) applied to $\mathrm{QP}(x_k, H_k, \Omega_k)$ that, for all $\omega \in \Omega_k$, all $t \in [0,1]$, and all k,

$$\begin{aligned}
&\phi(x_k + td_k, \omega) \\
&\quad \le (1-t)\phi(x_k,\omega) + t\{\phi(x_k,\omega) + \langle \nabla_x \phi(x_k,\omega), d_k\rangle\} + c_2 t^2 \|d_k\|^2 \\
&\quad \le (1-t)\phi(x_k,\omega) + t \max_{\omega\in\Omega_k}\{\phi(x_k,\omega) + \langle \nabla_x \phi(x_k,\omega), d_k\rangle\} + c_2 t^2 \|d_k\|^2 \\
&\quad = (1-t)\phi(x_k,\omega) + t \sum_{\omega\in\Omega_k} \mu_{k,\omega}\{\phi(x_k,\omega) + \langle \nabla_x \phi(x_k,\omega), d_k\rangle\} + c_2 t^2 \|d_k\|^2 \\
&\quad \le (1-t)\Phi(x_k) + t\Phi(x_k) \sum_{\omega\in\Omega_k} \mu_{k,\omega} + t \sum_{\omega\in\Omega_k} \mu_{k,\omega}\langle \nabla_x \phi(x_k,\omega), d_k\rangle + c_2 t^2 \|d_k\|^2 \\
&\quad = \Phi(x_k) - t\langle d_k, H_k d_k\rangle + c_2 t^2 \|d_k\|^2,
\end{aligned}$$

where again $\mu_{k,\omega}$, $\omega \in \Omega_k$ are the KKT multipliers associated with $\mathrm{QP}(x_k, H_k, \Omega_k)$. Thus, in view of Assumption 3, since $\alpha \in (0, \frac{1}{2})$,

$$\begin{aligned}
\phi(x_k + td_k, \omega) &\le \Phi(x_k) - \alpha t\langle d_k, H_k d_k\rangle + t(\alpha - 1)\langle d_k, H_k d_k\rangle + c_2 t^2 \|d_k\|^2 \\
&\le \Phi(x_k) - \alpha t\langle d_k, H_k d_k\rangle + t\|d_k\|^2\{(\alpha-1)\sigma_1 + c_2 t\} \\
&\le \Phi(x_k) - \alpha t\langle d_k, H_k d_k\rangle \quad \forall t \in [0, \bar{t}],
\end{aligned}$$

for all $\omega \in \Omega_k$, with $\bar{t} = (1-\alpha)\sigma_1/c_2 > 0$. □

Proof of Lemma 3.6. Since $\{d_k\}$ is bounded away from zero on K, it follows from Lemma 3.3(iv) that $\{t_k\}$ goes to zero on K. Without loss of generality assume that $t_k < 1$ for all $k \in K$. For each $k \in K$, let $\bar{\omega}_k \in \Omega_{k+1}$ be the value picked in Step 2 of Algorithm 2.1; $\bar{\omega}_k$ satisfies

$$\phi\left(x_k + \frac{t_k}{\beta} d_k, \bar{\omega}_k\right) > \Phi(x_k) - \alpha \frac{t_k}{\beta}\langle d_k, H_k d_k\rangle.$$

In view of Lemma 6.3, $\bar{\omega}_k \notin \Omega_k$ for all $k \in K$. Define $\Omega'_k = \Omega_k^b \cup \{\bar{\omega}_k\}$. Let v'_{k+1} denote the optimal value of $\mathrm{QP}(x_{k+1}, H_{k+1}, \Omega'_k)$. In view of the construction of Ω_{k+1}, $\Omega'_k \subseteq \Omega_{k+1}$. Thus, $|v_{k+1}| \le |v'_{k+1}|$. Therefore, it suffices to prove (3.3) with the left-hand side replaced by $|v'_{k+1}|$. Define the quadratic function in ν:

$$\begin{aligned}
Q(\nu) &= \frac{1}{2}\left\| \nu g_{k+1}(\bar{\omega}_k) + (1-\nu) \sum_{\omega\in\Omega_k} \mu_{k,\omega} g_{k+1}(\omega) \right\|^2 + \nu\gamma_{k+1}(\bar{\omega}_k) \\
&\qquad + (1-\nu) \sum_{\omega\in\Omega_k} \mu_{k,\omega}\gamma_{k+1}(\omega) \\
&= \frac{1}{2}\|\nu g_{k+1}(\bar{\omega}_k) + (1-\nu)p_k^+\|^2 + \nu\gamma_{k+1}(\bar{\omega}_k) + (1-\nu)\pi_k^+.
\end{aligned}$$

Let $\nu \in [0,1]$. Let $\mu_{k,\omega}, \omega \in \Omega_k$ be the KKT multipliers associated with $\mathrm{QP}(x_k, H_k, \Omega_k)$. With the (dual feasible) choice $\mu_{\bar{\omega}_k} = \nu$, $\mu_\omega = (1-\nu)\mu_{k,\omega}$ for all $\omega \in \Omega_k^b$, and $\mu_\omega = 0$ for all $\omega \in \Omega \backslash \Omega_k'$, the objective of the dual quadratic program $\overline{\mathrm{QP}}(x_{k+1}, H_{k+1}, \Omega_k')$ takes value $-Q(\nu)$. By duality, v_{k+1}' is the optimal objective value for both $\mathrm{QP}(x_{k+1}, H_{k+1}, \Omega_k')$ and $\overline{\mathrm{QP}}(x_{k+1}, H_{k+1}, \Omega_k')$. Thus,

$$|v_{k+1}'| \leq Q(\nu) \quad \forall \nu \in [0,1].$$

Thus, it suffices to prove (3.3) with the left-hand side replaced by $\min_{\nu\in[0,1]} Q(\nu)$. Expanding the quadratic term of $Q(\nu)$ yields

$$\begin{aligned} Q(\nu) &= \frac{1}{2}\nu^2 \|g_{k+1}(\bar{\omega}_k)\|^2 + \frac{1}{2}(1-\nu)^2\|p_k^+\|^2 + \nu(1-\nu)\langle g_{k+1}(\bar{\omega}_k), p_k^+\rangle \\ &\quad + \nu\gamma_{k+1}(\bar{\omega}_k) + (1-\nu)\pi_k^+ \\ &= \frac{1}{2}\|p_k^+\|^2 + \frac{\nu^2}{2}\|g_{k+1}(\bar{\omega}_k) - p_k^+\|^2 + \nu\langle g_{k+1}(\bar{\omega}_k), p_k^+\rangle + \nu\gamma_{k+1}(\bar{\omega}_k) \\ &\quad - \nu\|p_k^+\|^2 + (1-\nu)\pi_k^+. \end{aligned} \tag{6.8}$$

Note that

$$\langle g_{k+1}(\bar{\omega}_k), p_k\rangle = -\langle \nabla_x \phi(x_{k+1}, \bar{\omega}_k), H_{k+1}^{-\frac{1}{2}} H_k^{\frac{1}{2}} d_k\rangle.$$

Since $\{t_k\}$ converges to zero on K, $t_k < \delta$ for $k \in K$, k large enough. Since $\bar{\omega}_k \notin \Omega_k$, it follows from Step 2 in Algorithm 2.1 that $H_{k+1} = H_k$ for $k \in K$, k large enough. Thus, the assumptions of Lemma 6.1 are all satisfied. Given $\tilde{\alpha} \in (\alpha, \frac{1}{2})$, in view of Lemma 6.1 with $\omega = \bar{\omega}_k$, there exists an integer k_1 such that, for all $k \geq k_1$, $k \in K$,

$$\langle g_{k+1}(\bar{\omega}_k), p_k\rangle + \gamma_{k+1}(\bar{\omega}_k) \leq \gamma_{k+1}(\bar{\omega}_k) + \phi(x_{k+1}, \bar{\omega}_k) - \Phi(x_{k+1}) + \tilde{\alpha}\langle d_k, H_k d_k\rangle.$$

In view of the definition of γ_k and of relationships (3.1) and (3.2), it follows that

$$\langle g_{k+1}(\bar{\omega}_k), p_k\rangle + \gamma_{k+1}(\bar{\omega}_k) \leq -2\tilde{\alpha} v_k \quad \forall k \in K, k \geq k_1.$$

Hence, for all $k \geq k_1$, $k \in K$,

$$\langle g_{k+1}(\bar{\omega}_k), p_k^+\rangle + \gamma_{k+1}(\bar{\omega}_k) \leq -2\tilde{\alpha} v_k - \langle g_{k+1}(\bar{\omega}_k), p_k - p_k^+\rangle.$$

Also,

$$\begin{aligned} \|p_k^+\|^2 &= \|p_k - p_k + p_k^+\|^2 \\ &= \|p_k\|^2 + \|p_k - p_k^+\|^2 - 2\langle p_k, p_k - p_k^+\rangle \\ &= \|p_k\|^2 + O(\|p_k - p_k^+\|). \end{aligned}$$

On the other hand, since $M > 1$, inequality (6.1) implies that

$$\|g_{k+1}(\bar{\omega}_k) - p_k^+\|^2 \leq 4M^2.$$

Substituting all these into (6.8) yields, for all $k \geq k_1$, $k \in K$, $\nu \in [0,1]$,

$$\begin{aligned} Q(\nu) &\leq \frac{1}{2}\|p_k\|^2 + 2M^2\nu^2 - \nu(2\tilde{\alpha} v_k + \|p_k\|^2) + (1-\nu)\pi_k^+ + O(\|p_k - p_k^+\|) \\ &= \frac{1}{2}\|p_k\|^2 + \pi_k + 2M^2\nu^2 - \nu(2\tilde{\alpha} v_k + \|p_k\|^2 + \pi_k) - (1-\nu)(\pi_k - \pi_k^+) \\ &\quad + O(\|p_k - p_k^+\|). \end{aligned}$$

In view of Lemma 6.2, since duality holds,

$$|v_k| = -v_k = \frac{1}{2}\|p_k\|^2 + \pi_k.$$

Thus, for all $k \geq k_1$, $k \in K$,

$$\begin{aligned} Q(\nu) &\leq |v_k| + 2M^2\nu^2 - \nu(2\tilde{\alpha}v_k + |v_k|) - \frac{\nu}{2}\|p_k\|^2 \\ &\quad + O(|\pi_k - \pi_k^+|) + O(\|p_k - p_k^+\|) \\ &\leq |v_k| + 2M^2\nu^2 - \nu(1-2\tilde{\alpha})|v_k| + O(|\pi_k - \pi_k^+|) + O(\|p_k - p_k^+\|). \end{aligned}$$

The minimum of the right-hand side is achieved at $\bar{\nu}_k = 4c|v_k|$, with $c = (1 - 2\tilde{\alpha})/16M^2$. Since $\tilde{\alpha} < \frac{1}{2}$, $|v_k| \leq M$, and $M > 1$, it follows that $\bar{\nu}_k \in [0,1]$ and thus, for all $k \geq k_1$, $k \in K$,

$$\min_{\nu\in[0,1]} Q(\nu) \leq Q(\bar{\nu}_k) \leq |v_k| - 32M^2c^2|v_k|^2 + O(|\pi_k - \pi_{k^+}|) + O(\|p_k - p_k^+\|).$$

Now, in view of Lemma 3.3(i) and (iv) and Assumption 1 and since $H_{k+1} = H_k$ for k large enough, $k \in K$, $\{\pi_k - \pi_k^+\}$ and $\{p_k - p_k^+\}$ both tend to zero as k goes to infinity, $k \in K$. Thus, since v_k is bounded away from zero on K (Lemma 3.3(i)), there exists a positive integer N such that, for all $k \geq N$, $k \in K$,

$$\min_{\nu\in[0,1]} Q(\nu) \leq |v_k| - c|v_k|^2.$$

Therefore (3.3) follows from the inequality

$$|v_{k+1}| \leq |v'_{k+1}| \leq \min_{\nu\in[0,1]} Q(\nu). \qquad \square$$

6.2. Proof of Lemma 3.8. $z_0 - \eta z_0^2$ achieves its largest value with $z_0 = \frac{1}{2\eta}$, yielding a largest possible value for z_1 given by

$$z_1 = \frac{1}{2\eta} - \eta\left(\frac{1}{2\eta}\right)^2 = \frac{1}{4\eta}.$$

The mapping

$$z \mapsto z - \eta z^2$$

is monotonic increasing over $[0, \frac{1}{2\eta}]$. Thus, given any z_0, the sequence defined by

$$\begin{aligned} z_1 &= \frac{1}{4\eta}, \\ z_{i+1} &= z_i - \eta z_i^2, \quad i = 1, 2, \ldots \end{aligned}$$

is the largest of all nonnegative sequences satisfying the given inequality condition, in the sense that given any such sequence $\{y_i\}$,

$$y_i \leq z_i, \quad i = 1, 2, \ldots.$$

Now let i_0 be such that $z_i < \epsilon$ for all $i \geq i_0$. It follows that $y_i < \epsilon$ for all $i \geq i_0$. $\square$

6.3. Proof of Lemma 3.10. We first show that $(x^*, \Phi(x^*), \mu^*)$ is an isolated solution of the nonlinear system of equations in (x, x^0, μ) (see (2.1)):

$$\begin{cases} \sum_{\omega\in\Omega} \mu_\omega \nabla_x \phi(x,\omega) = 0, \\ \sum_{\omega\in\Omega} \mu_\omega = 1, \\ \mu_\omega(\phi(x,\omega) - x^0) = 0 \quad \forall \omega \in \Omega. \end{cases} \tag{6.9}$$

This will result from the inverse function theorem if the Jacobian of the left-hand side of this system of equations is nonsingular. To show that this is indeed the case, let $(d, d^0, \lambda) \in R^n \times R \times R^{|\Omega|}$ be such that

$$\begin{bmatrix} \nabla^2_{xx} L(x^*,\mu^*) & 0 & \nabla_x\phi(x^*,\omega_1) & \dots & \nabla_x\phi(x^*,\omega_\ell) \\ 0 & 0 & 1 & \dots & 1 \\ \mu^*_{\omega_1}\nabla_x\phi(x^*,\omega_1)^T & -\mu^*_{\omega_1} & \phi(x^*,\omega_1)-\Phi(x^*) & & \\ \vdots & \vdots & & \ddots & \\ \mu^*_{\omega_\ell}\nabla_x\phi(x^*,\omega_\ell)^T & -\mu^*_{\omega_\ell} & & & \phi(x^*,\omega_\ell)-\Phi(x^*) \end{bmatrix} \begin{bmatrix} d \\ d^0 \\ \\ \lambda \\ \\ \end{bmatrix} = 0,$$

where $\omega_1, \dots, \omega_\ell$ are the elements of Ω, i.e., suppose that

$$\nabla^2_{xx} L(x^*,\mu^*)d + \sum_{\omega\in\Omega} \lambda_\omega \nabla_x \phi(x^*,\omega) = 0, \tag{6.10}$$

$$\sum_{\omega\in\Omega} \lambda_\omega = 0, \tag{6.11}$$

$$\mu^*_\omega \langle \nabla_x \phi(x^*,\omega), d\rangle - \mu^*_\omega d^0 + \lambda_\omega(\phi(x^*,\omega) - \Phi(x^*)) = 0 \ \forall\ \omega \in \Omega. \tag{6.12}$$

From (6.12) and the last line in (2.1) it follows that

$$\lambda_\omega = 0 \quad \forall\omega \quad \text{s.t.} \quad \phi(x^*,\omega) < \Phi(x^*). \tag{6.13}$$

Together with (6.12) this implies that

$$\mu^*_\omega(\langle \nabla_x \phi(x^*,\omega), d\rangle - d^0) = 0 \quad \forall\omega \in \Omega$$

and, in view of the strict complementarity assumption,

$$\langle \nabla_x \phi(x^*,\omega), d\rangle = d^0 \quad \forall\omega \in \Omega_{\max}(x^*).$$

From the first two lines in (2.1), it follows that $d^0 = 0$; thus

$$\langle \nabla_x \phi(x^*,\omega), d\rangle = 0 \quad \forall\omega \in \Omega_{\max}(x^*). \tag{6.14}$$

Performing the inner product of both sides of (6.10) with d and using (6.13) and (6.14) yields

$$\langle d, \nabla^2_{xx} L(x^*,\mu^*)d\rangle = 0.$$

In view of Assumption 5, this together with (6.14) implies that $d = 0$. Substituting this in (6.10) and using (6.11)–(6.13) and Assumption 4, we conclude that $\lambda_\omega = 0$ for all $\omega \in \Omega_{\max}(x^*)$. Thus $(d, d^0, \lambda) = 0$ and the Jacobian of the left-hand side of (6.9) at $(x^*, \Phi(x^*), \mu^*)$ is nonsingular. Thus $(x^*, \Phi(x^*), \mu^*)$ is an isolated solution of (6.9) and therefore (x^*, μ^*) is an isolated KKT pair for (P). To complete the proof, we now proceed by contradiction. Thus suppose that there exists a sequence $\{x_k^*\}$ of KKT points for (P) that converges to x^* as k goes to infinity, with $x_k^* \neq x^*$ for all k, and let μ_k^* be a KKT multiplier vector associated with x_k^*. Since (x^*, μ^*) is an isolated KKT pair for (P), $\{\mu_k^*\}$ must be bounded away from μ^*. Since $\{\mu_k^*\}$ is bounded, we may assume without loss of generality that $\{\mu_k^*\}$ converges to $\hat{\mu}$ as k goes to infinity, with $\hat{\mu} \neq \mu^*$. In view of our continuity assumption, taking the limit of both sides of (2.1) as k goes to infinity shows that $(x^*, \hat{\mu})$ is a KKT pair for (P), in contradiction to the uniqueness of the KKT multiplier vector associated with x^*, which is guaranteed by Assumption 4.

Acknowledgments. The authors wish to thank Craig T. Lawrence for putting together the CFSQP implementation of the scheme proposed in this paper and for running the numerical tests and two referees for their insightful criticism and suggestions.

REFERENCES

[1] M. C. Biggs, *Constrained minimization using recursive equality quadratic programming*, in Numerical Methods for Non-Linear Optimization, F. A. Lootsma, ed., Academic Press, London, New York, 1972, pp. 411–428.

[2] S. P. Boyd and C. H. Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[3] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, *A survey of optimization techniques for integrated circuit design*, IEEE Proc., 69 (1981), pp. 1334–1362.

[4] A. R. Conn and Y. Li, *A structure-exploiting algorithm for nonlinear minimax problems*, SIAM J. Optim., 2 (1992), pp. 242–263.

[5] I. D. Coope and G. A. Watson, *A projected Lagrangian algorithm for semi-infinite programming*, Math. Programming, 32 (1985), pp. 337–356.

[6] H. A. Eschenauer, C. Mattheck, and N. Olhoff, eds., *Engineering Optimization in Design Processes*, Springer-Verlag, Berlin, 1991.

[7] M. K. H. Fan, A. L. Tits, J. L. Zhou, L. S. Wang, and J. Koninckx, *CONSOL-OPTCAD User's Manual (Version c1.2-s1.7, Released 8/1992)*, Tech. report 87-212r4, Institute for Systems Research, University of Maryland, College Park, MD, 1992.

[8] A. V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, in Academic Press, New York, London, R. Bellman, ed., Mathematics in Science and Engineering 165, 1983.

[9] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for QPSOL (Version 3.2): A Fortran Package for Quadratic Programming*, Tech. report SOL 84-6, Systems Optimization Laboratory, Stanford University, Stanford, CA, 1984.

[10] C. Gonzaga and E. Polak, *On constraint dropping schemes and optimality functions for a class of outer approximations algorithms*, SIAM J. Control Optim., 17 (1979), pp. 477–493.

[11] C. Gonzaga, E. Polak, and R. Trahan, *An improved algorithm for optimization problems with functional inequality constraints*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 49–54.

[12] G. Gramlich, R. Hettich, and E. W. Sachs, *Local convergence of SQP methods in semi-infinite programming*, SIAM J. Optim., 5 (1995), pp. 641–658.

[13] S. A. Gustafson, *A three-phase algorithm for semi-infinite programs*, in Semi-Infinite Programming and Applications, A. V. Fiacco and K. O. Kortanek, eds., Lecture Notes in Economics and Mathematical Systems 215, Springer-Verlag, New York, Heidelberg, Berlin, 1983, pp. 138–157.

[14] S. P. Han, *Superlinear Convergence of a Minimax Method*, Tech. report 78-336, Department of Computer Science, Cornell University, Ithaca, NY, 1978.

[15] S. P. HAN, *Variable metric methods for minimizing a class of nondifferentiable functions*, Math. Programming, 20 (1981), pp. 1–13.

[16] R. HETTICH, *An implementation of a discretization method for semi-infinite programming*, Math. Programming, 34 (1986), pp. 354–361.

[17] R. HETTICH AND W. VAN HONSTEDE, *On quadratically convergent methods for semi-infinite programming*, in Semi-Infinite Programming, R. Hettich, ed., Lecture Notes in Control and Information Sciences 15, Springer-Verlag, Berlin, New York, 1979, pp. 97–111.

[18] R. HETTICH AND K. O. KORTANEK, *Semi-infinite programming: Theory, methods, and applications*, SIAM Rev., 35 (1993), pp. 380–429.

[19] R. HETTCIH AND P. ZENCKE, *Numerische Methoden der Approximation und Semi-Infiniten Optimierung*, Teubner Studienbücher, Leipzig, Stuttgart, 1982.

[20] W. VAN HONSTEDE, *An approximation method for semi-infinite problems*, in Semi-Infinite Programming, R. Hettich, ed., Lecture Notes in Control and Information Sciences 15, Springer-Verlag, Berlin, New York, 1979, pp. 126–136.

[21] K. C. KIWIEL, *Methods of Descent in Nondifferentiable Optimization*, in Lecture Notes in Mathematics 1133, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985.

[22] C. T. LAWRENCE, J. L. ZHOU AND A. L. TITS, *User's Guide for CFSQP Version* 2.0: *A C Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*, Tech. report 94-16, Institute for Systems Research, University of Maryland, College Park, MD, 1994.

[23] C. LEMARÉCHAL, *Nondifferentiable optimization*, in Optimization, G. Nemhauser, A. Rinnooy-Kan, and M. Todd, eds., Handbooks in Operations Research and Management Science, North-Holland, Amsterdam, 1989.

[24] H. MINE, M. FUKUSHIMA, AND Y. TANAKA, *On the use of ϵ-most-active constraints in an exact penalty function method for nonlinear optimization*, IEEE Trans. Automat. Control, AC-29 (1984), pp. 1040–1042.

[25] C. A. MOTA SOARES, ED., *Computer-Aided Optimal Design: Structural and Mechanical Systems*, NATO ASI Series, Series F, 27, Springer-Verlag, Berlin, 1987.

[26] K. OETTERSHAGEN, *Ein Superlinear Konvergenter Algorithmus zur Losung Semi-Infiniter Optimierungsprobleme*, Ph.D. thesis, Bonn University, Bonn, Germany, 1982.

[27] E. R. PANIER AND A. L. TITS, *A globally convergent algorithm with adaptively refined discretization for semi-infinite optimization problems arising in engineering design*, IEEE Trans. Automat. Control, AC-34 (1989), pp. 903–908.

[28] ———, *A superlinearly convergent method of feasible directions for optimization problems arising in the design of engineering systems*, in Proc. 7th International Conf. on Analysis and Optimization of Systems, Antibes, 1986, A. Bensoussan and J. L. Lions, eds., Lecture Notes in Control and Information Sciences 83, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, June 1986, pp. 65–73.

[29] E. POLAK, *Computational Methods in Optimization*, Academic Press, New York, 1971.

[30] ———, *On the mathematical foundations of nondifferentiable optimization*, SIAM Rev., 29 (1987), pp. 21–89.

[31] E. POLAK AND L. HE, *Rate preserving discretization strategies for semi-infinite programming and optimal control*, in Proc. 29th IEEE Conf. on Decision and Control, 1990, pp. 2444–2449.

[32] E. POLAK AND D. Q. MAYNE, *An algorithm for optimization problems with functional inequality constraints*, IEEE Trans. Automat. Control, 21 (1976), pp. 184–193.

[33] E. POLAK, D. Q. MAYNE, AND D. M. STIMLER, *Control system design via semi-infinite optimization: A review*, IEEE Proc., 72 (1984), pp. 1777–1794.

[34] E. POLAK AND A. L. TITS, *A recursive quadratic programming algorithm for semi-infinite optimization problems*, Appl. Math. Optim., 8 (1982), pp. 325–349.

[35] M. J. D. POWELL, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630, G. A. Watson, ed., Springer-Verlag, Berlin, New York, 1978, pp. 144–157.

[36] ———, *A tolerant algorithm for linearly constrained optimization calculations*, Math. Programming, 45 (1989), pp. 547–566.

[37] J. RAKOWSKA, R. T. HAFTKA, AND L. T. WATSON, *Multi-objective control-structure optimization via homotopy methods*, SIAM J. Optim., 3 (1993), pp. 654–667.

[38] R. REEMTSEN, *Discretization methods for the solution of semi-infinite programming problems*, J. Optim. Theory Appl., 71 (1991), pp. 85–103.

[39] S. M. ROBINSON, *Perturbed Kuhn–Tucker points and rates of convergence for a class of nonlinear-programming algorithms*, Math. Programming, 7 (1974), pp. 1–16.

[40] K. SCHITTKOWSKI, *QLD : A FORTRAN Code for Quadratic Programming, User's Guide*, Mathematisches Institut, Universität Bayreuth, Bayreuth, Germany, 1986.

[41] ———, *Solving Nonlinear Programming Problems with Very Many Constraints*, Report No. 294, Mathematisches Institut, Universität Bayreuth, Bayreuth, Germany, 1991.

[42] G. A. WATSON, *Globally convergent methods for semi-infinite programming*, BIT, 21 (1981), pp. 362–373.

[43] W. ZHAO AND S. AZARM, *A cross-sectional shape multiplier method for two-level optimum design of frames*, in Advances in Design Automation, Volume II–Optimal Design and Mechanical Systems Analysis, ASME Publication DE, 23-2, September 1990, pp. 197–205.

[44] J. L. ZHOU, *Fast, Globally Convergent Optimization Algorithms, with Application to Engineering System Design*, Ph.D. thesis, University of Maryland, College Park, MD, 1992.

[45] J. L. ZHOU AND A. L. TITS, *Nonmonotone Line Search for Minimax Problems*, J. Optim. Theory Appl., 76 (1993), pp. 455–476.

[46] G. ZOUTENDIJK, *Methods of Feasible Directions*, Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 1960.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 488–512, May 1996
© 1996 Society for Industrial and Applied Mathematics
012

A PARALLEL METHOD FOR UNCONSTRAINED DISCRETE-TIME OPTIMAL CONTROL PROBLEMS*

DANIEL RALPH†

Abstract. Discrete-time optimal control (DTOC) problems are potentially large, highly structured optimization problems. In such a problem there are $N+1$ stages, or time steps; corresponding to time step i are a vector of control variables u_i and a vector of state variables x_i. When N is large relative to the dimension of each u_i and x_i, efficient methods for unconstrained DTOC problems have long been known. We propose a parallel method for unconstrained DTOC problems via stagewise decomposition: given $J > 1$ processors and $I_1 = 1 < I_2 < \cdots < I_{J+1} = N+1$, J subproblems are formed by grouping stages 1 to $I_2 - 1$, I_2 to $I_3 - 1$, and so on. The subproblems are independent except for separator variables, $x_{I_2}, \ldots, x_{I_J}$, which are treated specially to retain the quadratic local convergence typical of serial DTOC algorithms. Computational examples using the Intel Paragon OSF/1 computer are given.

Key words. discrete-time optimal control, stagewise decomposition, parallel algorithm, quadratic convergence, MIMD computer

AMS subject classifications. 49M40, 68Q22, 90C30, 90C39

1. Introduction.

Discrete-time optimal control (DTOC) includes optimization problems in thousands of nonlinear variables. The goal is to develop fast methods for these large problems; the key to this is taking advantage of the special structure given by DTOC. A host of applications includes the treatment of polluted groundwater [4], structural analysis [22], and aircraft control [14]. In each case there is a model of the state of a (physical) system which includes variables over which we have direct control. In aircraft control, the position of the craft in the atmosphere or space and its velocity may constitute its state at a given time; the controls may include the amount of thrust of the engines and the lift generated by the position of the wing flaps. One question is how to minimize the fuel necessary to move from one position to another in a given time. The need for obtaining solutions quickly, hence for parallel algorithms, is partly motivated by this aerospace problem: a real time (re-)optimization process may be needed to determine an efficient trajectory.

Constrained DTOC problems have the form

$$\begin{aligned} &\min_{x_i,u_i} \sum_{i=1}^{N} l_i(x_i,u_i) + l_{N+1}(x_{N+1}) \\ &\text{subject to } \; x_{i+1} = f_i(x_i,u_i),\ i = 1,\ldots,N, \quad x_1 \text{ fixed}, \\ &\qquad\qquad g_i(x_i,u_i) \le 0, \qquad i = 1,\ldots,N, \quad g_{N+1}(x_{N+1}) \le 0, \end{aligned}$$

where $l_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and $g_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ are twice differentiable functions with locally Lipschitz second derivatives, for each i. DTOC often arises from discretizations of continuous optimal control problems, in which the variables x and u are functions of continuous time, and an integral replaces the sum

* Received by the editors March 3, 1993; accepted for publication (in revised form) September 21, 1994. This research was partially supported by the Cornell Theory Center, which receives major funding from the U. S. National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute; the National Science Foundation; the U. S. Air Force Office for Scientific Research; the U. S. Office for Naval Research, through National Science Foundation grant DMS-8920550; and the Australian Research Council.

† Department of Mathematics, University of Melbourne, Parkville, Victoria 3052, Australia.

in the objective function. Here, however, x_i is the vector of state variables at time step or stage i, u_i is the corresponding vector of controls, and the number of stages in the discretization is $N+1$. The equality constraints, $x_{i+1} = f_i(x_i, u_i)$, are called transition constraints. There is no need, either in the formulation process or most algorithms, to fix the dimension of the states and controls across all stages at n and m, respectively; we do so for convenience.

In the absence of inequality constraints, given all controls u_i, the states x_i are feasibly determined by the transition constraints. Hence such DTOC problems are said to be unconstrained. The unconstrained objective function $\mathcal{J}(u_1, \ldots, u_N)$ that results by recursively eliminating the state variables from stage $N+1$ down to stage 2 in $\sum_i l_i$ apparently has no sparsity structure [16]: its Hessian is generally dense. If Newton's method, for finding where the gradient of $\mathcal{J}$ is zero, is applied naively, each iteration requires factorization of a dense matrix of order Nm, i.e., at least $O(N^3m^3)$ flops/iteration.

In contrast, the differential dynamic programming (DDP) method of Mayne [13] (see also Jacobson and Mayne [10]) uses efficient recurrence formulae that take only $O(N(n+m)^3)$ flops/iteration, a huge reduction in the computation time needed for linear algebra if N is large relative to n and m. Also the convergence properties of DDP are similar to or, it is claimed [12], better than Newton's method. More recently Dunn and Bertsekas [7] and de O. Pantoja [18] have used similar recurrences to perform one iteration of Newton's method in $O(N(n+m)^3)$ time. Polak [20] uses an alternative recurrence method with similar computational complexity. More generally, Wright [24] has given an implementation of sequential quadratic programming (SQP) that also achieves the same bound on flops/iteration.

These efficient methods seem difficult to parallelize, however, because of the recursive nature of the transition constraints. We propose a parallel method for such DTOC problems given $J > 1$ processors and $I_1 = 1 < I_J < \cdots < I_{J+1} = N+1$: J subproblems are formed by grouping stages 1 to $I_2 - 1$, I_2 to $I_3 - 1$, ..., I_J to $N+1$. The subproblems are independent except for separator variables, $x_{I_2}, \ldots, x_{I_J}$, which are treated as exogenous to allow parallel solution, in the remaining variables, of quadratic programs approximating each of the J nonlinear subproblems. The separator variables must then be updated carefully to retain the quadratic local convergence typical of serial DTOC algorithms. This proposal can be considered an extension of the parallel method of Chang, Chang, and Luh [2], which is restricted to affine (linear) transition functions. Such parallel methods are called stagewise partitioning or stagewise decomposition methods.

Other parallel methods include work by Wright using a general parallel band solver [26] within a sequential quadratic programming method [24] and stagewise partitioning based on efficient recurrence methods for DTOC [25] and work by Psiaki and Park [21] on a divide and conquer method. For DTOC problems with relatively large $n+m$, parallelism can readily be used within each time step, as demonstrated by Shoemaker et al. [23].

We note that the interest is often in constrained DTOC problems, that is, problems which include inequality constraints. It is beyond the scope of the current paper to deal with these, but we point out immediate extensions of (parallel) algorithms for unconstrained problems to the constrained case. When inequality constraints are restricted to lower and upper bounds on only control variables, it is possible to efficiently carry out a projected gradient/Newton method—see [1, 6, 24] and references therein. An approach that applies to more general inequality constraints is to use penalty

functions, as in Culver and Shoemaker [4], to "remove" the inequality constraints by augmenting each term $l_i(x_i, u_i)$ of the original objective function with a nonnegative term in (x_i, u_i) that measures the violation of the corresponding constraints, thereby converting the constrained problem to an unconstrained problem. See also [27].

The paper is set out as follows. In §2 we review some efficient methods for unconstrained DTOC problems. Section 3 describes the new stagewise decomposition algorithm, including quadratic local convergence. Section 4 concludes the paper with some computational examples performed on the Intel Paragon OSF/1 computer, a multiple-instruction multiple-data (MIMD) computer.

Some notation follows. A flop is the time taken for one floating point operation on a computer, such as adding, subtracting, multiplying, or dividing two real numbers. The Euclidean norm is denoted $\|\cdot\|_2$. By $O(d)$, where d is a vector in a normed space, we mean a function mapping that space to another normed space such that $O(d)/\|d\|$ is bounded as $d \to 0$, $d \neq 0$. We say *near* to mean *in a neighborhood of.* Gradients are denoted using ∇ and partial derivatives using ∂. To simplify notation we omit the transpose superscript "T" from vectors where this does not obscure the intent. For example, $(u_1, \ldots, u_N)$ denotes the column vector $(u_1^T, \ldots, u_N^T)^T$, and $f^i(x_i, u_i)p_{i+1}$ denotes the inner product $f^i(x_i, u_i)^T p_{i+1}$, where $p_{i+1} \in \mathbb{R}^n$. The transpose is retained for matrices, however, so that the first and second derivatives with respect to (x_i, u_i) of $f^i(x_i, u_i)p_{i+1}$ are written $\nabla f^i(x_i, u_i)^T p_{i+1}$ and $\nabla^2 f^i(x_i, u_i)^T p_{i+1}$, respectively.

2. Review.

2.1. Nonlinear programming. Given the initial state vector $x_1 \in \mathbb{R}^n$, let us rewrite the unconstrained DTOC problem

$$\text{DTOC}(x_1) \qquad \begin{aligned} \min_{x_i, u_i} \ & \sum_{i=1}^{N} l^i(x_i, u_i) + l^{N+1}(x_{N+1}) \\ \text{subject to} \quad & x_1 \text{ given}, \ x_{i+1} = f^i(x_i, u_i), \ i = 1, \ldots, N \end{aligned}$$

as a general equality constrained nonlinear program that is parameterized by x_1:

$$\min_{X,U} L(X, U) \ \text{ subject to } \ x_1 \text{ given}, \ F(X, U) = 0,$$

where

$$\begin{aligned} X &\stackrel{\text{def}}{=} (x_1, \ldots, x_{N+1}) \in \mathbb{R}^{n(N+1)}, \\ U &\stackrel{\text{def}}{=} (u_1, \ldots, u_N) \in \mathbb{R}^{mN}, \\ L(X, U) &\stackrel{\text{def}}{=} \sum_{i=1}^{N} l^i(x_i, u_i) + l^{N+1}(x_{N+1}) \in \mathbb{R}, \\ F(X, U) &\stackrel{\text{def}}{=} (f^1(x_1, u_1) - x_2, \ldots, f^N(x_N, u_N) - x_{N+1}) \in \mathbb{R}^{nN}. \end{aligned}$$

For details on the principles of nonlinear programming see Fletcher [8].

Given a multiplier $P \stackrel{\text{def}}{=} (p_2, \ldots, p_{N+1}) \in \mathbb{R}^{nN}$, the *Hamiltonian* of the ith stage of the system is

$$\mathcal{L}^i(x_i, u_i; p_{i+1}) \stackrel{\text{def}}{=} l^i(x_i, u_i) + f^i(x_i, u_i)p_{i+1}, \quad i = 1, \ldots, N.$$

The multiplier vectors p^{i+1}, $i = 1, \ldots, N$, are called *costates.* For future use, we introduce other vectors of multipliers in $\mathbb{R}^{nN}$: $P \stackrel{\text{def}}{=} (p_2, \ldots, p_{N+1})$ and $P^* \stackrel{\text{def}}{=} (p_2^*, \ldots, p_{N+1}^*)$.

The Lagrangian is

$$\begin{aligned}&\mathcal{L}(X,U;\,P)\\ &\quad\stackrel{\text{def}}{=}\ L(X,U)+F(X,U)^TP\\ &\quad=\ \mathcal{L}^1(x_1,u_1;\,p_2)+\sum_{i=2}^{N}(\mathcal{L}^i(x_i,u_i;\,p_{i+1})-x_ip_i)+l^{N+1}(x_{N+1})-x_{N+1}p_{N+1}.\end{aligned}$$

Given a local minimizer (X^*,U^*) of DTOC(x_1), since $\nabla F(X^*,U^*)$ is surjective by inspection, the following condition holds.

First order (necessary) condition. Using ∇ to mean derivatives with respect to (X,U), we have $x_1^*=x_1$ and, for some multiplier $P^*\in\mathbb{R}^{nN}$,

$$\begin{aligned}\nabla\mathcal{L}(X^*,U^*;\,P^*)=\nabla L(X^*,U^*)+\nabla F(X^*,U^*)^TP^*=0,\\ F(X^*,U^*)=0.\end{aligned}$$

It is sufficient for (X^*,U^*) to be a local minimizer that the following condition holds.

Second order (sufficient) condition. The first order condition holds for DTOC(x_1) at (X^*,U^*) with optimal multiplier P^*, and the Hessian of the Lagrangian, $\nabla^2\mathcal{L}(X^*,U^*;\,P^*)$, is positive definite on the null space of $\nabla F(X^*,U^*)$.

2.2. Sequential quadratic programming and its convergence. Given a current point and multiplier, (X,U) and P, for ease of notation we may denote the values of functions at these points by the names of the functions themselves: write L for $L(X,U)$, $\nabla^2\mathcal{L}$ for $\nabla^2\mathcal{L}(X,U;\,P)$, etc. Assuming $\nabla^2\mathcal{L}$ is positive definite on the null space of ∇F, the SQP method and the Lagrange–Newton method coincide: the change to (X,U), (Y,V), and an updated multiplier, P^+, are determined by solving a linearization of the above first order system:

$$\text{(1)}\qquad\begin{array}{rcrcrcl}\nabla L&+&\nabla^2\mathcal{L}\ (Y,V)&+&\nabla F^TP^+&=&0,\\ F&+&\nabla F\ (Y,V)&&&=&0.\end{array}$$

Our new point and multiplier are $(X,U)\stackrel{\text{def}}{=}(X+Y,U+V)$ and $P\stackrel{\text{def}}{=}P^+$, and we can repeat the SQP/Lagrange–Newton procedure.

There are many variants of this basic scheme which allow, for instance, different multiplier updates. See Nocedal and Overton [17] for a review. We expect Q-quadratic convergence in many interesting cases. The following result is noted by Wright [24].

THEOREM 2.1. *Let L and F be as above and (X^*,U^*) be a local minimizer of DTOC(x_1) with a corresponding optimal multiplier P^*. Suppose the second order condition holds for DTOC(x_1) at (X^*,U^*).*

1. *For any (X^0,U^0,P^0) near (X^*,U^*,P^*), the SQP sequence defined, for $k=0,1,\ldots$, by*

$$(X^{k+1},U^{k+1},P^{k+1})\ \stackrel{\text{def}}{=}\ (X^k+Y^k,U^k+V^k,P^{k+})$$

is well defined and Q-quadratically convergent to (X^*,U^*,P^*):

$$\begin{aligned}&\|(X^{k+1}-X^*,U^{k+1}-U^*,P^{k+1}-P^*)\|\\ &=O(\|(X^k-X^*,U^k-U^*,P^k-P^*)\|^2).\end{aligned}$$

2. *Given $k \geq 0$ and (X^k, U^k), let the next iterate be generated by choosing a multiplier P^k such that*

$$P^k - P^* \;=\; O((X^k - X^*, U^k - U^*)),$$

then using SQP to determine (Y^k, V^k) (and P^{k+} if necessary), and setting $(X^{k+1}, U^{k+1}) \stackrel{\text{def}}{=} (X^k + Y^k, U^k + V^k)$. Then for any (X^0, U^0) near (X^*, U^*), the sequence $((X^k, U^k))$ converges Q-quadratically to (X^*, U^*).

3. *Given $k \geq 0$ and U^k, let the next control vector be generated by first calculating a feasible state vector X^k, that is, $x_1^k = x_1$ and*

$$x_{i+1}^k \;=\; f^i(x_i^k, u_i^k), \;\; i = 2, \ldots, N,$$

second choosing a multiplier P^k as in part 2, third using SQP to determine V^{k+1} (and Y^k and/or P^{k+} if necessary), and finally setting $U^{k+1} \stackrel{\text{def}}{=} U^k + V^k$. Then, for any U^0 near U^*, the sequence (U^k) converges Q-quadratically to U^*.

Denote the partial derivative values of $\mathcal{L}^i$ at $(x_i^*, u_i^*, p_{i+1}^*)$ as

$$\mathcal{L}^i_{x^*} \stackrel{\text{def}}{=} \partial \mathcal{L}^i(x_i^*, u_i^*; p_{i+1}^*)/\partial x_i, \;\; \mathcal{L}^i_{x^*u^*} \stackrel{\text{def}}{=} \partial^2 \mathcal{L}^i(x_i^*, u_i^*; p_{i+1}^*)/\partial u_i \partial x_i,$$

etc. In detail, the first order optimality condition satisfied by (X^*, U^*, P^*) is that $x_1^* = x_1$ and

$$\begin{aligned}
\mathcal{L}^i_{x^*} - p_i^* &= 0, \quad i = 2, \ldots, N, \\
l^{N+1}_{x^*} - p_{N+1}^* &= 0, \\
\mathcal{L}^i_{u^*} &= 0, \quad i = 1, \ldots, N, \\
f^i(x_i^*, u_i^*) - x_{i+1}^* &= 0, \quad i = 1, \ldots, N.
\end{aligned} \tag{2}$$

Let $Y \stackrel{\text{def}}{=} (y_1, \ldots, y_{N+1})$ and $V \stackrel{\text{def}}{=} (v_1, \ldots, v_N)$, where each $y_i \in R^n$ and each $v_i \in R^m$. The linear Lagrange–Newton system for finding (Y, V, P^+) is parameterized by y_1, just as DTOC(x_1) is a parametric problem:

$$\begin{array}{lllllllll}
l_x^i & + & \mathcal{L}^i_{xx} y_i & + & \mathcal{L}^i_{xu} v_i & + & (f_x^i)^T p_{i+1}^+ & - & p_i^+ \;\; = 0, \\
 & & & & & & & & \quad i = 2, \ldots, N, \\
l_x^{N+1} & + & l_{xx}^{N+1} y_{N+1} & & & & & - & p_{N+1}^+ \;\; = 0, \\
l_u^i & + & \mathcal{L}^i_{ux} y_i & + & \mathcal{L}^i_{uu} v_i & + & (f_u^i)^T p_{i+1}^+ & & \;\; = 0, \\
 & & & & & & & & \quad i = 1, \ldots, N, \\
s_i & + & f_x^i y_i & + & f_u^i v_i & & & - & y_{i+1} \;\; = 0, \\
 & & & & & & & & \quad i = 1, \ldots, N,
\end{array} \tag{3}$$

where, for $i = 1, \ldots, N$, $s_i \stackrel{\text{def}}{=} f^i(x_i, u_i) - x_{i+1}$. The solution satisfies the first order optimality conditions for the SQP subproblem below, for $I = 1$ and $y_1 = 0$:

$$\begin{aligned}
\text{QP}(I; y_I) \qquad & \min_{y_i, v_i} \;\; \sum_{i=I}^{N} \left(l_x^i y_i \;+\; l_u^i v_i \;+\; \frac{1}{2}(y_i \mathcal{L}^i_{xx} y_i \;+\; 2 y_i \mathcal{L}^i_{xu} v_i \;+\; v_i \mathcal{L}^i_{uu} v_i) \right) \\
& \qquad + l_x^{N+1} y_{N+1} \;+\; \tfrac{1}{2} y_{N+1} l_{xx}^{N+1} y_{N+1} \\
& \text{subject to} \\
& \qquad y_I \text{ given}, \;\; y_{i+1} = s_i + f_x^i \, y_i + f_u^i \, v_i, \;\; i = I, \ldots, N.
\end{aligned}$$

Wright [24] solves the Lagrange–Newton system for QP(1;0) by writing it in banded form, namely, by symmetrically ordering the equations in blocks corresponding to $(u_1, p_2,\ x_2, u_2, p_3, \ldots, x_N, u_N, p_{N+1},\ x_{N+1})$. This system has $N(2n+m)$ unknowns and a half bandwidth of $2n+m$; factorization by banded Gauss elimination takes about

$$2N(2n+m)^3 \ = \ N(16n^3 \ + \ 24n^2m \ + \ 12nm^2 \ + \ 2m^3)$$

flops [9]. The number of flops used for forward and backward substitution, $O(N(2n+m)^2)$, is dominated by the factorization process. We note that symmetric indefinite factorization methods, such as Bunch–Kaufman factorization [9], do not seem attractive because the bandwidth of the system may be increased by diagonal pivoting.

2.3. Dynamic programming methods. Factorization of the Lagrange–Newton system (3) can be performed using about one quarter to one third of the number of flops above, by taking advantage of the block structure within the banding of the matrix and the structure of particular blocks. We describe such a backward block elimination procedure that constructs the Lagrange–Newton system recursively, and reduces, for specific choices of multipliers, to either DDP or a dynamic programming version of Newton's method (DPN). DPN is given by Dunn and Bertsekas [7] and, for a more restrictive form of problems, by de O. Pantoja [18].

Starting from $i = N$, each step will consist of choosing the multiplier p_{i+1} corresponding to the transition constraint at stage i, forming an SQP subproblem for a problem in y_i, v_i, y_{i+1}, and then eliminating y_{i+1} and v_i.

At stage $I \leq N$ we are given a quadratic *return* function $\Lambda^{I+1}(y_{I+1})$, which is equal to the optimal value function of QP$(I+1; y_{I+1})$ minus the constant term, and also the multiplier p_{I+1} used to construct $\mathcal{L}^I$. Λ^{I+1} is given in the form

$$\Lambda^{I+1}(y_{I+1}) \quad \stackrel{\text{def}}{=} \quad \Lambda_x^{I+1}\, y_{I+1} \ + \ \frac{1}{2} y_{I+1}\, \Lambda_{xx}^{I+1}\, y_{I+1},$$

for some $\Lambda_x^{I+1} \in \mathbb{R}^n$ and $\Lambda_{xx}^{I+1} \in \mathbb{R}^{n\times n}$. For $I = N$ it is clear that $\Lambda_x^{N+1} \stackrel{\text{def}}{=} l_x^{N+1}$ and $\Lambda_{xx}^{N+1} \stackrel{\text{def}}{=} l_{xx}^{N+1}$. The choice of the initial multiplier p_{N+1} and then a multiplier p_I for each stage $I = N, \ldots, 1$, will be discussed later. The elimination at each stage $i \geq I+1$ of the Lagrange–Newton system is represented by the linearized transition equation that determines y_{i+1},

$$y_{i+1} \ = \ s_i + f_x^i y_i + f_u^i v_i, \tag{4}$$

and by a vector $\gamma_i \in \mathbb{R}^m$ and matrix $\Gamma_i \in \mathbb{R}^{m\times n}$ that determine v_i,

$$v_i \ = \ \gamma_i + \Gamma_i y_i. \tag{5}$$

We are naturally interested in obtaining Λ^1 and the factorization of the Lagrange–Newton matrix from which the optimal values of (v_i, y_{i+1}), $i = 1, \ldots, N$, can be recovered.

Consider the problem of obtaining $\Lambda^I(y_I)$. The value of QP$(I; y_I)$ is given, up to a constant, by

$$\begin{aligned} &\min_{v_I, y_{I+1}} \Lambda^{I+1}(y_{I+1}) \ + \ l_x^I\, y_I \ + \ l_u^I\, v_I \\ &+ \frac{1}{2}(y_I\, \mathcal{L}_{xx}^I\, y_I \ + \ 2v_I\, \mathcal{L}_{ux}^I\, y_I \ + \ v_I\, \mathcal{L}_{uu}^I\, v_I) \\ &\text{subject to } \ y_I \text{ given}, \ y_{I+1} = s_I + f_x^I y_I + f_u^I v_I. \end{aligned}$$

By eliminating y_{I+1} and dropping constant terms, we get an unconstrained problem with the objective function

$$\lambda^I(y_I, v_I) \stackrel{\text{def}}{=} \lambda^I_x\, y_I \;+\; \lambda^I_u\, v_I \;+\; \frac{1}{2}(y_I\, \lambda^I_{xx}\, y_I \;+\; 2v_I\, \lambda^I_{ux}\, y_I \;+\; v_I\, \lambda^I_{uu}\, v_I),$$

where

$$\begin{array}{rcl} \lambda^I_x & \stackrel{\text{def}}{=} & l^I_x \;+\; (f^I_x)^T[\Lambda^{I+1}_x \;+\; \Lambda^{I+1}_{xx} s_I], \\ \lambda^I_u & \stackrel{\text{def}}{=} & l^I_u \;+\; (f^I_u)^T[\Lambda^{I+1}_x \;+\; \Lambda^{I+1}_{xx} s_I], \\ \lambda^I_{xx} & \stackrel{\text{def}}{=} & l^I_{xx} \;+\; (f^I_{xx})^T p_{I+1} \;+\; (f^I_x)^T \Lambda^{I+1}_{xx} f^I_x, \\ \lambda^I_{xu} & \stackrel{\text{def}}{=} & l^I_{xu} \;+\; (f^I_{xu})^T p_{I+1} \;+\; (f^I_x)^T \Lambda^{I+1}_{xx} f^I_u, \\ \lambda^I_{uu} & \stackrel{\text{def}}{=} & l^I_{uu} \;+\; (f^I_{uu})^T p_{I+1} \;+\; (f^I_u)^T \Lambda^{I+1}_{xx} f^I_u. \end{array} \tag{6}$$

Assuming that λ^I_{uu} is positive definite, a point we will discuss later, the problem $\min_{v_I} \lambda^I(y_I, v_I)$ is solvable for any y_I, and its solution is characterized by the first order condition that the gradient of λ^I with respect to v_I be zero:

$$\lambda^I_u \;+\; \lambda^I_{uu} v_I \;+\; \lambda^I_{ux} y_I \;=\; 0.$$

So $v^I = \gamma_I + \Gamma_I y_I$, where $\gamma_I \stackrel{\text{def}}{=} -(\lambda^I_{uu})^{-1}\lambda^I_u$ and $\Gamma_I \stackrel{\text{def}}{=} -(\lambda^I_{uu})^{-1}\lambda^I_{ux}$, and we can construct Λ^I_x and Λ^I_{xx} by defining

$$\Lambda^I(y_I) \stackrel{\text{def}}{=} \lambda^I(y_I, \gamma_I + \Gamma_I y_I) \;-\; \text{constant term}.$$

In particular, $\Lambda^I_x \stackrel{\text{def}}{=} \lambda^I_x + \Gamma^T_I \lambda^I_u$. Except for the choice of multiplier p_I, this completes one step of the backward block elimination.

In DPN we use the second equation of (2) to motivate our choice p_{N+1}: we can satisfy part of the first order optimality conditions at the current point by letting $p_{N+1} \stackrel{\text{def}}{=} l^{N+1}_x$. By the same reasoning, given (x_I, u_I) and p_{I+1} for $2 \le I \le N$, the first equation of (2) dictates that

$$p_I \stackrel{\text{def}}{=} \mathcal{L}^I_x \;=\; l^I_x + (f^I_x)^T p_{I+1}.$$

Wright [24] first observed this connection between dynamic programming methods for DTOC and the choice of multiplier in SQP in the context of Dunn and Bertsekas' version of DPN [7].

We now make a similar connection between DDP and multipliers in SQP; we thank Mark Psiaki[1] for suggesting in a conversation that there might be such a connection. In the DDP framework,

$$p_I \stackrel{\text{def}}{=} \Lambda^I_x + \Lambda^I_{xx} s_{I-1}.$$

In particular we have

$$p_{N+1} \stackrel{\text{def}}{=} \Lambda^{N+1}_x + \Lambda^{N+1}_{xx} s_N \;=\; l^{N+1}_x + l^{N+1}_{xx} s_N.$$

[1] Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY.

We motivate this choice based on the derivation of DDP given by Murray and Yakowitz [16]. Consider the nonlinear return function for stage I, which is parametric in x_I:

$$(7) \qquad \begin{aligned} \min_{u_i, x_i} \quad & \sum_i l^i(x_i, u_i) + l^{N+1}(x_{N+1}) \\ \text{subject to} \quad & x_I \text{ given}, \ x_{i+1} = f^i(x_i, u_i), \ i = I, \ldots, N. \end{aligned}$$

Λ^I is defined as a quadratic approximation, based at

$$(x_I, u_I, \ldots, x_N, u_N, x_{N+1}),$$

of the change in the optimal value of (7) if x_I is changed to $x_I + y_I$. So given $\Lambda^{I+1}(y_{I+1})$, x_I, u_I, and x_{I+1}, we may obtain $\Lambda^I(x_I)$ by considering the following nonlinear problem which is parametric in x_I:

$$\begin{aligned} \min_{v_I, y_{I+1}} \quad & l^I(x_I + y_I, u_I + v_I) + \Lambda^{I+1}(y_{I+1}) \\ \text{subject to} \quad & x_I \text{ given}, \ y_{I+1} = f^I(x_I + y_I, u_I + v_I) - x_{I+1}. \end{aligned}$$

The vector y_{I+1} can immediately be eliminated to yield an unconstrained problem whose objective function is

$$l^I(x_I + y_I, u_I + v_I) + \Lambda^{I+1}(f^I(x_I + y_I, u_I + v_I) - x_{I+1}).$$

The quadratic Taylor approximation about (x_I, u_I), less the constant term, coincides with λ^I, from (6), when $p_{I+1} = \Lambda_x^{I+1} + \Lambda_{xx}^{I+1} s_I$, that is, when p_{I+1} is the above DDP multiplier.

The DDP and DPN recurrence formulae for feasible (X, U) — i.e., x_1 is given and $s_2, \ldots, s_{N+1}$ are zero — follow.

DDP/DPN backward elimination for feasible (X, U)
$\Lambda_x^{N+1} \leftarrow l_x^{N+1}$, $\Lambda_{xx}^{N+1} \leftarrow l_{xx}^{N+1}$, $p_{N+1} \leftarrow l_x^{N+1}$

for $i = N, \ldots, 2$
  auxiliary data:
    Set λ_x^i, λ_u^i, λ_{xx}^i, λ_{ux}^i, and λ_{uu}^i as in (6) with $I = i$.
  save these:
    $\gamma_i \leftarrow -(\lambda_{uu}^i)^{-1}\lambda_u^i$, $\Gamma_i \leftarrow -(\lambda_{uu}^i)^{-1}\lambda_{ux}^i$.
  auxiliary data:
    $\Lambda_x^i \leftarrow \lambda_x^i + \Gamma_i^T \lambda_u^i$
    DDP: $p_i \leftarrow \Lambda_x^i$, or DPN: $p_i \leftarrow l_x^i + (f_x^i)^T p_{i+1}$
    $\Lambda_{xx}^i \leftarrow \lambda_{xx}^i + (\lambda_{ux}^i)^T \Gamma_i$
endfor
$(i = 1)$
auxiliary:
  Set λ_u^1 and λ_{uu}^1 as in (6) with $I = 1$
save:
  $\gamma_1 \leftarrow -(\lambda_{uu}^1)^{-1}\lambda_u^1$

"Auxiliary" data at stage i may need to be saved for stage $i - 1$. Data in the "save" category are needed after the backward pass has been completed.

For infeasible (X, U) two changes must be made. First, DDP assigns $p_i \stackrel{\text{def}}{=} \Lambda_x^i + \Lambda_{xx}^i s_i$ while, as before, DPN assigns $p_i \stackrel{\text{def}}{=} l_x^i + (f_x^i)^T p_{i+1}$, for each $i = N + 1$ down to

2. Second, the step for $i = 1$ must produce Γ_1 since y_1 need not be zero; hence the derivatives λ^1_x, λ^1_{xx}, and λ^1_{ux} must also be calculated.

For feasible (X, U), a comparison of the Dunn and Bertsekas' DPN recurrences, as they appear in [24], with the above version shows that these DPN methods are identical.

In principle, both DDP and DPN backward elimination require

$$N(4n^3 + 7n^2m + 4nm^2 + (2/3)m^3)$$

flops, plus lower order terms. This flop count assumes symmetry of derivatives is taken into account so that the calculation of λ^i_{xx} uses $4n^3$ flops rather than $6n^3$ flops, and Cholesky factorization is used to factor λ^i_{uu} in $(2/3)m^3$ flops; see [9]. If, as is usual, n is larger than m then the leading term is $4Nn^3$. This is Nn^3 higher than the corresponding flop count given by Wright [25] for DPN. There, however, the cost of forming $\mathcal{L}^i_{xx}$, given the matrix $l^i_{xx} \in \mathbb{R}^{n\times n}$ and the multiplier p_{i+1}, is excluded from the flop count, the difference over all time stages being Nn^3.

Our implementation does not take advantage of symmetry; hence we use around

$$N(6n^3 + 8n^2m + 6nm^2 + (4/3)m^3)$$

flops. In particular the leading term of the flop count is $6Nn^3$ because general matrix multiplication routines are used to calculate λ^i_{xx}.

Forward substitution to find the new (X, U) using the equations (4) and (5) is straightforward. To distinguish between the current stage i variables (x_i, u_i) and the updated variables $(x_i + y_i, u_i + v_i)$, we denote the latter by (x_i^+, u_i^+). For simplicity assume x_1 is already at its required value, so $x_1^+ = x_1$ and $x_1 = 0$. We will see that the SQP variant which uses the DDP or DPN backward elimination and forward substitution, as follows, is covered by part 2 of Theorem 2.1.

SQP forward substitution
$v_1 \leftarrow \gamma_1,\ u_1^+ \leftarrow u_1 + v_1$
$y_2 \leftarrow s_1 + f^1_u v_1,\ x_2^+ \leftarrow x_2 + y_2$
for $i = 2, \ldots, N$
  $v_i \leftarrow \gamma_i + \Gamma_i y_i,\ u_i^+ \leftarrow u_i + v_i$
  $y_{i+1} \leftarrow s_i + f^i_x y_i + f^i_u v_i,\ x^+_{i+1} \leftarrow x_{i+1} + y_{i+1}$
endfor

In both DDP and DPN the (linear) substitution process is augmented by nonlinear substitution, namely, given the new point (x_i^+, u_i^+), let $x^+_{i+1} \leftarrow f^i(x_i^+, u_i^+)$. We will see that each of DDP and DPN are described by part 3 of Theorem 2.1.

DPN forward substitution
$v_1 \leftarrow \gamma_1,\ u_1^+ \leftarrow u_1 + v_1$
$y_2 \leftarrow s_1 + f^1_u v_1,\ x_2^+ \leftarrow f^1(x_1^+, u_1^+)$
for $i = 2, \ldots, N$
  $v_i \leftarrow \gamma_i + \Gamma_i y_i,\ u_i^+ \leftarrow u_i + v_i$
  $y_{i+1} \leftarrow s_i + f^i_x y_i + f^i_u v_i,\ x^+_{i+1} \leftarrow f^i(x_i^+, u_i^+)$
endfor

DPN forward substitution is equivalent to SQP forward substitution to obtain the updated states x_i^+ and controls u_i^+, followed by a second update of the states, this time nonlinear: let $x_1^{++} = x_1$, and $x^{++}_{i+1} \leftarrow f^i(x_i^{++}, u_i^+)$ for $i = 1, \ldots, N$. DDP

substitution takes the nonlinear update of x_i into account during rather than after the calculation of the new u_i.

DDP forward substitution for feasible (X, U)
$u_1^+ \leftarrow u_1 + \gamma_1$
$y_2 \leftarrow f^1(x_1^+, u_1^+) - x_2, \;\; x_2^+ \leftarrow f^1(x_1^+, u_1^+)$
for $i = 2, \ldots, N$
$\quad u_i^+ \leftarrow u_i + \gamma_i + \Gamma_i y_i$
$\quad y_{i+1} \leftarrow f^{i+1}(x_{i+1}^+, u_{i+1}^+) - x_i, \;\; x_{i+1}^+ \leftarrow f^i(x_i^+, u_i^+)$
endfor

The computational complexity of forward substitution, excluding the function evaluations $f^i(x_i^+, u_i^+)$, is low order compared to the complexity of backward elimination.

See Liao and Shoemaker [12] for a comparison of DDP, DPN, and a hybrid that uses DPN backward elimination and DDP forward substitution. As noted there, the update to the control vector is the same for both DDP and DPN for the case of affine transition functions. For problems with highly nonlinear transition functions, however, it is claimed that there is a slight advantage of DDP backward elimination over the DPN version — the argument being [10] that the DDP multiplier update takes more of the nonlinearity of the transition functions into account — and a large advantage of DDP forward substitution over the DPN version.

However, it is still not generally accepted that DDP offers significant computational advantages over DPN, especially with regard to the backward elimination. We paraphrase two points made by a referee on this sometimes contentious issue. First, a comparison of local, i.e., full step versions of DDP and DPN, is needed (in [12] a line search is used), both theoretically, in order to see if DDP really does have a better than quadratic convergence rate, and computationally. Second, methods like steepest descent and approximate Newton methods (e.g., in which second derivatives are approximated by either reusing second derivatives calculated in a previous iteration or by a quasi-Newton technique) may yield competitive solution times from the perspective of global, rather than local, convergence.

2.4. Convergence of dynamic programming methods. We briefly mention a selection of milestones in the development of DDP and related methods. Mayne [13] introduced DDP, primarily for DTOC, and followed this with a book co-authored with Jacobson [10] that includes extensive discussions of the discrete and the continuous cases. The DDP method and a globally convergent extension of it, using a line search, are given for the discrete case in [10]; however, convergence is barely mentioned although convergence is claimed for the global extension [10, p. 114]. The first proof of quadratic (local) convergence of DDP is given by Murray [15]. Yakowitz and Rutherford [28] give the first convergence proof of the globally convergent extension. Murray and Yakowitz [16] later make the comparison between the step obtained by Newton's method and the step obtained by DDP and prove quadratic (local) convergence by showing that the DDP step is sufficiently close to the Newton step. Later, de O. Pantoja [18] and Dunn and Bertsekas [7] give further insight into Newton's method and DDP by presenting the former as DPN. Then Wright [24] places DPN in the context of large-scale nonlinear programming by showing that DPN defines a block-matrix factorization of the Hessian of the appropriate Lagrangian function.

Our next, well-known result gives two characterizations of the second order condition at a local minimizer. The former is useful in showing convergence of DDP and DPN. The latter gives some insight into the second order condition.

PROPOSITION 2.2. *The second order condition for DTOC(x_1) at (X^*,U^*) is equivalent to the first order condition with either of the following statements.*

1. *There are constants $0 < c_1 < c_2$ such that for each feasible (X,U) near (X^*,U^*), for each $i = N,\dots,1$ in either DDP or DPN backward elimination, the eigenvalues of λ^i_{uu} lie between c_1 and c_2.*

2. *The unconstrained objective function $\mathcal{J}(U)$, given by recursively eliminating the state variables from $L(X,U)$ using the equations $x_{i+1} = f^i(x_i,u_i)$ for $i = N,\dots,1$, has a positive definite Hessian $\nabla^2\mathcal{J}(U^*)$.*

Proof. We recall that the above DPN method is equivalent to the method given by Dunn and Bertsekas [7]. Wright [24] showed, using [7], that the second order condition is equivalent both to statement 2 and to statement 1 as it applies to DPN. Murray and Yakowitz [16] showed that statement 2 is equivalent to statement 1 as it applies to DDP. □

Given part 1, the backward factorization process must succeed for any point near a minimizer that satisfies the second order condition. We see that the backward factorization is actually a "null space" implementation of SQP: we work in the feasible set of the linearized constraints and thereby are able to detect indefiniteness of the Hessian in this space. See [28, 11] for some computational details of dealing with indefiniteness in DDP.

As pointed out in [24], it is easy to prove by induction that the DPN multiplier P is within distance $O((X - X^*, U - U^*))$ of the optimal multiplier P^*. The DDP multiplier, although more complex to derive, satisfies the same distance bound; this is a corollary of Proposition 2.2, part 1. Finally, it is easy to inductively show that the DPN or DDP forward substitution gives a new point (X,U) such that $X - X^* = O(U - U^*)$. This yields the more or less well-known results below.

PROPOSITION 2.3. *Let (X^*,U^*) be a local minimizer of DTOC(x_1) at which the second order condition holds.*

1. *Let the sequence of points (X^k,U^k) be generated by either DDP or DPN backward elimination and SQP forward substitution. For any (X^0,U^0) near (X^*,U^*), (X^k,U^k) converges Q-quadratically to (X^*,U^*).*

2. *Let the sequence of points (X^k,U^k) be generated by either DDP or DPN backward elimination and either DDP or DPN forward substitution. For any U^0 near U^*, U^k converges Q-quadratically to U^*.*

Proof. With regard to DPN, part 1 is shown by Wright [24] and part 2 by Dunn and Bertsekas [7] (see also de O. Pantoja [18]). Regarding DDP, part 2 has been shown by Murray [15]. With regard to the mixed algorithm which uses DPN backward elimination and DDP forward substitution, part 2 has been noted by Liao and Shoemaker [12]. □

3. A parallel method.

3.1. Overview. Let $J > 1$ be the number of processors available, and divide the data of DTOC(x_1) into J groups of consecutive stages: stages $1,\dots,I_2-1$, stages $I_2,\dots,I_3-1$, and so on up to stages $I_J,\dots,N+1$, where

$$I_1 = 1 \ < \ I_2 \ < \ \cdots \ < \ I_{J+1} = N+1.$$

These groups of stages are independent of each other except for separator variables $x_{I_2},\dots,x_{I_J}$. The algorithm we will describe generates iterates (X,U) such that x_1 is given, and $x_i = f^{i-1}(x_{i-1},u_{i-1})$ for each $i = 2,\dots,N+1$ except possibly $i \in \{I_2,I_3,\dots,I_J\}$. We say such points are feasible except for $x_{I_2},\dots,x_{I_J}$. For affine

transition constraints, the motivation for this approach is given by Chang, Chang, and Luh [2]. We consider nonlinear transition functions and have developed a method that is oriented towards DDP rather than DPN. Quadratic local converge is shown.

Suppose $j \in \{1, \dots, J-1\}$ and (X, U) is feasible except for $x_{I_2}, \dots, x_{I_J}$. Consider the nonlinear program

$$\begin{aligned} &\min_{x_i, u_i} \sum_{i=I_j}^{I_{j+1}-1} l_i(x_i, u_i) \\ &\text{subject to } x_{I_j} \text{ given}, \quad x_{i+1} = f_i(x_i, u_i),\ i = I_j, \dots, I_{j+1} - 1, \\ &\qquad\qquad x_{I_{j+1}} \text{ given}. \end{aligned}$$

Denote the change in the separator variables by

$$\xi_j \overset{\text{def}}{=} y_{I_j}, \quad j = 1, \dots, J.$$

Processor j works with an SQP subproblem of the form

$$\begin{aligned} \text{QP}^j(\hat{\imath}; y_{\hat{\imath}}, \xi_{j+1}) \qquad & \min_{y_i, v_i} \sum_{i=\hat{\imath}}^{I_{j+1}-1} l_x^i\, y_i + l_u^i\, v_i + \frac{1}{2}(y_i\, \mathcal{L}_{xx}^i\, y_i + 2 v_i\, \mathcal{L}_{ux}^i\, y_i + v_i\, \mathcal{L}_{uu}^i\, v_i) \\ & \text{subject to} \\ & \quad y_{\hat{\imath}} \text{ given}, \ y_{i+1} = s_i + f_x^i y_i + f_u^i v_i,\ i = \hat{\imath}, \dots, I_{j+1} - 1, \\ & \quad y_{I_{j+1}} (= \xi_{j+1}) \text{ given}, \end{aligned}$$

where $\hat{\imath} \in \{I_j, \dots, I_{j+1} - 1\}$. In fact processor j solves $\text{QP}^j(I_j; \xi_j, \xi_{j+1})$, yielding the quadratic optimal value function $\tilde{\Lambda}^j(\xi_j, \xi_{j+1})$. A variant of DDP backward elimination that accounts for the upper endpoint constraint (ξ_{j+1} is given) is used to determine $\tilde{\Lambda}^j$. We need an assumption, the "surjectivity condition" below, to ensure $\text{QP}^j(I_j; \xi_j, \xi_{j+1})$ is feasible for all ξ_j, ξ_{j+1}. For $j = 1$, since x_1 is given (so $0 = y_1 = \xi_1$) we need only calculate $\tilde{\Lambda}^1(\xi_2)$. It turns out that the backward elimination for $j = 1$ is the same as for each $j = 2, \dots, J-1$ except for the final stage, $i = 1$, at which we do not compute derivatives of $\tilde{\Lambda}^1$ with respect to y_1.

For $j = J$, the corresponding nonlinear program is parametric only in x_{I_J} because x_{N+1} is a variable, not a given state. The SQP subproblem for processor J, denoted $\text{QP}^J(I_J; \xi_J)$, is $\text{QP}(I; y_I)$ from §2 with $I = I_j$ and $y_I = \xi_J$. So the SQP subproblem for $j = J$ can be directly factored by DDP, or DPN, to yield a quadratic return function $\tilde{\Lambda}^J(\xi_J)$.

To calculate the optimal values of $\xi_2, \dots, \xi_J$ it remains to solve the unconstrained reduced problem

$$\min_{\xi_j} \ \tilde{\Lambda}^1(\xi_2) + \sum_{j=2}^{J-1} \tilde{\Lambda}^j(\xi_j, \xi_{j+1}) + \tilde{\Lambda}^J(\xi_J). \tag{8}$$

The optimal vectors $\xi_2, \dots, \xi_J$ are distributed — ξ_2 to processor 1, ξ_j and ξ_{j+1} to each processor $j = 2, \dots, J-1$, and ξ_J to processor J — and each processor performs DDP-like forward substitution to get the new state and control values.

For (X, U) near to (X^*, U^*), the second order condition guarantees not only the existence of each $\tilde{\Lambda}^j$, but that the objective function of the reduced problem is strictly convex (see the proof of Theorem 3.2), hence obtaining $\xi_2, \dots, \xi_J$ corresponds

to solving a positive definite, block tridiagonal linear system in $J-1$ blocks of n variables. If $J << I_{j+1} - I_j$ for $1 \leq j \leq J$, it is relatively cheap to do this on one processor, called the coordinator, using banded factorization. Assuming that the data is split into roughly equal parts for load balancing, $I_{j+1} - I_j$ is approximately N/J, so we need $J^2 << N$ in order to use processors efficiently. If J^2 is relatively large, the decomposition procedure could be applied recursively within each group of time steps $I_j, \ldots, I_{j+1}$. We restrict our attention to the nonrecursive case, however.

3.2. The algorithm. We begin by outlining the backward factorization. Choose $j \in \{1, \ldots, J-1\}$. The first step of the backward pass is difficult because the last state variable $x_{I_{j+1}}$ is fixed rather than free, which may affect feasibility of $\mathrm{QP}^j(i; y_i, \xi_{j+1})$ and also makes the choice of multiplier $p_{I_{j+1}}$ difficult. We choose $\mathcal{I}_{j+1} \in \{I_j, \ldots, I_{j+1} - 1\}$ such that the constraints of $\mathrm{QP}^j(\mathcal{I}_{j+1}; y_{\mathcal{I}_{j+1}}, \xi_{j+1})$ are feasible for any choice of $y_{\mathcal{I}_{j+1}}$ and ξ_{j+1}. Such $\mathcal{I}_{j+1}$ exists for (X, U) near (X^*, U^*) under an assumption.

Surjectivity condition. For each $y_{\mathcal{I}_{j+1}}$ and $y_{I_{j+1}}$, the system

$$y_{i+1} \;=\; f^i_{x^*}\, y_i \;+\; f^i_{u^*}\, v_i, \quad i = \mathcal{I}_{j+1}, \ldots, I_{j+1} - 1$$

has a solution.

The surjectivity condition is a *controllability* property of the constraints. This condition implies, for each $i \in \{I_j, \ldots, \mathcal{I}_{j+1}\}$ and $y_i, \xi_{j+1} \in \mathbb{R}^n$, that $\mathrm{QP}^j(i; y_i, \xi_{j+1})$ is feasible.

We postpone to §3.3 details of choosing multipliers p_i for $i = \mathcal{I}_{j+1} + 1, \ldots, I_{j+1}$ needed to specify $\mathrm{QP}^j(\mathcal{I}_{j+1}; y_{\mathcal{I}_{j+1}}, \xi_{j+1})$, although we note that there are many possibilities such as using multipliers updated at the end of the previous iteration (SQP multipliers). Our choice, however, will involve DPN multipliers. Given these multipliers, we form and solve $\mathrm{QP}^j(\mathcal{I}_{j+1}; y_{\mathcal{I}_{j+1}}, \xi_{j+1})$ parametrically to obtain a solution for all stages $i = \mathcal{I}_{j+1}, \ldots, I_{j+1} - 1$, for which the change in the controls are given in the form

$$v_i = \gamma_i + \Gamma_i y_{\mathcal{I}_{j+1}} + \Xi_i \xi_{j+1}. \tag{9}$$

To describe a DDP-like step for some $i \in \{I_j, \ldots, \mathcal{I}_{j+1} - 1\}$, suppose that we are given $\mathrm{QP}^j(i+1; y_{i+1}, \xi_{j+1})$; $\Lambda^{i+1}(y_{i+1}, \xi_{j+1})$, the quadratic optimal value function, less the constant term, of this quadratic program; and the first and second partial derivatives $\Lambda^{i+1}_x, \Lambda^{i+1}_\xi \in \mathbb{R}^n$ and $\Lambda^{i+1}_{xx}, \Lambda^{i+1}_{x\xi}, \Lambda^{i+1}_{\xi\xi} \in \mathbb{R}^{n\times n}$ of Λ^{i+1} at zero. We begin by setting the multiplier $p_{i+1} \stackrel{\text{def}}{=} \Lambda^{i+1}_x$ to specify $\mathrm{QP}^j(i; y_i, \xi_{j+1})$. The optimal value of this quadratic program, up to a constant, coincides with the optimal value of $\min_{v_i} \lambda^i(y_i, v_i, \xi_{j+1})$, where

$$\begin{aligned}\lambda^i(y_i, v_i, \xi_{j+1}) \;&\stackrel{\text{def}}{=}\; \Lambda^{i+1}(f^i_x\, y_i \;+\; f^i_u\, v_i,\; \xi_{j+1}) \\ &+\; l^i_x\, y_i \;+\; l^i_u\, v_i \;+\; \frac{1}{2}(y_i\, \mathcal{L}^i_{xx}\, y_i \;+\; 2 v_i\, \mathcal{L}^i_{ux}\, y_i \;+\; v_i\, \mathcal{L}^i_{uu}\, v_i) y_{i+1}.\end{aligned}$$

We can easily calculate the derivatives of λ^i at zero, yielding $\lambda^i_x, \lambda^i_\xi \in \mathbb{R}^n$, $\lambda^i_u \in \mathbb{R}^m$, $\lambda^i_{xx}, \lambda^i_{x\xi}, \lambda^i_{\xi\xi} \in \mathbb{R}^{n\times n}$, $\lambda^i_{uu} \in \mathbb{R}^{m\times m}$, and $\lambda^i_{ux}, \lambda^i_{u\xi} \in \mathbb{R}^{m\times n}$. If $\mathrm{QP}^j(i; y_i, \xi_{j+1})$ is feasible and uniquely solvable for each (y_i, ξ_{j+1}), then λ^i_{uu} is positive definite and the minimizer v_i of $\lambda^i(y_i, \cdot, \xi_{j+1})$ is given by the first order condition

$$0 \;=\; \frac{\partial \lambda^i(y_i, v_i, \xi_{j+1})}{\partial v_i} \;=\; \lambda^i_u + \lambda^i_{uu} v_i + \lambda^i_{ux} y_i + \lambda^i_{u\xi} \xi_{j+1}.$$

Thus

$$v_i = \gamma_i + \Gamma_i y_i + \Xi_i \xi_{j+1}, \tag{10}$$

where $\gamma_i \overset{\text{def}}{=} -(\lambda^i_{uu})^{-1}\lambda^i_u$ and $\Gamma_i \overset{\text{def}}{=} -(\lambda^i_{uu})^{-1}\lambda^i_{ux}$, as in DDP, and $\Xi_i \overset{\text{def}}{=} -(\lambda^i_{uu})^{-1}\lambda^i_{u\xi}$. Note that this is closer than (9) to the serial version (5) because each Γ_i is applied to y_i, not to $y_{\mathcal{I}_{j+1}}$. The optimal value function, less the constant, of $\mathrm{QP}^j(i; y_i, \xi_{j+1})$ is therefore given by

$$\Lambda^i(y_i, \xi_{j+1}) \overset{\text{def}}{=} \lambda^i(y_i,\ \gamma_i + \Gamma_i y_i + \Xi_i \xi_{j+1},\ \xi_{j+1}),$$

completing one step of the backward elimination.

In short, apart from the initial step, parallel DDP backward elimination is performed as in the serial case with one systematic change: every function Λ^i and λ^i depends on the additional variable ξ_{j+1}.

Parallel backward elimination for processor $j = 2, \ldots, J-1$.
Let (X, U) be feasible except for $x_{I_2}, \ldots, x_{I_J}$.

Let $\mathcal{I}_{j+1} \in \{I_j, \ldots, I_{j+1} - 1\}$, then determine:
  $p_{i+1},\ i = \mathcal{I}_{j+1}, \ldots, I_{j+1} - 1$
  $\gamma_i, \Gamma_i, \Xi_i,\ i = \mathcal{I}_{j+1}, \ldots, I_{j+1} - 1$
  $\Lambda_x^{\mathcal{I}_{j+1}}, \Lambda_\xi^{\mathcal{I}_{j+1}}, \Lambda_{xx}^{\mathcal{I}_{j+1}}, \Lambda_{x\xi}^{\mathcal{I}_{j+1}}, \Lambda_{\xi\xi}^{\mathcal{I}_{j+1}}$

for $i = \mathcal{I}_{j+1} - 1, \ldots, I_j$
  auxiliary:
    set $\lambda^i_u, \lambda^i_{uu}, \lambda^i_x, \lambda^i_{xx}, \lambda^i_{ux}$ as in DDP/DPN
    $\lambda^i_\xi \leftarrow \Lambda^{i+1}_\xi,\ \lambda^i_{u\xi} \leftarrow (f^i_u)^T \Lambda^{i+1}_{x\xi},\ \lambda^i_{x\xi} \leftarrow (f^i_x)^T \Lambda^{i+1}_{x\xi},\ \lambda^i_{\xi\xi} \leftarrow \Lambda^{i+1}_{\xi\xi}$
  save:
    set γ_i, Γ_i as in DDP/DPN
    $\Xi_i \leftarrow -(\lambda^i_{uu})^{-1}\lambda^i_{u\xi}$
  auxiliary:
    $p_i \leftarrow \Lambda^i_x$ (DDP multiplier)
    set $\Lambda^i_x, \Lambda^i_{xx}$ as in DDP/DPN
    $\Lambda^i_\xi \leftarrow \lambda^i_\xi + \Xi_i^T \lambda^i_u,\ \Lambda^i_{x\xi} \leftarrow \lambda^i_{x\xi} + \Gamma_i^T \lambda^i_{u\xi},\ \Lambda^i_{\xi\xi} \leftarrow \lambda^i_{\xi\xi} + \Xi_i^T \lambda^i_{u\xi}$
endfor

In the above algorithm, the last stage I_j is dealt with by processor j and the multiplier p_{I_j} is defined but not used. In fact this multiplier is defined differently (and used) by processor $j-1$, assuming $j > 1$.

The backward elimination for $j = 1$ is as above except for $i = I_1 = 1$: no derivatives with respect to y_1 (denoted by the subscript x) are calculated. The backward elimination for processor J is given by the DDP algorithm from §2, starting at stage $N+1$ and proceeding downward to stage I_J.

The factorization of each DDP-like step is recorded using the linearized transition constraint (4), from §2, and the equations (9) and (10). At the end of the backward pass, processor 1 has produced $\Lambda^1(\xi_2)$, each processor $j = 2, \ldots, J-1$ has produced $\Lambda^{I_j}(\xi_j, \xi_{j+1})$, and processor J has produced $\Lambda^{I_J}(\xi_J)$. The quadratic return functions $\tilde{\Lambda}^j$ that make up the reduced problem are given by $\tilde{\Lambda}^j \overset{\text{def}}{=} \Lambda^{I_j}$ for each j.

To complete the factorization process the coordinator must solve the reduced problem (8). By setting the gradient of the reduced objective function to zero, we

see that solving (8) corresponds to solving a symmetric block-tridiagonal system in $J-1$ blocks of equations. The right-hand side of the system is the negative gradient vector,

$$-(\Lambda_\xi^{I_1} + \Lambda_x^{I_2}, \ldots, \Lambda_\xi^{I_{J-1}} + \Lambda_x^{I_J}).$$

Block (j,j) of the coefficient matrix is $\Lambda_{\xi\xi}^{I_j} + \Lambda_{xx}^{I_{j+1}}$, for $j = 1, \ldots, J-1$, while block $(j, j+1)$ is $\Lambda_{x\xi}^{I_{j+1}}$, for $j = 1, \ldots, J-2$.

Parallel forward substitution uses a straightforward adaptation of DDP forward substitution.

Parallel forward substitution for processor $j = 2, \ldots, J-1$.
Let (X, U) be feasible except for $x_{I_2}, \ldots, x_{I_J}$.

$x_{I_j}^+ \leftarrow x_{I_j} + \xi_j,\ u_{I_j}^+ \leftarrow u_{I_j} + \gamma_{I_j} + \Gamma_{I_j}\xi_j + \Xi_{I_j}\xi_{j+1}$
$y_{I_j+1} \leftarrow f^{I_j}(x_{I_j}^+, u_{I_j}^+) - x_{I_j+1},\ x_{I_j+1}^+ \leftarrow f^{I_j}(x_{I_j}^+, u_{I_j}^+)$
for $i = I_j + 1, \ldots, \mathcal{I}_{j+1} - 1$
  $u_i^+ \leftarrow u_i + \gamma_i + \Gamma_i y_i + \Xi_i \xi_{j+1}$
  $y_{i+1} \leftarrow f^i(x_i^+, u_i^+) - x_{i+1},\ x_{i+1}^+ \leftarrow f^i(x_i^+, u_i^+)$
endfor

for $i = \mathcal{I}_{j+1}, \ldots, I_{j+1} - 1$
  $x_i^+ \leftarrow f^{i-1}(x_{i-1}^+, u_{i-1}^+),\ u_i^+ \leftarrow u_i + \gamma_i + \Gamma_i y_{\mathcal{I}_{j+1}} + \Xi_i \xi_{j+1}$
endfor

The case $j = 1$ differs from the above for $i = I_1 = 1$: x_1 is given and $u_1^+ \leftarrow u_1 + \gamma_1 + \Xi_1\xi_2$. For $j = J$ we start by assigning $x_{I_J}^+ \leftarrow x_{I_J} + \xi_J$, $u_{I_J}^+ \leftarrow u_{I_J} + \gamma_{I_J} + \Gamma_{I_J}\xi_J$; we then proceed for $i > I_J$ exactly as in (serial) DDP forward substitution.

3.3. Multipliers to start the backward elimination. We return to the matter of obtaining multipliers p_i for $i = \mathcal{I}_{j+1} + 1, \ldots, I_{j+1}$. Let $\mathcal{I} = \mathcal{I}_{j+1}$ and $I = I_{j+1}$ for ease of notation. Given p_I, we simply use the DPN multiplier formula to obtain the other multipliers: for $i = I-1$ down to $\mathcal{I}+1$ let $p_i \stackrel{\text{def}}{=} l_x^i + (f_x^i)^T p_{i+1}$.

We estimate p_I without p_{I+1}, however, to avoid extra communication between processors. Let U_j be the subvector $(u_\mathcal{I}, \ldots, u_{I-1})$ of U and U_j^* be the corresponding subvector of U^*. Let $L^j(x_\mathcal{I}, U_j)$ be the function $\sum_{i=\mathcal{I}}^{I-1} l^i(x_i, u_i)$ after recursively substituting $f^{i-1}(x_{i-1}, u_{i-1})$ for x_i, for $i = I-1$ down to $i = \mathcal{I}+1$. Likewise $F^j(x_\mathcal{I}, U_j)$ is the transition function $f^{I-1}(x_{I-1}, u_{I-1})$ after recursively eliminating $x_{I-1}, \ldots, x_{\mathcal{I}+1}$.

LEMMA 3.1. *Let $j \in \{1, \ldots, J-1\}$, $I = I_{j+1}$, and $\mathcal{I} = \mathcal{I}_{j+1}$. The functions F^j and L^j, defined above, are twice differentiable with locally Lipschitz second derivatives. Moreover,*

1. *The surjectivity condition is equivalent to surjectivity of $F_{U^*}^j \stackrel{\text{def}}{=} \partial F^j(x_\mathcal{I}^*, U_j^*)/\partial U_j$.*

2. *Let $L_{U^*}^j \stackrel{\text{def}}{=} \partial L^j(x_\mathcal{I}^*, U_j^*)/\partial U_j$. Then the optimal multiplier p_I^* is such that*

$$L_{U^*}^j + (F_{U^*}^j)^T p_I^* = 0.$$

Proof. These results follow by straightforward induction, starting at $\mathcal{I} = I_{j+1} - 1$. The smoothness claims and part 1 are elementary. Part 2 relies on the identities $l_{u^*}^i + (f_{u^*}^i)^T p_{i+1}^* = 0$ and $l_{x^*}^i + (f_{x^*}^i)^T p_{i+1}^* = p_i^*$, from (2). □

In view of Lemma 3.1 part 2, we define p_I as the solution of the least squares problem

$$\min_{p\in\mathbb{R}^m} \|L_U^j + (F_U^j)^T p\|_2^2, \tag{11}$$

where L_U^j and F_U^j are the respective partial derivatives of L^j and F^j at $(x_{\mathcal{I}}, U_j)$ with respect to U_j. These derivatives have the form

$$L_U^j = (g_u^{\mathcal{I}}, \ldots, g_u^{I-1}) \in \mathbb{R}^{m(I-\mathcal{I})},$$
$$F_U^j = (\Phi_u^{\mathcal{I}}, \ldots, \Phi_u^{I-1}) \in \mathbb{R}^{n\times(m(I-\mathcal{I}))}$$

and can be calculated by the chain rule as follows.

$g_x^I \leftarrow 0 \in \mathbb{R}^n$, $\Phi_x^I \leftarrow$ identity $\in \mathbb{R}^{n\times n}$
for $i = I-1, \ldots, \mathcal{I}$
  save:
    $g_u^i \leftarrow l_u^i + (f_u^i)^T g_x^{i+1}$, $\Phi_u^i \leftarrow \Phi_x^{i+1} f_u^i$
  auxiliary:
    $g_x^i \leftarrow l_x^i + (f_x^i)^T g_x^{i+1}$, $\Phi_x^i \leftarrow \Phi_x^{i+1} f_x^i$
endfor

3.4. Implementation details. Let $1 \le j \le J-1$, and $I = I_{j+1}$, $\mathcal{I} = \mathcal{I}_{j+1}$.

To obtain p_I we first calculate L_U^j and F_U^j in about $2(I-\mathcal{I})(n^3+n^2m)$ flops. Then we solve (11) using a QR factorization of $(F_U^j)^T$, which takes fewer than $2(I-\mathcal{I})n^2m$ flops. In addition we can use the QR factorization to estimate the column rank of $(F_U^j)^T$, which should be n if the surjectivity condition is valid. Note that the surjectivity condition implies $\mathcal{I}$ is small enough such that the number of variables of this system, $(I-\mathcal{I})(m+n)-n$, is at least as large as the number of equations, $(I-\mathcal{I})n$, i.e., $I-\mathcal{I} \ge n/m$. In most applications $n > m$; hence $I-\mathcal{I} \ge 2$. In computational examples we have chosen $\mathcal{I}$ rather naively—as the greatest integer such that $I-\mathcal{I}$ does not exceed $(3n)/(2m)$. We have not detected rank deficiency in our limited computational tests. Calculating the DPN multipliers p_{i+1} for $i = I-1$ down to $i = \mathcal{I}$ takes $2(I-\mathcal{I})n^2$ flops, low order work.

To solve $\mathrm{QP}^j(\mathcal{I}; y_{\mathcal{I}}, \xi_{j+1})$ parametrically, we have adapted the banded SQP method of [24]. The main work is in the factorization of a banded matrix consisting of about $(I-\mathcal{I})(2n+m)$ variables and a half bandwidth of $2n+m$ and obtaining the matrices $\Gamma_{\mathcal{I}}, \ldots, \Gamma_{I-1}$ and $\Xi_{\mathcal{I}}, \ldots, \Xi_{I-1}$ by $2n$ backward and forward solves. This requires, in total, about $(I-\mathcal{I})(32n^3+40n^2m+16nm^2+2m^3)$ flops.

Thus the flop count per stage $i = \mathcal{I}_{j+1}, \ldots, I_{j+1}-1$ of the parallel elimination, including the cost of estimating the associated multipliers, is

$$34n^3 + 44n^2m + 16nm^2 + 2m^3$$

flops. This is two to four times as expensive per stage of the elimination as is the parallel method for the remaining stages (see below). Hence the larger the gap $I_{j+1} - \mathcal{I}_{j+1}$ required for the surjectivity condition, the more this overhead will detract from the parallel efficiency of the method. The relatively high flop count for these stages is mostly due to our relatively simple implementation, namely, that we are using banded factorization to solve $\mathrm{QP}^j(\mathcal{I}_{j+1}; x_{\mathcal{I}_{j+1}}, \xi_{j+1})$.

Let $1 \leq j \leq J$ and $\mathcal{I}_{J+1} = N + 1$. The flop count for stages $i = \mathcal{I}_{j+1} - 1$ down to I_j, for processor j in the parallel backward elimination, is more expensive than the serial version because we have extra data to manipulate, namely, derivatives with respect to ξ_{j+1}. The additional flops required for each stage, discarding low order terms, total

$$2n^3 + 6n^2m + 2nm^2$$

flops. This overhead is about one half of the flops required by DDP/DPN in principle and closer to one third of the flops required by our implementation of DDP. Overall, we require

$$8n^3 + 14n^2m + 8nm^2 + (4/3)m^3$$

flops for stages $i = \mathcal{I}_{j+1} - 1$ down to I_j.

Compared to serial DDP/DPN, the parallel backward elimination of [25] requires an additional

$$3n^3 + 4n^2m + 2nm^2$$

flops/stage. If $n > 2m$, then Wright's method will need more flops/stage than the method presented here, at least for $i = \mathcal{I}_{j+1} - 1, \ldots, I_j$. On the other hand, the larger n is relative to m, the larger the difference $I_{j+1} - \mathcal{I}_{j+1}$ $(\geq n/m)$ must be for the surjectivity condition to hold, hence the more flops we will need in order to deal with stages $\mathcal{I}_{j+1}, \ldots, I_{j+1}$.

We have implemented the coordination step using a general banded matrix factorization routine on a system of $(J-1)n$ variables $(\xi_2, \ldots, \xi_J)$ with a half bandwidth of $2n$; approximately $8Jn^3$ flops are needed. Forward substitution takes low order flops, apart from function evaluations.

The issue of a stopping criterion for the algorithm must still be considered. We use the residual of the first order conditions, a measure of the size of the projected gradient, and the constraint violation. In DDP, the current iterate (X, U) is always feasible and the quantity

$$\begin{aligned} r(U) &\stackrel{\text{def}}{=} \left(-\sum_1^N \gamma_i \lambda_u^i \right)^{1/2} / \max\{1, L(X,U)\} \\ &= \left(\sum_1^N \lambda_u^i (\lambda_{uu}^i)^{-1} \lambda_u^i \right)^{1/2} / \max\{1, L(X,U)\} \end{aligned}$$

is used as a measure of optimality: if the second order condition holds at (X^*, U^*) then, for feasible (X, U) near (X^*, U^*), $r(U) = 0$ if and only if $(X, U) = (X^*, U^*)$. The quantity $r(U)$ measures a scaled projected gradient. See Yakowitz and Rutherford [28] for a discussion of the role of $\sum_1^N \gamma_i \lambda_u^i$ in global convergence of DDP.

A similar optimality measure is used for the parallel method. Suppose $J \geq 2$. Recall that the current iterate (X, U) of the parallel method is feasible except for $x_{I_2}, \ldots, x_{I_J}$. Let $\gamma^j \stackrel{\text{def}}{=} (\gamma_{\mathcal{I}_{j+1}}, \ldots, \gamma_{I_{j+1}-1})$, and define $r^J(x_{I_2}, \ldots, x_{I_J}, U)$ as the quantity

$$\left(-\sum_{j=1}^{J} \sum_{i=I_j}^{\mathcal{I}_{j+1}-1} \gamma_i \lambda_u^i - \sum_{j=1}^{J-1} \gamma^j [L_U^j + (F_U^j)^T p_{I_{j+1}}] - \sum_{j=1}^{J-1} \xi_j (\Lambda_\xi^{I_j} + \Lambda_x^{I_{j+1}}) \right)^{1/2}$$

divided by $\max\{1, L(X,U)\}$. Each summand of $r^J(x_{I_2}, \ldots, x_{I_J}, U)^2$ is the inner product of a scaled projected gradient and the associated Newton step (corresponding, for instance, to a given time step), hence, from the second order condition, is nonnegative for (X,U) near the optimal vector. We also determine the Euclidean norm of the violation of the constraints,

$$v^J(x_{I_2}, \ldots, x_{I_J}, U) \\ \stackrel{\text{def}}{=} \|(x_{I_2} - f^{I_2-1}(x_{I_2-1}, u_{I_2-1}), \ \ldots \ , x_{I_J} - f^{I_J-1}(x_{I_J-1}, u_{I_J-1}))\|_2.$$

It follows from the second order condition that, for such (X,U) near (X^*,U^*), r^J and v^J both have value 0 at $(x_{I_2}, \ldots, x_{I_J}, U)$ if and only if $(X,U) = (X^*,U^*)$. So as a residual for the parallel method we use $\max\{r^J, v^J\}$.

The flop count for calculating the value of each function r, r^J, and v^J is of low order, excluding function evaluations, compared to the flop count for the (parallel) backward elimination.

3.5. Convergence and discussion. As one might expect, convergence of this method is locally quadratic.

THEOREM 3.2. *Let (X^*,U^*) be a local minimizer of DTOC(x_1) at which the second order condition holds. Let J be the number of processors and $I_1, \ldots, I_{J+1}$ be as above. For each $j = 1, \ldots, J-1$ let $\mathcal{I}_{j+1} \in \{I_j, \ldots, I_{j+1} - 1\}$ be such that the surjectivity condition holds.*

For any (X^0,U^0) near (X^*,U^*) the parallel method produces a sequence $((X^k,U^k))$, each iterate of which is feasible except for $x^k_{I_2}, \ldots, x^k_{I_J}$, that converges Q-quadratically to (X^*,U^*). Indeed,

$$(x^k_{I_2}, \ldots, x^k_{I_J}, U^k) \to (x^*_{I_2}, \ldots, x^*_{I_J}, U^*) \quad \textit{Q-quadratically.}$$

Proof. Suppose our current point (X,U) is feasible except for $x_{I_2}, \ldots, x_{I_J}$. In keeping with §2, we observe that the parallel backward elimination and coordination phases constitute a factorization scheme for an SQP subproblem of DTOC(x_1), which is based at (X,U) and uses a particular multiplier P. Let Z and Z^* denote the respective vectors of separator variables, $(x_{I_2}, \ldots, x_{I_J})$ and $(x^*_{I_2}, \ldots, x^*_{I_J})$. We will show that $(X-X^*, U-U^*) = O((Z-Z^*, U-U^*))$ and $P-P^* = O((X-X^*, U-U^*))$, from which the result follows.

Observe that continuous differentiability of each f^i implies that it is Lipschitz near (x^*_i, u^*_i). By choice of (X,U), for each $i = 2, \ldots, N+1$ except $i \in \{I_2, \ldots, I_J\}$, x_i is within distance $O((x_{i-1} - x^*_{i-1}, u_{i-1} - u^*_{i-1}))$ of x^*_{i-1}. By induction we have that $X - X^* = O((Z - Z^*, U - U^*))$.

Choose $j \in \{1, \ldots, J-1\}$ and let $I = I_{j+1}$, $\mathcal{I} = \mathcal{I}_{j+1}$. We start another induction process by claiming that for $\mathcal{I} < i \le I$, p_i is continuous in (X,U) near (X^*,U^*) and is within distance $O((X - X^*, U - U^*))$ of p^*_i. Twice continuous differentiability of L^j and F^j, from Lemma 3.1, ensures that L^j_u and F^j_u are Lipschitz functions of (X,U), near (X^*,U^*). Moreover since $F^j_{u^*}$ is surjective, from part 1 of Lemma 3.1, it is well known that the solution p_I of $\min_{p \in \mathbb{R}^m} \|L^j_u + (F^j_u)^T p\|^2$ is not only unique, but is actually a Lipschitz function of (X,U), near (X^*,U^*). Now p^*_I is the (unique) solution of $\min_{p \in \mathbb{R}^m} \|L^j_{u^*} + (F^j_{u^*})^T p\|^2$ because $L^j_{u^*} + (F^j_{u^*})^T p^*_I = 0$ from part 2 of Lemma 3.1; thus $p_I - p^*_I$ is $O((X - X^*, U - U^*))$. It now follows easily for $i = I-1$ down to $\mathcal{I}+1$ using induction and the identity $p^*_i = l^i_{x^*} + (f^i_{x^*})^T p^*_{i+1}$ from (2) that the DPN multiplier $p_i \stackrel{\text{def}}{=} l^i_x + (f^i_x)^T p_{i+1}$ depends continuously on (X,U), near (X^*,U^*), and is within distance $O((X - X^*, U - U^*))$ of being optimal.

Let $I_j \leq \hat{\imath} < \mathcal{I}$. Suppose that for $\hat{\imath} \leq i < I$, p_{i+1} is continuous in (X,U), near (X^*,U^*), such that $p_{i+1} - p^*_{i+1} = O((X-X^*, U-U^*))$. We will show that this also holds for $i = \hat{\imath}-1$. The data of $\mathrm{QP}^j(\hat{\imath}+1; y_{\hat{\imath}+1}, \xi_{j+1})$ — namely the derivatives of l^i and f^i at (x_i,u_i) and the multiplers p_{i+1}, for each $i = \hat{\imath}+1,\ldots,I-1$ — vary continuously with (X,U), near (X^*,U^*). Since $\hat{\imath}+1 \leq \mathcal{I}$, for (X,U) near (X^*,U^*), the surjectivity condition ensures that this quadratic program is always feasible. Moreover the Hessian of the objective function is positive definite on the null space of the constraints, for (X,U) near (X^*,U^*). This follows from the second order condition, given that each multiplier $p_{\hat{\imath}+2},\ldots,p_I$ converges to the corresponding optimal multiplier as (X,U) converges to (X^*,U^*) and that $\mathrm{QP}^j(\hat{\imath}+1; y_{\hat{\imath}+1}, \xi_{j+1})$ can be equated with $\mathrm{QP}(1;0)$ (from §2) with the additional constraints that $v_i = 0$ if $i \notin \{\hat{\imath}+1,\ldots,I\}$, $y_{\hat{\imath}+1}$ is given, and $y_I = \xi_{j+1}$. Thus $\mathrm{QP}^j(\hat{\imath}+1; y_{\hat{\imath}+1}, \xi_{j+1})$ has a unique solution that is continuous not only in $(y_{\hat{\imath}+1}, \xi_{j+1})$ but in (X,U).

Now the optimal value function of $\mathrm{QP}^j(\hat{\imath}+1; y_{\hat{\imath}+1}, \xi_{j+1})$, less the constant term, is $\Lambda^{\hat{\imath}+1}(y_{\hat{\imath}+1}, \xi_{j+1})$. It follows that the derivatives $\Lambda^{\hat{\imath}+1}_x$ and $\Lambda^{\hat{\imath}+1}_{xx}$ are continuous in (X,U) near (X^*,U^*), as are

$$\begin{aligned}
\lambda^{\hat{\imath}}_{uu} &\stackrel{\text{def}}{=} l^{\hat{\imath}}_{uu} + (f^{\hat{\imath}}_{uu})^T \Lambda^{\hat{\imath}+1}_x + (f^{\hat{\imath}}_u)^T \Lambda^{\hat{\imath}+1}_{xx} f^{\hat{\imath}}_u, \\
\lambda^{\hat{\imath}}_{ux} &\stackrel{\text{def}}{=} l^{\hat{\imath}}_{ux} + (f^{\hat{\imath}}_{ux})^T \Lambda^{\hat{\imath}+1}_x + (f^{\hat{\imath}}_u)^T \Lambda^{\hat{\imath}+1}_{xx} f^{\hat{\imath}}_x.
\end{aligned}$$

Substituting $\hat{\imath}$ for $\hat{\imath}+1$ above yields that $\mathrm{QP}^j(\hat{\imath}; y_{\hat{\imath}}, \xi_{j+1})$ is uniquely solvable, and its optimal value function less the constant term, $\Lambda^{\hat{\imath}}(y_{\hat{\imath}}, \xi_{j+1})$, varies continuously in (X,U) near (X^*,U^*). Since $\Lambda^{\hat{\imath}}(y_{\hat{\imath}}, \xi_{j+1})$ is also the optimal value function of $\min_{v_{\hat{\imath}}} \lambda^{\hat{\imath}}(y_{\hat{\imath}}, v_{\hat{\imath}}, \xi_{j+1})$, we must have that $\lambda^{\hat{\imath}}_{uu}$ is positive definite for (X,U) near (X^*,U^*). We are now in a position to tackle the DDP multiplier,

$$p_{\hat{\imath}} \stackrel{\text{def}}{=} \Lambda^{\hat{\imath}}_x \stackrel{\text{def}}{=} \lambda^{\hat{\imath}}_x + (\Gamma^{\hat{\imath}})^T \lambda^{\hat{\imath}}_u,$$

where $\Gamma^{\hat{\imath}} \stackrel{\text{def}}{=} -(\lambda^{\hat{\imath}}_{uu})^{-1} \lambda^{\hat{\imath}}_{ux}$.

We first deal with $\lambda^{\hat{\imath}}_x$:

$$\begin{aligned}
\lambda^{\hat{\imath}}_x &= l^{\hat{\imath}}_x + (f^{\hat{\imath}}_x)^T p_{i+1} \\
&= l^{\hat{\imath}}_{x^*} + (f^{\hat{\imath}}_{x^*})^T p^*_{i+1} + O((X-X^*, U-U^*)),
\end{aligned}$$

by the induction hypothesis. Thus, using (2), $\lambda^{\hat{\imath}}_x = p^*_{\hat{\imath}} + O((X-X^*, U-U^*))$. Similarly we have

$$\begin{aligned}
\lambda^{\hat{\imath}}_u &= l^{\hat{\imath}}_u + (f^{\hat{\imath}}_u)^T p_{\hat{\imath}+1} \\
&= l^{\hat{\imath}}_{u^*} + (f^{\hat{\imath}}_{u^*})^T p^*_{\hat{\imath}+1} + O((X-X^*, U-U^*)) \\
&= O((X-X^*, U-U^*)),
\end{aligned}$$

by the induction hypothesis and (2). Also we have seen, for (X,U) near (X^*,U^*), that $\lambda^{\hat{\imath}}_{uu}$ is invertible and continuous and $\lambda^{\hat{\imath}}_{ux}$ is continuous; thus $\Gamma^{\hat{\imath}}$ is continuous. So $(\Gamma^{\hat{\imath}})^T \lambda^{\hat{\imath}}_u = O((X-X^*, U-U^*))$, and $p_{\hat{\imath}} = p^*_{\hat{\imath}} + O((X-X^*, U-U^*))$. Finally, by continuity of $p_{\hat{\imath}+1}$ and $\Gamma^{\hat{\imath}}$, it is clear that $p_{\hat{\imath}}$ depends continuously on (X,U), near (X^*,U^*). Thus the induction hypothesis is verified for $\hat{\imath}-1$.

We have shown, for an iterate of the parallel method (X,U) near (X^*,U^*), that each processor $j = 1,\ldots,J-1$ is able to successfully perform the backward factorization process and determine multipliers

$$p_i = p^*_i + O((X-X^*, U-U^*)),$$

$i = I_j+1, \ldots, I$. Processor J actually performs DDP steps so, as we have seen in §2, it is also able to do this successfully and produce multipliers within $O((X-X^*, U-U^*))$ of their optimal values.

To complete the proof we make some observations about the reduced system (8) to show that it is strictly convex, hence solvable by factoring the associated Hessian matrix. Consider the overall SQP subproblem formed by concatenating all the quadratic programs $\mathrm{QP}^1(1;0,\xi_2)$, $\mathrm{QP}^2(I_2;\xi_2,\xi_3)$, ..., $\mathrm{QP}^{J-1}(I_{J-1};\xi_{J-1},\xi_J)$, $\mathrm{QP}^J(I_J;\xi_J)$ and letting $\xi_2, \ldots, \xi_J$ be variables in addition to (Y,V). The objective function (set of constraints) of the overall subproblem is the sum (union, respectively) of the objective functions (sets of constraints, respectively) of each quadratic program. The reduced system is in fact the Schur complement [9] of the Lagrangian system of the overall subproblem, with respect to the submatrix corresponding only to (Y,V). Now the overall subproblem is equivalent to $\mathrm{QP}(1;0)$ (from §2) and hence is strictly convex in the feasible region. It follows that the reduced problem is also strictly convex in its feasible region, i.e., in $\mathbb{R}^{(J-1)n}$. □

We briefly note that adapting this method to obtain global convergence, using line search techniques based on Yakowitz and Rutherford [28], is nontrivial because the iterates (X,U) are not necessarily feasible. This could be remedied at the expense of parallel efficiency by using a serial forward substitution process: each processor $j > 1$ updates x_{I_j} using the value calculated by processor $j-1$ instead of using the increment $y_{I_j} = \xi_j$ calculated by the coordinator. Alternatively, the current parallel algorithm could be embedded in a globally convergent scheme for nonlinear programs by using a suitable merit function [8] instead of the objective function in the line search.

We conclude this section with some remarks relating to other stagewise decomposition methods, namely those given by Chang, Chang, and Luh [2] and Wright [25]. As previously mentioned, the former paper was the starting point for this project. The fact that [2] restricts its attention to problems with affine transition functions leads to a major difference between the approach there and our approach, namely, in [2], the choice of multiplier is of little consequence. Given affine constraints, multipliers are *not* needed to form $\mathrm{QP}^j(i;y_i,\xi_{j+1})$ for $I_j \le i \le I_{j+1}-1$ because second derivatives of the constraints are identically zero; hence the update in the controls is independent of the choice of DDP or DPN or SQP multipliers. This considerably simplifies the algorithm and any analysis of its behaviour. Nonlinear transitions, however, rely on delicate choices of multipliers to obtain particular convergence results, as Theorem 2.1 shows.

We list some particular features of [2] which differ from the present paper. First, convergence is not analyzed in [2], although some arguments are presented to relate the decomposition method to a form of Newton's method that is applied to a problem equivalent to the original DTOC problem. Second, all iterates remain feasible when the transition functions are affine, so there is no need to include constraint violation in the first order conditions. In any case, details of the residual measuring the projected gradient in the parallel case are not given. Third, for simplicity of implementation, it is assumed that the surjectivity condition holds for $\mathcal{I}_{j+1} = I_{j+1} - 1$ (so $m \ge n$).

Like [2], the method presented here requires a surjectivity assumption not needed for convergence of DDP/DPN because each of our subproblems, for $j = 1, \ldots, J-1$, has an extra constraint that is not part of the standard formulation of DTOC problems. There are well-posed DTOC problems for which the surjectivity condition does not hold. It still may be possible to adapt the parallel method described to

this situation by identifying the *reachable set* of each subproblem, i.e., given ξ_j, find the set of ξ_{j+1} such that $\mathrm{QP}^j(I_j; \xi_j, \xi_{j+1})$ is feasible. The reachable set for processor j is an affine subspace which depends linearly on ξ_j, and in principle it could be included as an affine constraint in the reduced problem solved by the coordinator. The challenge for this case is to design an efficient implementation. In contrast, the the surjectivity condition guarantees that the reachable set is $\mathbb{R}^n$ for each subproblem, and the reduced problem is unconstrained.

For comparison consider the QP subproblems formed in [25]. These subproblems do not have any additional constraints, but each requires the extra assumption that the Hessian of the objective function be positive definite on the null space of the constraints. The latter assumption need not follow from the second order condition, although there is a broad class of continuous problems [7, 25] that satisfy a continuous analog of this positive definiteness assumption, which leads to the required conditions in discretized problems. Similarly, it may be that there is a continuous analog of the surjectivity condition that would give a class of continuous problems whose discretizations satisfy the surjectivity conditions.

We also point out that, unlike the methods of [25], the way we start the backward elimination for processor $j < J$ is ad hoc and not particularly efficient. (This need not, however, affect computational performance, as we will see in §4.) Our method is arguably better suited to problems in which x_{N+1} is specified in addition to x_1, in which case subproblem J would have the same form as each of the other subproblems. Such problems are outside the standard framework of unconstrained DTOC, although they can be formulated in constrained DTOC.

4. Computational examples. DDP and the parallel method are coded in Fortran 77 with message passing extensions for an Intel Paragon OSF/1 computer, a MIMD machine. The Paragon used had 60 available processors (compute nodes) and an additional 4 processors dedicated to system administration (service nodes). Each compute node was an i860XP processor running at 50Mhz, with 32Mb memory, and 400Mb/s burst transfer between nodes. The Intel library CLASSPACK was used for basic linear algebra subroutines (BLAS), including matrix multiplication, Gaussian elimination, and banded Gaussian elimination.

Code development and debugging, however, were carried out almost entirely on a Sun SPARCstation 2, using the iPSC simulator of Chinchalkar [3] which is linked to the LINPACK [5] library of BLAS. This simulator is to be recommended for the development of message passing code for Intel machines in a dedicated workstation environment.

We present computational results by solving some problems from the literature. We thank Steve Wright[2] for Fortran subroutines to evaluate the function and derivative values in these problems. Starting points are chosen near local solutions in order to compare local convergence of the (serial) DDP method against the parallel method. Denote by $x_{i,j}$ the jth component of x_i.

Problem 1 (Bertsekas [1]) ($m = 1, n = 2$). Choosing $h = 1/N$,

$$\begin{aligned} \min \quad & h\sum_{i=1}^{N}(6u_i^2 + 2{x_{i+1,1}}^2 + {x_{i+1,2}}^2) \\ \text{subject to} \quad & x_{i+1} = \begin{bmatrix} 1 & h \\ -h & 1 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ h \end{bmatrix} u_i, \quad i = 1, \ldots, N. \end{aligned}$$

[2] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL.

The first state vector is given as $x_1 = (15, 5)$.

Problem 2 (di Pillo et al. [19, Ex. 1]) ($m = 1, n = 2$). Choosing $h = 1/N$,

$$\begin{aligned} l^1(x_1, u_1) &= \frac{5h}{2}\left({x_{1,1}}^2 + {x_{1,2}}^2\right) + 5h{u_1}^2, \\ l^i(x_i, u_i) &= 5h\left({x_{1,1}}^2 + {x_{1,2}}^2 + {u_1}^2\right), \quad i = 2, \ldots, N-1, \\ l^{N+1}(x_{N+1}) &= \frac{5h}{2}\left({x_{1,1}}^2 + {x_{1,2}}^2\right), \\ f^i(x_i, u_i) &= \begin{bmatrix} x_{i,1} + 5h\left[(1 - {x_{i,2}}^2)x_{i,1} - x_{i,2} + u_i\right], \\ x_{i,1} + 5hx_{i,1} \end{bmatrix}, \end{aligned}$$

and $x_1 = (0, 1)$.

Problem 3 (derived from di Pillo et al. [19, Ex. 2]) ($m = 1, n = 1$). Choosing $h = 1/N$,

$$\begin{aligned} \min \quad & h\sum_{i=1}^{N}({x_i}^2 + {u_i}^2) \\ \text{subject to} \quad & x_{i+1} = x_i + h({x_i}^2 - u_i), \quad i = 1, \ldots, N, \end{aligned}$$

with $x_1 = 1$.

The timing and parallel efficiency results, comparing DDP to the parallel algorithm for each of these problems, are classed in three types: (i) results for an average iteration, for $J = 1, 2, 4, 8, 16, 32, 60$ and $N = 500$; (ii) results comparing the time needed to reduce a residual measuring violation of the first order conditions below 10^{-9}, for $J = 1, 2, 4, 8, 16, 32, 60$ and $N = 500$; and (iii) results for an average iteration for $J = 1, 2, 4, 8, 16, 32, 60$ and N such that $N/J = 500$, to test scalability. For type (ii) results, the residual for $J = 1$ is given by the function r, and the residual for $J \geq 2$ is given by $\max\{r^J, v^J\}$; see §3.4 for details. Results of types (i) and (ii) are given in Table 1. Results of type (iii) are given in Table 2.

We summarize the results. Table 1 shows that for problems with few states and controls per stage ($n, m \leq 2$), parallel efficiency degrades rapidly when each node handles fewer than about 60 stages of the original problem. So the parallel method, like others in the literature, will not be efficient for smaller problems if many processors are available. Table 2, however, presents results for problems with hundreds of stages per processor and shows that parallel methods do have a role in large-scale computation.

Consider Table 1, in which Problems 1–3 with $N = 500$ are solved. The rows corresponding to "msec./itn." show the time in milliseconds per iteration of the parallel method ($J \geq 2$) or DDP ($J = 1$), averaged over 10 iterations. The time required for 10 iterations is taken from the second of two consecutive runs on the same problem, without exiting the program. The need for two runs is due to the high variability of the time needed for the first run, whereas the time for the second run remained within a few milliseconds of constant in different invocations of the code. Likewise the time needed for third and even fourth runs remained within a few milliseconds of the time needed for the second run.

The rows in Table 1 corresponding to "# itn." show the number of iterations required for the residual to be less than 10^{-9}. The percentage in the row "% eff." and the column corresponding to J processors is the average time/iteration required by DDP ($J = 1$) divided by the average time/iteration required by J processors and

TABLE 1
Results for $N = 500$.

Problem	(m, n)	Result	Number of Processors (J)						
			1	2	4	8	16	32	60
1	(1,2)	msec./itn.	202.7	149.4	79.3	42.2	27.5	25.1	38.4
		# itn.	2	2	2	2	2	2	2
		% eff.	100	67	63	60	46	25	8
		% rel. eff.	100	67	63	60	46	25	8
2	(1,2)	msec./itn.	241.6	168.7	86.7	47.3	29.9	26.9	39.7
		# itn.	5	5	4	4	5	5	5
		% eff.	100	71	69	63	50	28	10
		% rel. eff.	100	71	87	86	50	28	10
3	(1,1)	msec./itn.	188.5	126.9	65.1	35.4	22.7	20.1	25.9
		# itn.	13	12	12	13	14	15	14
		% eff.	100	74	72	66	51	29	12
		% rel. eff.	100	80	78	66	47	25	11

multiplied by $100/J$. The entry in row "% rel. eff." and column J is calculated by taking the corresponding percentage efficiency, multiplying it by the number of iterations required by DDP, and dividing it by the number of iterations required by the parallel method when using J processors.

In all problems, the time required is actually more for 60 processors than for 32, which shows the limit of the usefulness of parallelism.

Note that Problem 1 is linear-quadratic; hence a solution is found after the first iteration, although it takes a second iteration to calculate the value of r or r^J. Moreover, no linear algebra associated with second derivatives of the transition function is performed when solving this problem, which helps to explain why the efficiency percentages for this problem are slightly less than those of Problems 2 and 3.

For Problems 2 and 3, we observe that the number of iterations required actually decreases as J initially increases. This phenomenon seems to indicate that if, through refining a discretization, the number of stages N grows, a decomposition scheme like the parallel method proposed here may improve numerical stability. However, as the number of groups J making up the decomposition becomes large, we see from the increase in the number of steps of the parallel method that numerical stability is usually reduced.

We note a trend that is not revealed in Table 1: For Problems 1–3, the constraint violation v^J converges to zero much faster than r^J, which measures the projected gradient, and v^J is usually several orders of magnitude less than 10^{-9} after the final iteration (when r^J is, for the first time, below 10^{-9}).

Consider Table 2, in which for a given number of processors J, we solve Problems 1–3 with $N = 500J$. As in Table 1, rows corresponding to "msec./itn." show the time in milliseconds per iteration of the parallel method ($J \geq 2$) or DDP ($J = 1$), averaged over 10 iterations of the second of two consecutive runs of the same problem, and rows "% eff." show the efficiency of the parallel method. The important feature of these results is that the parallel method scales well: as the problem size $500J$ and the number of processors J increase, the parallel efficiency of the method is virtually constant until the number of processors jumps to 60. In other words, the degradation in efficiency as J increases is not significant for problems involving hundreds of time steps per processor when using relatively few (≤ 32) processors.

We mention in passing that another measure of parallelism is speedup, which is the quotient of the time taken by the serial algorithm and the time taken by the

TABLE 2
Results for $N/J = 500$.

Problem	(m, n)	Result	Number of Processors (J)						
			1	2	4	8	16	32	60
1	(1,2)	msec./itn.	202.3	297.5	299.2	303.0	304.4	311.6	330.6
		% eff.	100	68	67	66	66	64	61
2	(1,2)	msec./itn.	241.5	336.2	337.4	339.4	343.0	350.7	371.1
		% eff.	100	71	71	71	70	68	65
3	(1,1)	msec./itn.	188.4	253.5	254.8	258.3	261.1	264.8	274.3
		% eff.	100	74	72	72	72	71	68

parallel algorithm (= %efficiency $\times J/100$). For instance, looking at Problem 1 in Table 1, the speedup for $J = 2$ is 1.35, and the speedup for $J = 60$ is 5.27. Linear speedup, that is, speedup J for J processors, is perfect speedup, whereas perfect efficiency is 100%.

The above parallel efficiency results are among the best seen for any stagewise decomposition method; see [25, 24, 21, 2], in which efficiencies above 70% are relatively high. Efficiencies of 25% are obtained in [24] when solving Problems 1 and 3 with $J = 8$ and $N = 1000$, using parallel banded factorization on the Alliant FX/8. Efficiency of 54% is obtained in [25] for Problem 1 with $J = 8$ and $N = 2000$, using the "PDP-RI" (partitioned dynamic programming-Ricatti) stagewise decomposition method on the Alliant FX/8. From Table 1, we obtained efficiencies of 60–68% for these problems with $J = 8$ and $N = 500$. In [2], parallel efficiency of 68% is shown by simulating parallel computation with $J = 3$ on a serial computer, for a problem with $N = 336$.

Direct comparisons of our efficiency results with other results in the literature, however, are of limited value because the speed of a processor relative to the speed of communication depends on the computer. For example, preliminary computation of the parallel DTOC method on an Intel iPSC/860 hypercube gave higher efficiencies than the results obtained on the Paragon (up to 3 percentage points higher for $J = 2$ and up to 13 percentage points higher for $J = 32$) because the hypercube nodes were approximately twice as slow at arithmetic as the Paragon nodes, whereas time needed for communication was similar on both machines. We also recall from §3 that our parallel method, similar to others, requires around one third more flops per iteration than DDP requires; hence 75% is a rule of thumb for the best efficiency possible if linear algebra takes almost all the time used in computation.

Acknowledgments. I thank Tom Coleman for introducing me to the area of DTOC and supporting this project in the Advanced Computing Research Institute at Cornell University, where this manuscript has appeared as a technical report (CTC93TR118, 02/93), and also Li-Zhi Liao for his expertise in DDP. This paper has also benefitted from valuable discussions with Shirish Chinchalkar, Mark Psiaki, and Steve Wright and detailed comments from anonymous referees.

REFERENCES

[1] D. P. BERTSEKAS, *Projected Newton's method for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[2] S. C. CHANG, T. S. CHANG, AND P. B. LUH, *A hierarchical decomposition for large-scale optimal control problems with parallel processing structure*, Automatica, 25 (1989), pp. 77–86.

[3] S. CHINCHALKAR, *A hypercube simulator (version* 2.0), Cornell University, Ithaca, NY, 1992, manuscript.
[4] T. B. CULVER AND C. A. SHOEMAKER, *Dynamic optimal control for groundwater remediation with flexible management periods*, Water Resources Research, 28 (1992), pp. 629–641.
[5] J. J. DONGARRA, C. B. MOLER, J. R. BUNCH, AND G. W. STEWART, *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.
[6] J. C. DUNN, *A projected Newton method for minimization problems with nonlinear inequality constraints*, Numer. Math., 53 (1988), pp. 377–409.
[7] J. C. DUNN AND D. P. BERTSEKAS, *Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems*, J. Optim. Theory Appl., 63 (1989), pp. 23–89.
[8] R. FLETCHER, *Practical Methods of Optimization*, Wiley-Interscience, New York, 1987.
[9] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[10] D. JACOBSON AND D. MAYNE, *Differential Dynamic Programming*, Elsevier Science Publications, New York, 1970.
[11] L.-Z. LIAO AND C. A. SHOEMAKER, *Convergence in unconstrained discrete-time differential dynamic programming*, IEEE Trans. Automat. Control, 36 (1991), pp. 692–706.
[12] ———, *Advantages of Differential Dynamic Programming Over Newton's Method for Discrete-time Optimal Control Problems*, Tech. report CTC92 TR97, Cornell Theory Center, Ithaca, NY, 1992.
[13] D. MAYNE, *A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems*, Internat. J. Control, 3 (1966), pp. 85–95.
[14] A. MIELE AND W. Y. LEE, *Optimal trajectories for hypervelocity flight*, in Proc. 1989 American Control Conference, Pittsburgh, PA, 3, 1989.
[15] D. M. MURRAY, *Differential Dynamic Programming for the Efficient Solution of Optimal Control Problems*, Ph.D. thesis, Department of Mathematics, University of Arizona, Tucson, AZ, 1978.
[16] D. M. MURRAY AND S. J. YAKOWITZ, *Differential dynamic programming and Newton's method for discrete optimal control problems*, J. Optim. Theory Appl., 43 (1984), pp. 395–414.
[17] J. NOCEDAL AND M. L. OVERTON, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.
[18] J .F. A. DE O. PANTOJA, *Differential dynamic programming and Newton's method*, Internat. J. Control, 47 (1988), pp. 1539–1553.
[19] G. DI PILLO, L. GRIPPO, AND F. LAMPARIELLO, *A class of structured quasi-Newton algorithms for optimal control problems*, in IFAC Applications of Nonlinear Programming to Optimization and Control, 1983.
[20] E. POLAK, *Computational Methods in Optimization*, Academic Press, New York, 1970.
[21] M. L. PSIAKI AND K. PARK, *A parallel solver for trajectory optimization search directions*, J. Optim. Theory Appl., 73 (1992), pp. 519–546.
[22] C. A. SHOEMAKER, L.-Z. LIAO, B. AUBERT, AND J. F. ABEL, *Optimal nonlinear control with geometrically nonlinear finite element analysis for flexible structures*, in Computational Mechanics, A. N. Atluri and G. Yagawa, eds., Springer-Verlag, Berlin, New York, 1988.
[23] C. A. SHOEMAKER, L.-Z. LIAO, H. M. CAFFEY, AND L.-C. CHANG, *Optimal control of nonlinear engineering systems*, in Computer Assisted Modeling on the IBM 3690: Contest Prize Papers, Vol. 1, Baldwin Press, University of Georgia, 1992, pp. 403–446.
[24] S. J. WRIGHT, *Solution of discrete-time optimal control problems on parallel computers*, Parallel Comput., 16 (1990), pp. 221–237.
[25] ———, *Partitioned dynamic programming for optimal control*, SIAM J. Optim., 1 (1991), pp. 620–642.
[26] ———, *Parallel algorithms for banded linear systems*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 824–842.
[27] ———, *Interior-point methods for optimal control of discrete-time systems*, J. Optim. Theory Appl., 77 (1993), pp. 161–187.
[28] S. YAKOWITZ AND B. RUTHERFORD, *Computational aspects of discrete-time optimal control*, Appl. Math. Comput., 15 (1984), pp. 29–45.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 513–530, May 1996
© 1996 Society for Industrial and Applied Mathematics
013

A GLOBAL SEARCH METHOD FOR DISCRETE STOCHASTIC OPTIMIZATION*

SIGRÚN ANDRADÓTTIR†

Abstract. This paper is concerned with the problem of optimizing the performance of a stochastic system over a finite set of alternatives in situations where the performance of the system cannot be evaluated analytically, but must be estimated or measured, for instance, through simulation. We present two variants of a new method for solving such discrete stochastic optimization problems. This new method uses global search to look for the optimal solution. It generates a sequence taking values in the set of feasible alternatives, where each new element of the sequence is generated by comparing the current element with another candidate alternative and letting the next element of the sequence be the one of the current and candidate alternatives that appears to yield better performance. For both versions of the proposed method, the element of the set of feasible alternatives that the generated sequence visits most often is shown to converge almost surely to a globally optimal solution of the underlying optimization problem. Thus the method spends most of the computational effort at the global optimizer. We also show how one variant of the proposed method can be used to solve discrete optimization problems in both transient and steady-state simulation, show how the other variant can be used to optimize probabilities, and present some numerical results.

Key words. global optimization, simulation, Markov chains

AMS subject classifications. 60F15, 60J20, 62E25, 90C15, 90C50

1. Introduction.

Consider the problem of finding the element of a finite set of alternative system configurations that optimizes the performance of the system in situations where the performance of the system cannot be evaluated analytically, but must be measured or estimated, for instance, through simulation. Such design problems occur frequently in practice. For instance, the problem of determining the optimal buffer allocation in a manufacturing network is an optimization problem of this form. Since the evaluation of the performance of the different system configurations includes noise, conventional (deterministic) optimization methods cannot be used to solve this optimization problem. Instead, it is necessary to develop specialized methods that can be used to determine the optimal system configuration despite the presence of noise. In this paper, we present and analyze a new method for discrete stochastic optimization, and we show how this method can be applied to solve simulation optimization problems that involve discrete decision parameters.

Simulation optimization with respect to discrete decision variables is currently a relatively undeveloped field. However, a fair amount of related literature is available. When the number of different alternatives is small, methods for selecting the best system can be used to solve discrete stochastic optimization problems. Goldsman, Nelson, and Schmeiser [10] present a brief overview of three methods for selecting the best system: interactive analysis, ranking and selection, and multiple comparisons. For more details on ranking and selection procedures, see Gibbons, Olkin, and Sobel [9] and Gupta and Panchapakesan [12]. For more details on multiple comparison procedures, see Hochberg and Tamhane [13].

The multiarmed bandit problem is also related to discrete stochastic optimization. However, while discrete stochastic optimization is concerned with determining the

* Received by the editors December 15, 1993; accepted for publication (in revised form) November 17, 1994. This research was supported by National Science Foundation grant DDM–9210679.

† Department of Industrial Engineering, University of Wisconsin–Madison, 1513 University Avenue, Madison, WI 53706. Current address: School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

parameter value that optimizes the objective function, the multiarmed bandit problem involves optimizing the average observed function value over a period of time. More specifically, it is assumed that one can sequentially draw observations of the function values at the different parameter values and that the value of the objective function is the expected reward associated with obtaining an observation at the parameter value in question. The objective is to draw these observations in such a way that the observed reward per unit time is optimized. Robbins [19] presented a policy for obtaining observations of the objective function in which the average observed value of the objective function per unit time converges to the optimal value of the objective function as time goes to infinity. Later Lai and Robbins [17] developed optimal policies for some specific families of probability distributions. For more recent work on the multiarmed bandit problem, see, for example, Agrawal [1] and Yakowitz and Lowe [24].

Another related body of research has appeared in the learning automata literature. Here it is of interest to draw samples from a finite number of alternative populations in such a way that the probability of sampling from the best population converges to one as the total number of samples goes to infinity. Tsetlin's pioneering paper [21] looks at the situation where the observations can take only two values, "penalty" or "nonpenalty," and the best population is the one with the lowest probability of observing a penalty. This situation is also referred to as the P-model. Later the Q-model, where observations can take more than two, but finitely many values, and the S-model, where the observations can take values in the unit interval, were also considered. Relevant references include, for example, Shapiro and Narendra [20], Viswanathan and Narendra [22], Lakshmivarahan and Thathachar [16], and Devroye [8]. In addition, although most random search methods are designed to solve deterministic optimization problems with continuous decision variables, related work includes Devroye [6, 7] and Yakowitz and Lugosi [23]. Many of the methods we have discussed resemble the method discussed in this paper in that they generate a (homogeneous or nonhomogeneous) Markov chain on the set of feasible parameter values (populations). However, they differ from the method discussed here in terms of the objective of the study (we are interested only in determining the best population, whereas they are also interested in maximizing the probability of sampling from the optimal population), in terms of the criteria used to estimate the optimal solution (our estimate of the solution is always the population from which the most samples have been drawn so far, even if no samples from this population were drawn in the last iteration), and in terms of the specific Markov chain generated. In addition, to our knowledge, the application of these methods to optimization in discrete event simulation has not been explored.

Recently, Yan and Mukai [25] have proposed a method for discrete stochastic optimization that is related to simulated annealing. Instead of solving the original optimization problem, this method solves an alternative optimization problem whose solution agrees with the solution of the original optimization problem under certain conditions. Essentially, the method proposed by Yan and Mukai compares the observations of the objective function with the observations of a predetermined uniform random variable (a "stochastic ruler"). If the range of this uniform random variable includes all possible observations of the objective function, then the original and alternative optimization problems are equivalent. Yan and Mukai show that under certain conditions, their method converges in probability to a global solution of the alternative optimization problem.

Gong, Ho, and Zhai [11] have also proposed and analyzed a method for discrete stochastic optimization. Their method also solves an alternative optimization problem instead of solving the original optimization problem. However, the alternative optimization problem solved by Gong, Ho, and Zhai is different from the alternative optimization problem solved by Yan and Mukai. Essentially, the method proposed by Gong, Ho, and Zhai differs from the method proposed by Yan and Mukai in that Gong, Ho, and Zhai use a specific neighborhood structure (they assume that any two parameter values are neighbors), and in each iteration, they compare the observations of the objective function at two different parameter values (rather than using a stochastic ruler). Gong, Ho, and Zhai show that the original and alternative optimization problems they consider are equivalent if the noise in the evaluations of the objective function at two different parameter values is independent and identically distributed and if the noise terms have a symmetric and continuous probability density function with mean zero for each different parameter value. They also show that their method converges in probability to a global solution of the alternative optimization problem.

Andradóttir [2, 3] has also proposed a method for discrete stochastic optimization. In each iteration of this method, two neighboring parameter values are compared and the one that appears to be better (has a better observed objective function value) is passed on to the next iteration. Unlike the methods proposed by Yan and Mukai and by Gong, Ho, and Zhai, the basic method proposed by Andradóttir uses only one observation of the objective function at each parameter value in every iteration. (The potential number of observations of the objective function per iteration goes to infinity as the number of iterations increases in the methods proposed by Yan and Mukai and by Gong, Ho, and Zhai.) Andradóttir proves that her method converges almost surely to a local optimizer of the objective function under certain conditions (more general than the conditions required by Gong, Ho, and Zhai; see their Lemma 1). She also discusses how her method can be applied to solve optimization problems in both transient and steady-state simulation. For this purpose, she introduces an extension of the procedure that can be applied in certain situations where the basic procedure is not guaranteed to converge. (Unlike the basic method proposed by Andradóttir, this extension of the basic method resembles the methods proposed by Yan and Mukai and by Gong, Ho, and Zhai in that it requires an increasing number of observations of the objective function per iteration as the number of iterations increases.)

In this paper we present and analyze a new method for discrete stochastic optimization. This method uses the same idea as the method proposed by Andradóttir discussed previously, except that Andradóttir's method exploits neighbor-information, whereas the new method assumes that any two points are neighbors. Thus, every iteration of the older procedure consists of a *local* search for a point that is better than the current point, whereas every iteration of the method proposed in this paper consists of a *global* search for a better point. Because this ensures that a candidate solution is compared with all the other alternatives (and not only with its neighbors), the proposed method can be shown to converge to a *global* optimizer of the objective function under certain conditions. We therefore expect that the new method will perform better than the older method in situations where the underlying problem does not have a natural neighborhood structure and where the objective function has many local minimizers. The organization of the paper is as follows. In §2, we present the proposed method and prove that under certain conditions, the new method converges almost surely to a global solution of the underlying optimization problem. In §3, we present and analyze a generalized version of this method and discuss how it can be

applied to solve discrete optimization problems in both transient and steady-state simulation. In §4, we show how the proposed method can be applied to optimize a special class of objective functions. Finally, §5 contains some concluding remarks.

2. An algorithm for discrete stochastic optimization. Suppose we are interested in solving a stochastic optimization problem involving a finite number (say K) of different alternatives. Let $\mathcal{K} = \{1, 2, \ldots, K\}$. By relabeling the states (if necessary), we can assume, without loss of generality, that the set of feasible alternatives is $\mathcal{K}$. So consider the following optimization problem:

$$\min_{n \in \mathcal{K}} f(n) = E\{X_n\}, \tag{1}$$

where $\{X_n\}$ is a sequence of random variables. Let $\mathcal{S} \subset \mathcal{K}$ denote the set of (global) minimizers of the function f. When the expected values $E\{X_n\}$, $n = 1, \ldots, K$, can be evaluated analytically, the optimization problem (1) can be solved using standard integer programming techniques. However, these techniques cannot be applied in situations where the objective function f cannot be evaluated analytically, but must be measured or estimated, for instance, through simulation. We present a new method that is designed to locate an element of the set $\mathcal{S}$ in situations where the expected values $E\{X_n\}$, $n = 1, \ldots, K$, cannot be evaluated analytically. We need the following assumption.

Assumption 1. For each $i, j \in \mathcal{K}$, there exists a random variable $Y^{(i \to j)}$ such that for all $i \in \mathcal{S}$, $j \notin \mathcal{S}$, and $n \in \mathcal{K}$, $n \neq i, j$,

$$P\{Y^{(j \to i)} > 0\} > P\{Y^{(i \to j)} > 0\}, \tag{2}$$

$$P\{Y^{(n \to i)} > 0\} \geq P\{Y^{(n \to j)} > 0\}, \tag{3}$$

and

$$P\{Y^{(i \to n)} \leq 0\} \geq P\{Y^{(j \to n)} \leq 0\}. \tag{4}$$

Remark. Gong, Ho, and Zhai [11] show that the alternative optimization problem they solve is equivalent to the original optimization problem (1) if $X_n = f(n) + W_n$ for all $n \in \mathcal{K}$, where $\{W_n : n \in \mathcal{K}\}$ are random variables that satisfy the following two conditions: (a) $\{W_n : n \in \mathcal{K}\}$ are independent and identically distributed, and (b) each W_n, where $n \in \mathcal{K}$, has a symmetric, continuous probability density function with zero mean. By the proofs of Lemmas 1 and 2 in Gong, Ho, and Zhai, this assumption implies that Assumption 1 holds (take $Y^{(i \to j)} = X_i - X_j$ for all $i, j \in \mathcal{K}$).

For all $i, j \in \mathcal{K}$, the random variable $Y^{(i \to j)}$ will be used to compare state i with state j. If in a given iteration of the algorithm we have decided to compare the current point i with an alternative point j, then we conduct an experiment or a simulation to obtain an observation of the random variable $Y^{(i \to j)}$. (Typically, the random variable $Y^{(i \to j)}$ is a function only of measurements taken at the points i and j.) If $Y^{(i \to j)} \leq 0$, then we let i be our next point, and if $Y^{(i \to j)} > 0$, then we let j be the next point. Now suppose that $i \in \mathcal{S}$ and $j \notin \mathcal{S}$. Then equation (2) states that the probability of moving from j to i (and thereby making a correct decision) is larger than the probability of moving from i to j (and thereby making an incorrect decision). Equation (3) states that if we are currently at point $n \neq i, j$ and we have decided to compare the point n with the point k, then the probability of moving away from the current point n is larger if $k = i$ than if $k = j$. Finally, equation (4) states that if we are currently at a point k and we want to compare the current point with

the point $n \neq i, j$, then the probability of staying at the current point k is larger when $k = i$ than when $k = j$. The details of the algorithm for solving the optimization problem (1) are given below. Note that after m iterations, k_m is the current point, for all $n \in \mathcal{K}$, $N_m(n)$ is the number of times the algorithm has visited state n so far, and k_m^* is the state that the algorithm has visited most often so far.

ALGORITHM 2.1.

Step 0: *Select a starting point* $k_0 \in \mathcal{K}$. *Let* $N_0(k_0) = 1$ *and* $N_0(n) = 0$ *for all* $n \in \mathcal{K}, n \neq k_0$. *Let* $m = 0$ *and* $k_m^* = k_0$. *Go to step* 1.

Step 1: *Generate a uniform random variable* J_m *such that for all* $n \in \mathcal{K}$, $n \neq k_m$, $J_m = n$ *with probability* $\frac{1}{K-1}$. *Go to step* 2.

Step 2: *Generate an observation* R_m *of* $Y^{(k_m \to J_m)}$. *If* $R_m > 0$, *then let* $k_{m+1} = J_m$. *Otherwise, let* $k_{m+1} = k_m$. *Go to step* 3.

Step 3: *Let* $m = m + 1$, $N_m(k_m) = N_{m-1}(k_m) + 1$, *and* $N_m(n) = N_{m-1}(n)$ *for all* $n \in \mathcal{K}, n \neq k_m$. *If* $N_m(k_m) > N_m(k_{m-1}^*)$, *then let* $k_m^* = k_m$. *Otherwise, let* $k_m^* = k_{m-1}^*$. *Go to step* 1.

Remarks. 1. It is clearly not necessary to store the sequences $\{k_m\}$, $\{k_m^*\}$, $\{J_m\}$, $\{R_m\}$, and $\{N_m(n)\}$, where $n \in \mathcal{K}$. In each iteration, it is possible to overwrite the earlier values since they will not be used again by the algorithm.

2. When the state space $\mathcal{K}$ is large, an implementation of Algorithm 2.1 may need a large amount of memory since it is necessary to store the values of $N_m(n)$ for all $n \in \mathcal{K}$.

3. As was mentioned in §1, the primary difference between the algorithms discussed in this paper and the algorithms proposed by Andradóttir [2, 3] is the choice of neighborhood structure. The algorithms discussed by Andradóttir [2, 3] assume that each point has at most two neighbors, whereas the algorithms discussed in this paper resemble the algorithm proposed by Gong, Ho, and Zhai [11] in that any two points are assumed to be neighbors. Thus, unlike simulated annealing and the method proposed by Yan and Mukai [25], the methods discussed in this paper and the methods proposed by Andradóttir [2, 3] and by Gong, Ho, and Zhai [11] all involve assuming that the neighborhood structure has a specific form. It would be of interest to extend this work to cover more general neighborhood structures.

The sequence $\{k_m\}$ generated by Algorithm 2.1 is a Markov chain on the state space $\mathcal{K}$ with transition probability matrix $\mathcal{P} = [\mathcal{P}(i,j)]$, where

$$\mathcal{P}(i,j) = P\{k_{m+1} = j | k_m = i\} = \frac{1}{K-1} P\{Y^{(i \to j)} > 0\} \tag{5}$$

for all $i, j \in \mathcal{K}$, $i \neq j$, and

$$\begin{aligned} \mathcal{P}(i,i) = P\{k_{m+1} = i | k_m = i\} &= 1 - \sum_{j \in \mathcal{K} \setminus \{i\}} \mathcal{P}(i,j) \\ &= 1 - \frac{1}{K-1} \sum_{j \in \mathcal{K} \setminus \{i\}} P\{Y^{(i \to j)} > 0\} \\ &= \frac{1}{K-1} \sum_{j \in \mathcal{K} \setminus \{i\}} [1 - P\{Y^{(i \to j)} > 0\}] \\ &= \frac{1}{K-1} \sum_{j \in \mathcal{K} \setminus \{i\}} P\{Y^{(i \to j)} \leq 0\} \end{aligned} \tag{6}$$

for all $i \in \mathcal{K}$. (This of course assumes that R_m is independent of $R_0, \ldots, R_{m-1}$ for all $m \in \mathbb{N}$.)

In general, one would not expect the Markov chain $\{k_m\}$ to converge almost surely to an element of the set $\mathcal{S}$. In fact, the Markov chain $\{k_m\}$ can be irreducible under Assumption 1 and may thereby visit each element of the set $\mathcal{K}$ infinitely often. Instead, we show that the sequence $\{k_m^*\}$ converges almost surely to an element of $\mathcal{S}$, where k_l^* is the parameter value that the Markov chain $\{k_m\}$ has visited most often after l iterations of the algorithm, $l = 0, 1, 2, \ldots$.

THEOREM 2.1 (convergence of Algorithm 2.1). *If Assumption* 1 *is satisfied, then the sequence* $\{k_m^*\}$ *generated by Algorithm* 2.1 *converges almost surely to an element of* $\mathcal{S}$ *(in the sense that there exists a set* A *such that* $P(A) = 1$ *and for all* $\omega \in A$, *there exists* $M_\omega > 0$ *such that* $k_m^*(\omega) \in \mathcal{S}$ *for all* $m \geq M_\omega$*).*

Remarks. 1. Note that as long as Assumption 1 is satisfied, Theorem 2.1 shows that it is not necessary to assume that the objective function f of equation (1) is finite for all $n \in \mathcal{K}$ or that the noise in the evaluations of the objective function is bounded in any sense in order to guarantee the convergence of Algorithm 2.1. Such assumptions are frequently needed to prove the convergence of stochastic optimization procedures. For example, Yan and Mukai [25] assume that the noise in the evaluations of the objective function has a bounded variance (assumption A0), and they show that if the (noisy) evaluations of the objective function are bounded, with known upper and lower bounds, then the solution to the alternative optimization problem they solve agrees with the solution to the original optimization problem (Theorem 3.2). Similarly, Gong, Ho, and Zhai [11] assume that the noise in the evaluations of the objective function has finite variance at all points in the search space.

2. It should be pointed out that the reason Algorithm 2.1 converges is very much related to the reason why simulated annealing with a fixed temperature spends most of its time at a global minimum on a complete graph (see, for example, Metropolis et al. [18]).

Proof. Suppose the Markov chain $\{k_m\}$ is irreducible. Let $\pi_1, \ldots, \pi_K$ denote the stationary distribution of $\{k_m\}$. By equations (5) and (6), we have that for all $k \in \mathcal{K}$,

$$\begin{aligned}\pi_k &= \sum_{n=1}^{K} \mathcal{P}(n,k)\pi_n \\ &= \frac{1}{K-1} \sum_{n \in \mathcal{K}\setminus\{k\}} P\{Y^{(n\to k)} > 0\}\pi_n + \frac{\pi_k}{K-1} \sum_{n \in \mathcal{K}\setminus\{k\}} P\{Y^{(k\to n)} \leq 0\}.\end{aligned}$$

Suppose that $i \in \mathcal{S}$ and $j \notin \mathcal{S}$. By Assumption 1, we have

$$\begin{aligned}\pi_j \leq{}& \frac{1}{K-1} \sum_{n \in \mathcal{K}\setminus\{i,j\}} P\{Y^{(n\to i)} > 0\}\pi_n + \frac{\pi_j}{K-1} \sum_{n \in \mathcal{K}\setminus\{i,j\}} P\{Y^{(i\to n)} \leq 0\} \\ &+ \frac{\pi_i}{K-1} P\{Y^{(i\to j)} > 0\} + \frac{\pi_j}{K-1} P\{Y^{(j\to i)} \leq 0\},\end{aligned}$$

so that

$$\begin{aligned}\pi_i - \pi_j \geq{}& \frac{\pi_i - \pi_j}{K-1} \sum_{n \in \mathcal{K}\setminus\{i,j\}} P\{Y^{(i\to n)} \leq 0\} \\ &+ \frac{\pi_j}{K-1} \Big[P\{Y^{(j\to i)} > 0\} - P\{Y^{(j\to i)} \leq 0\}\Big]\end{aligned}$$

$$
\begin{aligned}
&+ \frac{\pi_i}{K-1}\left[P\{Y^{(i\to j)} \le 0\} - P\{Y^{(i\to j)} > 0\}\right] \\
&= \frac{\pi_i - \pi_j}{K-1} \sum_{n\in\mathcal{K}\setminus\{i,j\}} P\{Y^{(i\to n)} \le 0\} + \frac{\pi_j}{K-1}\left[2P\{Y^{(j\to i)} > 0\} - 1\right] \\
&\quad + \frac{\pi_i}{K-1}\left[1 - 2P\{Y^{(i\to j)} > 0\}\right] \\
&> \frac{M}{K-1}(\pi_i - \pi_j), \qquad (7)
\end{aligned}
$$

where

$$
M = \sum_{n\in\mathcal{K}\setminus\{i,j\}} P\{Y^{(i\to n)} \le 0\} + 1 - 2P\{Y^{(i\to j)} > 0\} \le K-1. \qquad (8)
$$

From equation (7), it is clear that $\pi_i \neq \pi_j$. If $\pi_i < \pi_j$, then equation (7) gives $M > K-1$. This contradicts equation (8). We have shown that $\pi_i > \pi_j$. Since $i \in \mathcal{S}$ and $j \notin \mathcal{S}$ are arbitrary, this shows that $\arg\max \pi_n \in \mathcal{S}$.

For all n, let e_n denote the expected number of iterations between two successive visits by the Markov chain $\{k_m\}$ to state n. Then it is well known that $\pi_n = 1/e_n$ for all n. Furthermore, $N_m(n)$ (as a function of m) is the (general) renewal process defined by the number of times the Markov chain $\{k_m\}$ enters the state n. Therefore, by the elementary renewal theorem, $N_m(n)/m \to 1/e_n = \pi_n$ almost surely (a.s.) as $m \to \infty$. Now let A be such that $P(A) = 1$ and for all $\omega \in A$, $N_m(n,\omega)/m \to \pi_n$ as $m \to \infty$, for all $n \in \mathcal{K}$. Let $\omega \in A$. Then, since $\arg\max \pi_n \in \mathcal{S}$, there exists $M_\omega > 0$ such that for all $m \ge M_\omega$, $k_m^*(\omega) \in \mathcal{S}$. This completes the proof in the case when the Markov chain $\{k_m\}$ is irreducible.

Now suppose that the Markov chain $\{k_m\}$ is reducible. Suppose $j \notin \mathcal{S}$. If j is transient, then $N_m(j)/m \to 0$ a.s. as $m \to \infty$, so with probability one, the sequence $\{k_m^*\}$ will not converge to j as $m \to \infty$. On the other hand, when j is positive recurrent, since $\mathcal{P}(j,i) > 0$ for all $i \in \mathcal{S}$ by Assumption 1, j and all the elements of $\mathcal{S}$ belong to the same equivalence class. By applying the argument that was used for the case when the Markov chain $\{k_m\}$ is irreducible to this equivalence class, we see that the sequence $\{k_m^*\}$ converges almost surely to an element of $\mathcal{S}$ as $m \to \infty$. □

The main difficulty in applying Algorithm 2.1 is coming up with random variables $Y^{(i\to j)}$, for all $i, j \in \mathcal{K}$, that satisfy Assumption 1. In some cases, choosing $Y^{(i\to j)} = X_i - X_j$ for all i, j works. In other cases, the choice of $Y^{(i\to j)}$ is less straight-forward. In §4, we propose choices of $Y^{(i\to j)}$ that work when the objective function f (see equation (1)) is of the form $f(n) = bE\{I_n\}$, for all $n \in \mathcal{K}$, where $b \neq 0$ and I_n is an indicator random variable (so $E\{I_n\}$ is a probability) for all $n \in \mathcal{K}$. In §3, we present a generalization of Algorithm 2.1 that can be applied to solve discrete optimization problems in both transient and steady-state simulation.

3. Discrete optimization in simulation. As mentioned in §2, the scope of applicability of Algorithm 2.1 is restricted to discrete stochastic optimization problems for which random variables $Y^{(i\to j)}$, $i, j \in \mathcal{K}$, that satisfy Assumption 1 can be found. In this section, we present a generalized version of Algorithm 2.1 that can be applied to solve discrete optimization problems of the form (1) in both transient and steady-state simulation. The basic idea is that if in iteration m we want to compare states i and j, then instead of using the random variable $Y^{(i\to j)}$ to decide whether i or j should be the next state, we will use the random variable $Z_{l_m}^{(i\to j)}$ for this purpose, where $l_m \to \infty$ as $m \to \infty$ (see Assumption 3). The random variables $Z_l^{(i\to j)}$ are chosen so

that Assumption 1 is satisfied in the limit as $l \to \infty$ (see Assumption 2). In §§3.1 and 3.2, we show how sequences of random variables $\{Z_l^{(i\to j)}\}$ that satisfy Assumption 2 can be obtained for discrete stochastic optimization problems where the objective function is evaluated through transient and steady-state simulation, respectively.

Assumptions 2 and 3 are now stated formally.

Assumption 2. For each $i, j \in \mathcal{K}$ and $l \in \mathbb{N}$, there exists a random variable $Z_l^{(i\to j)}$ such that the limit $\lim_{l\to\infty} P\{Z_l^{(i\to j)} > 0\}$ exists for all $i, j \in \mathcal{K}$, and for all $i \in \mathcal{S}$, $j \notin \mathcal{S}$, $n \in \mathcal{K}$, $n \neq i, j$, and $l \in \mathbb{N}$,

$$\lim_{l\to\infty} P\{Z_l^{(j\to i)} > 0\} > \lim_{l\to\infty} P\{Z_l^{(i\to j)} > 0\}, \tag{9}$$

$$\lim_{l\to\infty} P\{Z_l^{(n\to i)} > 0\} \geq \lim_{l\to\infty} P\{Z_l^{(n\to j)} > 0\}, \tag{10}$$

and

$$\lim_{l\to\infty} P\{Z_l^{(i\to n)} \leq 0\} \geq \lim_{l\to\infty} P\{Z_l^{(j\to n)} \leq 0\}. \tag{11}$$

Assumption 3. $\{l_m\}$ is a sequence of positive integers such that $l_m \to \infty$ as $m \to \infty$.

We now state the generalized version of Algorithm 2.1.

ALGORITHM 3.1.

Step 0: *Select a starting point* $k_0 \in \mathcal{K}$. *Let* $N_0(k_0) = 1$ *and* $N_0(n) = 0$ *for all* $n \in \mathcal{K}, n \neq k_0$. *Let* $m = 0$ *and* $k_m^* = k_0$. *Go to step* 1.

Step 1: *Generate a uniform random variable* J_m *such that for all* $n \in \mathcal{K}$, $n \neq k_m$, $J_m = n$ *with probability* $\frac{1}{K-1}$. *Go to step* 2.

Step 2: *Generate an observation* R_m *of* $Z_{l_m}^{(k_m\to J_m)}$. *If* $R_m > 0$, *then let* $k_{m+1} = J_m$. *Otherwise, let* $k_{m+1} = k_m$. *Go to step* 3.

Step 3: *Let* $m = m + 1$, $N_m(k_m) = N_{m-1}(k_m) + 1$, *and* $N_m(n) = N_{m-1}(n)$ *for all* $n \in \mathcal{K}, n \neq k_m$. *If* $N_m(k_m) > N_m(k_{m-1}^*)$, *then let* $k_m^* = k_m$. *Otherwise, let* $k_m^* = k_{m-1}^*$. *Go to step* 1.

Recall that the sequence $\{k_m\}$ generated by Algorithm 2.1 is a time-homogeneous Markov chain. On the other hand, the sequence $\{k_m\}$ generated by Algorithm 3.1 is a nonhomogeneous Markov chain (assuming that R_m is independent of $R_0, \ldots, R_{m-1}$ for all $m \in \mathbb{N}$). The convergence theory for Algorithm 3.1 is therefore more difficult than the convergence theory for Algorithm 2.1. We will need the following lemma that is proved in Andradóttir [3].

LEMMA 3.1 (time averages for nonhomogeneous Markov chains). *Suppose that* $\mathcal{P}_m$, $m = 0, 1, 2, \ldots$, *and* $\mathcal{P}$ *are Markov matrices on the state-space* $\mathcal{K}$ *such that* $\mathcal{P}$ *is irreducible and aperiodic and* $\mathcal{P}_m \to \mathcal{P}$ *as* $m \to \infty$. *If* $g : \mathcal{K} \to \mathbb{R}$, *then*

$$\frac{1}{M} \sum_{m=1}^{M} g(D_m) \to \sum_{d=1}^{K} \pi_d g(d) \text{ a.s. as } M \to \infty,$$

where $\pi^T = (\pi_1, \ldots, \pi_K)$ *is the steady-state distribution corresponding to* $\mathcal{P}$ *and*

$$P\{D_{m+1} = d | D_0, \ldots, D_m\} = \mathcal{P}_m(D_m, d)$$

for all $d \in \mathcal{K}$ *and* $m = 0, 1, 2, \ldots$.

When Assumption 2 holds, we can define a Markov matrix $\mathcal{P} = [\mathcal{P}(i,j)]$ by the following equations:

$$\mathcal{P}(i,j) = \frac{1}{K-1} \lim_{l\to\infty} P\{Z_l^{(i\to j)} > 0\} \tag{12}$$

for all $i, j \in \mathcal{K}$, $i \neq j$, and

$$\mathcal{P}(i,i) = \frac{1}{K-1} \sum_{j\in\mathcal{K}\setminus\{i\}} \lim_{l\to\infty} P\{Z_l^{(i\to j)} \leq 0\} \tag{13}$$

for all $i \in \mathcal{K}$. We need the following assumption.

Assumption 4. The Markov matrix $\mathcal{P}$ of equations (12) and (13) is irreducible.

We can now prove the following result.

THEOREM 3.2 (convergence of Algorithm 3.1). *If Assumptions* 2, 3, *and* 4 *are satisfied, then the sequence* $\{k_m^*\}$ *generated by Algorithm* 3.1 *converges almost surely to an element of* $\mathcal{S}$ *(in the sense that there exists a set* A *such that* $P(A) = 1$ *and for all* $\omega \in A$, *there exists* $M_\omega > 0$ *such that* $k_m^*(\omega) \in \mathcal{S}$ *for all* $m \geq M_\omega$*).*

Proof. It is clear that $\{k_m\}$ is a nonhomogeneous Markov chain with transition probabilities

$$\mathcal{P}_m(i,j) = P\{k_{m+1} = j | k_m = i\} = \frac{1}{K-1} P\{Z_{l_m}^{(i\to j)} > 0\}$$

for all $i, j \in \mathcal{K}$, $i \neq j$, and $m = 0, 1, 2, \ldots$, and

$$\mathcal{P}_m(i,i) = P\{k_{m+1} = i | k_m = i\} = \frac{1}{K-1} \sum_{j\in\mathcal{K}\setminus\{i\}} P\{Z_{l_m}^{(i\to j)} \leq 0\}$$

for all $i \in \mathcal{K}$ and $m = 0, 1, 2, \ldots$. As $m \to \infty$, it is clear from Assumptions 2 and 3 that $\mathcal{P}_m \to \mathcal{P}$ as $m \to \infty$, where $\mathcal{P}$ is given by equations (12) and (13). The Markov matrix $\mathcal{P}$ is irreducible by Assumption 4. Moreover, by Assumption 2, we have that for all $i \in \mathcal{S}$ and $j \notin \mathcal{S}$, $\lim_{l\to\infty} P\{Z_l^{(i\to j)} > 0\} < 1$. This implies that $\lim_{l\to\infty} P\{Z_l^{(i\to j)} \leq 0\} > 0$ for all $i \in \mathcal{S}$ and $j \notin \mathcal{S}$. By equation (13), this shows that $\mathcal{P}(i,i) > 0$ for all $i \in \mathcal{S}$. (We can assume that $\mathcal{S} \neq \mathcal{K}$ since Theorem 3.2 is trivial otherwise.) This proves that the Markov matrix $\mathcal{P}$ is aperiodic. By Lemma 3.1,

$$\frac{N_M(n)}{M} = \frac{1}{M} \sum_{m=0}^{M} I_{\{k_m = n\}} \to \pi_n \text{ a.s. as } M \to \infty,$$

for all $n \in \mathcal{K}$, where $\pi_1, \ldots, \pi_K$ is the stationary distribution corresponding to the Markov matrix $\mathcal{P}$. The remainder of this proof resembles the proof of Theorem 2.1. □

3.1. Discrete optimization in transient simulation. For all $n \in \mathcal{K}$, suppose that $\{X_i(n)\}$ is a stochastic process whose distribution depends on the parameter n. Suppose τ_n is a stopping time with respect to $\{X_i(n)\}$ for all $n \in \mathcal{K}$ and that we are concerned with solving the optimization problem

$$\min_{n\in\mathcal{K}} f(n) = E\{g(n, X_0(n), \ldots, X_{\tau_n}(n))\}. \tag{14}$$

Since $f(n)$ depends only on $X_0(n), \ldots, X_{\tau_n}(n)$ for all $n \in \mathcal{K}$, we have that when sample paths of the stochastic process $\{X_i(n)\}$ can be generated through simulation,

(14) is a discrete optimization problem in transient simulation. Suppose that $f(n)$ is finite for all $n \in \mathcal{K}$ and that it is possible to generate independent observations $X_0^k(n), \ldots, X_{\tau_n^k}^k(n)$, $k = 1, 2, \ldots$, of $X_0(n), \ldots, X_{\tau_n}(n)$ for all $n \in \mathcal{K}$. By the strong law of large numbers, we have that

$$\frac{1}{l} \sum_{k=1}^{l} g(n, X_0^k(n), \ldots, X_{\tau_n^k}^k(n)) \to f(n) \text{ a.s. as } l \to \infty,$$

for all $n \in \mathcal{K}$. Let $0 < \epsilon < \frac{1}{2}$ and $z \neq 0$. For all $l \in \mathbb{N}$, let $Z_l^{(i \to i)} = 0$ for all $i \in \mathcal{K}$, and let

$$(15)\quad Z_l^{(i \to j)} = \begin{cases} -z & \text{with probability } \epsilon, \\ z & \text{with probability } \epsilon, \\ \frac{1}{l} \sum_{k=1}^{l} \Big[g(i, X_0^k(i), \ldots, X_{\tau_i^k}^k(i)) & \\ \qquad - g(j, X_0^k(j), \ldots, X_{\tau_j^k}^k(j)) \Big] & \text{with probability } 1 - 2\epsilon \end{cases}$$

for all $i, j \in \mathcal{K}$, $i \neq j$. Define

$$\sigma^2(n) = \operatorname{Var}\{g(n, X_0(n), \ldots, X_{\tau_n}(n))\}$$

for all $n \in \mathcal{K}$. We can now prove the following theorem.

THEOREM 3.3 (application of Algorithm 3.1 in transient simulation). *For all $n \in \mathcal{K}$, suppose that $\{X_i(n)\}$ is a stochastic process whose distribution depends on the parameter n and τ_n is a stopping time with respect to $\{X_i(n)\}$. Moreover, suppose that for all $n \in \mathcal{K}$, $f(n) < \infty$, and that $X_0^k(n), \ldots, X_{\tau_n^k}^k(n)$, $k = 1, 2, \ldots$, are independent observations of $X_0(n), \ldots, X_{\tau_n}(n)$. Finally, suppose that if $i, j \in \mathcal{K}$, $i \neq j$, are such that $f(i) = f(j)$, then $0 < \sigma^2(i), \sigma^2(j) < \infty$. Then the sequences of random variables $\{Z_l^{(i \to j)}\}$ defined by equation (15) satisfy Assumptions 2 and 4.*

Remarks. 1. Note that when the random variables $Z_l^{(i \to j)}$ are given by equation (15), Algorithm 3.1 becomes "random" in the sense that the decision whether the algorithm stays at the current point k_m or moves to the alternative point J_m does not depend only on measurements taken at the points k_m and J_m. For any $0 < \epsilon < \frac{1}{2}$, the choice of $z \neq 0$ does not affect the performance of the algorithm. However, the choice of the parameter $\epsilon \in (0, \frac{1}{2})$ will clearly affect the convergence behavior of Algorithm 3.1. As larger values of the parameter ϵ make the algorithm more "random" in that measurements taken at the points k_m and J_m are used less often to decide which one of the two points k_m and J_m is more promising (and thereby becomes the next current point k_{m+1}), we expect that values of ϵ that are close to zero will result in better behavior than values of ϵ that are close to $\frac{1}{2}$. However, additional (theoretical and numerical) work on how the rate of convergence of Algorithm 3.1 depends on the parameter ϵ is needed.

2. To preserve the Markov property, the observation of the random variable $Z_{l_m}^{(k_m \to J_m)}$ generated in iteration m of Algorithm 3.1 must be independent of the past.

Proof. Define

$$\Sigma_l^{(i \to j)} = \frac{1}{l} \sum_{k=1}^{l} \Big[g(i, X_0^k(i), \ldots, X_{\tau_i^k}^k(i)) - g(j, X_0^k(j), \ldots, X_{\tau_j^k}^k(j)) \Big]$$

for all $i, j \in \mathcal{K}$, $i \neq j$, and $l \in \mathbb{N}$. By the strong law of large numbers and the central limit theorem, it is clear that

$$\lim_{l\to\infty} P\{\Sigma_l^{(i\to j)} > 0\} = \begin{cases} 1 & \text{if } f(i) > f(j), \\ 0 & \text{if } f(i) < f(j), \\ \frac{1}{2} & \text{if } f(i) = f(j). \end{cases} \tag{16}$$

We have

$$P\{Z_l^{(i\to j)} > 0\} = \epsilon + (1-2\epsilon)P\{\Sigma_l^{(i\to j)} > 0\},$$
$$P\{Z_l^{(i\to j)} \leq 0\} = \epsilon + (1-2\epsilon)P\{\Sigma_l^{(i\to j)} \leq 0\}$$

for all $i, j \in \mathcal{K}$, $i \neq j$. By equation (16), this shows that the limit $\lim_{l\to\infty} P\{Z_l^{(i\to j)} > 0\}$ exists for all $i, j \in \mathcal{K}$. It also shows that Assumption 4 is satisfied. It remains to show that equations (9), (10), and (11) are satisfied. We have

$$P\{Z_l^{(j\to i)} > 0\} - P\{Z_l^{(i\to j)} > 0\} = (1-2\epsilon)[P\{\Sigma_l^{(j\to i)} > 0\} - P\{\Sigma_l^{(i\to j)} > 0\}], \tag{17}$$

$$P\{Z_l^{(n\to i)} > 0\} - P\{Z_l^{(n\to j)} > 0\} = (1-2\epsilon)[P\{\Sigma_l^{(n\to i)} > 0\} - P\{\Sigma_l^{(n\to j)} > 0\}], \tag{18}$$

$$\begin{aligned} P\{Z_l^{(i\to n)} \leq 0\} - P\{Z_l^{(j\to n)} \leq 0\} &= (1-2\epsilon)[P\{\Sigma_l^{(i\to n)} \leq 0\} - P\{\Sigma_l^{(j\to n)} \leq 0\}] \\ &= (1-2\epsilon)[P\{\Sigma_l^{(j\to n)} > 0\} - P\{\Sigma_l^{(i\to n)} > 0\}]. \end{aligned} \tag{19}$$

Now suppose $i \in \mathcal{S}$, $j \notin \mathcal{S}$, and $n \in \mathcal{K}$, $n \neq i, j$. Then $f(i) < f(j)$, so by equations (16) and (17), we have

$$\lim_{l\to\infty} P\{Z_l^{(j\to i)} > 0\} - \lim_{l\to\infty} P\{Z_l^{(i\to j)} > 0\} = 1 - 2\epsilon > 0.$$

This shows that equation (9) holds. It remains to show that equations (10) and (11) hold. There are four cases.

Case 1: $f(i) = f(n) < f(j)$. Then

$$\lim_{l\to\infty} P\{Z_l^{(n\to i)} > 0\} - \lim_{l\to\infty} P\{Z_l^{(n\to j)} > 0\} = \tfrac{1}{2} - \epsilon > 0,$$
$$\lim_{l\to\infty} P\{Z_l^{(i\to n)} \leq 0\} - \lim_{l\to\infty} P\{Z_l^{(j\to n)} \leq 0\} = \tfrac{1}{2} - \epsilon > 0$$

by equations (16), (18), and (19), so equations (10) and (11) hold.

Case 2: $f(i) < f(n) < f(j)$. Then

$$\lim_{l\to\infty} P\{Z_l^{(n\to i)} > 0\} - \lim_{l\to\infty} P\{Z_l^{(n\to j)} > 0\} = 1 - 2\epsilon > 0,$$
$$\lim_{l\to\infty} P\{Z_l^{(i\to n)} \leq 0\} - \lim_{l\to\infty} P\{Z_l^{(j\to n)} \leq 0\} = 1 - 2\epsilon > 0$$

by equations (16), (18), and (19), so equations (10) and (11) hold.

Case 3: $f(i) < f(j) = f(n)$. Then

$$\lim_{l\to\infty} P\{Z_l^{(n\to i)} > 0\} - \lim_{l\to\infty} P\{Z_l^{(n\to j)} > 0\} = \tfrac{1}{2} - \epsilon > 0,$$
$$\lim_{l\to\infty} P\{Z_l^{(i\to n)} \leq 0\} - \lim_{l\to\infty} P\{Z_l^{(j\to n)} \leq 0\} = \tfrac{1}{2} - \epsilon > 0$$

by equations (16), (18), and (19), so equations (10) and (11) hold.

Case 4: $f(i) < f(j) < f(n)$. Then

$$\lim_{l\to\infty} P\{Z_l^{(n\to i)} > 0\} - \lim_{l\to\infty} P\{Z_l^{(n\to j)} > 0\} = 0,$$
$$\lim_{l\to\infty} P\{Z_l^{(i\to n)} \le 0\} - \lim_{l\to\infty} P\{Z_l^{(j\to n)} \le 0\} = 0$$

by equations (16), (18), and (19), so equations (10) and (11) hold.

This shows that Assumption 2 is satisfied in all four cases, and the proof is thus complete. □

3.2. Discrete optimization in steady-state simulation. For all $n \in \mathcal{K}$, suppose that $\{X_i(n)\}$ is a stochastic process whose distribution depends on the parameter n. Suppose $X_i(n) \Rightarrow X(n)$ as $i \to \infty$ for all $n \in \mathcal{K}$ and that we are concerned with solving the optimization problem

$$\min_{n\in\mathcal{K}} f(n) = E\{g(n, X(n))\}. \tag{20}$$

Since $f(n)$ depends on the steady-state limit $X(n)$ of $\{X_i(n)\}$ for all $n \in \mathcal{K}$, we have that when sample paths of the stochastic process $\{X_i(n)\}$ can be generated through simulation, (20) is a discrete optimization problem in steady-state simulation. Let $0 < \epsilon < \frac{1}{2}$ and $z \neq 0$. For all $l \in \mathbb{N}$, let $Z_l^{(i\to i)} = 0$ for all $i \in \mathcal{K}$, and let

$$Z_l^{(i\to j)} = \begin{cases} -z & \text{with probability } \epsilon, \\ z & \text{with probability } \epsilon, \\ \frac{1}{l}\sum_{k=1}^{l} [g(i, X_k(i)) - g(j, X_k(j))] & \text{with probability } 1-2\epsilon \end{cases} \tag{21}$$

for all $i, j \in \mathcal{K}$, $i \neq j$. The proof of the following theorem resembles the proof of Theorem 3.3.

THEOREM 3.4 (application of Algorithm 3.1 in steady-state simulation). *For all $n \in \mathcal{K}$, suppose that $\{X_i(n)\}$ is a stochastic process whose distribution depends on the parameter n and that $X_i(n) \Rightarrow X(n)$. Moreover, suppose that for all $n \in \mathcal{K}$,*

$$\frac{1}{l}\sum_{k=1}^{l} g(n, X_k(n)) \to f(n) \text{ a.s. as } l \to \infty, \tag{22}$$

and if $i, j \in \mathcal{K}$, $i \neq j$, are such that $f(i) = f(j)$, then there exists $0 < \sigma^2(i,j) < \infty$ such that

$$\frac{1}{\sqrt{l}}\sum_{k=1}^{l} [g(i, X_k(i)) - g(j, X_k(j))] \Rightarrow N(0, \sigma^2(i,j)) \text{ as } l \to \infty. \tag{23}$$

Then the sequences of random variables $\{Z_l^{(i\to j)}\}$ defined by equation (21) satisfy Assumptions 2 and 4.

Remarks. 1. For all $l \in \mathbb{N}$, define $\Sigma_l^{(i\to j)} = \frac{1}{l}\sum_{k=1}^{l}[g(i, X_k(i)) - g(j, X_k(j))]$ for all $i, j \in \mathcal{K}$, $i \neq j$. The assumptions—that equation (22) holds for all $n \in \mathcal{K}$ and that if $i, j \in \mathcal{K}$, $i \neq j$, are such that $f(i) = f(j)$, then there exists $0 < \sigma^2(i,j) < \infty$ such that equation (23) holds—are needed only to prove that equation (16) in the proof of Theorem 3.3 holds.

2. For results that can be used to show that for all $i, j, n \in \mathcal{K}$, $i \neq j$, there exists $0 < \sigma^2(i,j) < \infty$, such that equations (22) and (23) in Theorem 3.4 hold, the reader

is referred to Asmussen [4, §V.3], and to Iosifescu and Theodorescu [15, §§1.1.3.2 and 1.1.4.2].

3. To preserve the Markov property, the observation of the random variable $Z_{l_m}^{(k_m \to J_m)}$ generated in iteration m of Algorithm 3.1 must be independent of the past.

When the stochastic process $\{X_i(n)\}$ is regenerative, it is possible to use the regenerative method (see, for instance, Crane and Lemoine [5] and Iglehart and Shedler [14]) to obtain sequences of random variables $\{Z_l^{(i \to j)}\}$ that satisfy Assumptions 2 and 4. Suppose that for all $n \in \mathcal{K}$, $\{X_i(n)\}$ is regenerative with regeneration times $\tau_0(n), \tau_1(n), \tau_2(n), \ldots$, and let $Y_k(n) = \sum_{i=\tau_{k-1}(n)}^{\tau_k(n)-1} g(n, X_i(n))$ for all $k \in \mathbb{N}$. Let $0 < \epsilon < \frac{1}{2}$ and $z \neq 0$. For all $l \in \mathbb{N}$, let $Z_l^{(i \to i)} = 0$ for all $i \in \mathcal{K}$, and let

$$Z_l^{(i \to j)} = \begin{cases} -z & \text{with probability } \epsilon, \\ z & \text{with probability } \epsilon, \\ \dfrac{\sum_{k=1}^{l} Y_k(i)}{\sum_{k=1}^{l} \tau_k(i)} - \dfrac{\sum_{k=1}^{l} Y_k(j)}{\sum_{k=1}^{l} \tau_k(j)} & \text{with probability } 1 - 2\epsilon \end{cases} \tag{24}$$

for all $i, j \in \mathcal{K}$, $i \neq j$. The proof of the following theorem resembles the proof of Theorem 3.3.

THEOREM 3.5 (application of Algorithm 3.1 in regenerative simulation). *For all $n \in \mathcal{K}$, suppose that $\{X_i(n)\}$ is a regenerative stochastic process whose distribution depends on the parameter n and that $X_i(n) \Rightarrow X(n)$. Moreover, suppose that for all $n \in \mathcal{K}$,*

$$\frac{\sum_{k=1}^{l} Y_k(n)}{\sum_{k=1}^{l} \tau_k(n)} \to f(n) \text{ a.s. as } l \to \infty, \tag{25}$$

and if $i, j \in \mathcal{K}$, $i \neq j$, are such that $f(i) = f(j)$, then there exists $0 < \sigma^2(i,j) < \infty$ such that

$$\sqrt{l} \left\{ \frac{\sum_{k=1}^{l} Y_k(i)}{\sum_{k=1}^{l} \tau_k(i)} - \frac{\sum_{k=1}^{l} Y_k(j)}{\sum_{k=1}^{l} \tau_k(j)} \right\} \Rightarrow N(0, \sigma^2(i,j)) \text{ as } l \to \infty. \tag{26}$$

Then the sequences of random variables $\{Z_l^{(i \to j)}\}$ defined by equation (24) satisfy Assumptions 2 and 4.

Remarks. 1. For all $l \in \mathbb{N}$, define $\Sigma_l^{(i \to j)} = \sum_{k=1}^{l} Y_k(i) / \sum_{l=1}^{l} \tau_k(i) - \sum_{k=1}^{l} Y_k(j) / \sum_{l=1}^{l} \tau_k(j)$ for all $i, j \in \mathcal{K}$, $i \neq j$. The assumptions—that equation (25) holds for all $n \in \mathcal{K}$ and that if $i, j \in \mathcal{K}$, $i \neq j$, are such that $f(i) = f(j)$, then there exists $0 < \sigma^2(i,j) < \infty$ such that equation (26) holds—are needed only to prove that equation (16) in the proof of Theorem 3.3 holds.

2. For results that can be used to show that for all $i, j, n \in \mathcal{K}$, $i \neq j$, there exists $0 < \sigma^2(i,j) < \infty$, such that equations (25) and (26) in Theorem 3.5 hold, the reader is referred to Crane and Lemoine [5] and Iglehart and Shedler [14].

3. To preserve the Markov property, the observation of the random variable $Z_{l_m}^{(k_m \to J_m)}$ generated in iteration m of Algorithm 3.1 must be independent of the past.

4. Applications. The main difficulty in applying Algorithm 2.1 to solve discrete stochastic optimization problems lies in finding random variables $Y^{(i \to j)}$, for all $i, j \in$

$\mathcal{K}$, $i \neq j$, that satisfy Assumption 1. In §3, we showed that for simulation optimization problems, this difficulty can be avoided by using a generalized variant of Algorithm 2.1 (Algorithm 3.1). However, when Algorithm 2.1 is applicable, it requires only one replication per iteration, whereas Algorithm 3.1 requires l_m replications in iteration m, where $l_m \to \infty$ as $m \to \infty$ (see Assumption 3). It is therefore reasonable to expect that when Algorithm 2.1 is applicable, it will probably perform better than Algorithm 3.1. In the remark following the statement of Assumption 1, we showed how random variables $Y^{(i \to j)}$ satisfying Assumption 1 can be obtained for a class of problems. We now show how random variables $Y^{(i \to j)}$ that satisfy Assumption 1 can be found when it is of interest to optimize probabilities.

Consider the following discrete optimization problem:

$$\min_{n \in \mathcal{K}} f(n) = bE\{I_n\}, \tag{27}$$

where $\{I_n\}$ is a sequence of indicator random variables (so that $E\{I_n\}$ is a probability), $\mathcal{K} = \{1, \ldots, K\}$, and $b \neq 0$. For all $i, j \in \mathcal{K}$, let

$$Y^{(i \to j)} = b(I_i - I_j). \tag{28}$$

Suppose $i \in \mathcal{S}$, $j \notin \mathcal{S}$, and $n \in \mathcal{K}$, $n \neq i, j$, and that it is possible to generate independent observations of I_i, I_j, and I_n. For all $n \in \mathcal{K}$, let $x_n = E\{I_n\}$. When $b > 0$, we have $x_j > x_i$ and $0 \leq x_n \leq 1$. Therefore,

$$\begin{aligned} P\{Y^{(j \to i)} > 0\} - P\{Y^{(i \to j)} > 0\} &= P\{I_j > I_i\} - P\{I_i > I_j\} \\ &= x_j(1 - x_i) - x_i(1 - x_j) \\ &= x_j - x_i > 0. \end{aligned}$$

This shows that equation (2) is satisfied. Moreover,

$$\begin{aligned} P\{Y^{(n \to i)} > 0\} - P\{Y^{(n \to j)} > 0\} &= P\{I_n > I_i\} - P\{I_n > I_j\} \\ &= x_n(1 - x_i) - x_n(1 - x_j) \\ &= x_n(x_j - x_i) \geq 0. \end{aligned}$$

This shows that equation (3) is satisfied. Finally,

$$\begin{aligned} P\{Y^{(i \to n)} \leq 0\} - P\{Y^{(j \to n)} \leq 0\} &= P\{I_i \leq I_n\} - P\{I_j \leq I_n\} \\ &= 1 - x_i(1 - x_n) - 1 + x_j(1 - x_n) \\ &= (1 - x_n)(x_j - x_i) \geq 0. \end{aligned}$$

This shows that equation (4) is satisfied. We have shown that Assumption 1 is satisfied when $b > 0$. The case when $b < 0$ is similar.

We now present a numerical example illustrating the performance of Algorithm 2.1 when applied to solve an optimization problem of the form (27). Additional (theoretical and numerical) work on how the speed of convergence of Algorithms 2.1 and 3.1 depends on the choice and properties of the random variables $Y^{(i \to j)}$ and $Z_l^{(i \to j)}$, $i, j \in \mathcal{K}$, would be helpful to better understand the behavior of these algorithms.

Example 4.1. Suppose that the demand for a certain product has a Poisson distribution with parameter λ. Consider the problem of determining the order size n that maximizes the probability that the demand D equals n, subject to the constraint

that at most N units can be ordered. This is an optimization problem of the form (27), with $\mathcal{K} = \{0, \ldots, N\}$, $b = -1$, and $I_n = I_{\{D=n\}}$ for all $n \in \mathcal{K}$.

The remainder of this section will be devoted to the application of Algorithm 2.1 to solve the discrete optimization problem just described. Suppose the random variables $Y^{(i \to j)}$ are given by equation (28), for all $i, j \in \mathcal{K}$. Then $Y^{(i \to i)} = 0$ for all $i \in \mathcal{K}$, and

$$Y^{(i \to j)} = I_{\{D_1 = j\}} - I_{\{D_2 = i\}} \tag{29}$$

for all $i, j \in \mathcal{K}$, $i \neq j$, where D_1 and D_2 are independent random variables having the Poisson distribution with parameter λ. It is clear that the objective function f of the optimization problem (27) is given by

$$f(n) = \frac{\lambda^n}{n!} e^{-\lambda} \tag{30}$$

for all n, so the solution to the optimization problem (27) can be determined analytically. Because we can compare the analytical solution with the solution obtained by using Algorithm 2.1, this example will serve to illustrate the behavior of Algorithm 2.1.

Suppose $\lambda \in \{1, 10\}$ and $N \in \{10, 100\}$. When $\lambda = 1$, the set of global solutions is $\mathcal{S} = \{0, 1\}$, and when $\lambda = 10$, $\mathcal{S} = \{9, 10\}$. To test the performance of Algorithm 2.1 on these four problems, we ran it 100 times with the starting point k_0 having the uniform distribution on the set $\{0, 1, \ldots, N\}$. The performance of the algorithm is shown in Table 1 (the number of replications for which $k_m^* \in \mathcal{S}$ is displayed for a number of different values for m).

TABLE 1
Performance of Algorithm 2.1 when $Y^{(i \to j)}$ is given by equation (29), for all $i, j \in \mathcal{K}$.

Iteration	Number of convergent paths			
(m)	$\lambda = 1$, $N = 10$	$\lambda = 1$, $N = 100$	$\lambda = 10$, $N = 10$	$\lambda = 10$, $N = 100$
10	58	5	16	4
100	100	49	48	12
1,000	100	99	64	24
5,000	100	100	82	43
10,000	100	100	86	48
20,000	100	100	88	57
50,000	100	100	96	66
100,000	100	100	99	80
200,000	100	100	99	82
300,000	100	100	100	90
400,000	100	100	100	99
500,000	100	100	100	97

In all four cases, Algorithm 2.1 moves from the starting point k_0 toward the set $\mathcal{S}$, but the speed of convergence depends on the size of the feasible region $(N+1)$ and on the parameter λ. The speed of convergence decreases as the size of the feasible set increases (so as N increases). The speed of convergence also decreases when the parameter λ increases. This is because if $i \in \mathcal{S}$ and $j \notin \mathcal{S}$, then $|f(i) - f(j)|$ (where f is the objective function given in equation (30)) is larger when $\lambda = 1$ than when $\lambda = 10$ (when $\lambda = 1$, we have $0.184 \leq |f(i) - f(j)| \leq 0.368$, and when $\lambda = 10$, we have $0.011 \leq |f(i) - f(j)| \leq 0.125$). The effect of increasing the parameter λ is more pronounced than the effect of increasing the size of the feasible region.

The choice of the random variables $Y^{(i\to j)}$, $i,j \in \mathcal{K}$, is not necessarily unique, and the performance of Algorithm 2.1 depends on how these random variables are selected. In our case, we can obtain another choice for $Y^{(i\to j)}$ by using common random numbers. Define $Y^{(i\to i)} = 0$ for all $i \in \mathcal{K}$, and

$$Y^{(i\to j)} = I_{\{D=j\}} - I_{\{D=i\}} \tag{31}$$

for all $i,j \in \mathcal{K}$, $i \neq j$, where D has the Poisson distribution with parameter λ. Suppose $i \in \mathcal{S}$, $j \notin \mathcal{S}$, and $n \in \mathcal{K}$, $n \neq i,j$, and for all $n \in \mathcal{K}$, let $x_n = E\{I_{\{D=n\}}\} = P\{D=n\}$. Then $x_i > x_j$, and

$$\begin{aligned} P\{Y^{(j\to i)} > 0\} - P\{Y^{(i\to j)} > 0\} &= P\{D=i\} - P\{D=j\} \\ &= x_i - x_j > 0. \end{aligned}$$

This shows that equation (2) is satisfied. Moreover,

$$\begin{aligned} P\{Y^{(n\to i)} > 0\} - P\{Y^{(n\to j)} > 0\} &= P\{D=i\} - P\{D=j\} \\ &= x_i - x_j > 0. \end{aligned}$$

This shows that equation (3) is satisfied. Finally,

$$\begin{aligned} P\{Y^{(i\to n)} \leq 0\} - P\{Y^{(j\to n)} \leq 0\} &= P\{D \neq n\} - P\{D \neq n\} \\ &= 0. \end{aligned}$$

This shows that equation (4) is satisfied. We have shown that Assumption 1 is satisfied. The performance of Algorithm 2.1 with this choice of $Y^{(i\to j)}$ is shown in Table 2. Again, $\lambda \in \{1,10\}$, $N \in \{10,100\}$, and k_0 is uniformly distributed on the set $\{0,1,\ldots,N\}$. The tables show the performance of the algorithm over 100 runs.

TABLE 2
Performance of Algorithm 2.1 *when* $Y^{(i\to j)}$ *is given by equation* (31), *for all* $i,j \in \mathcal{K}$.

Iteration (m)	Number of convergent paths			
	$\lambda = 1$, $N = 10$	$\lambda = 1$, $N = 100$	$\lambda = 10$, $N = 10$	$\lambda = 10$, $N = 100$
10	55	4	23	3
100	96	41	38	12
1,000	100	99	66	28
5,000	100	100	79	51
10,000	100	100	81	61
20,000	100	100	89	59
50,000	100	100	95	68
100,000	100	100	100	78
200,000	100	100	100	82
300,000	100	100	100	82
400,000	100	100	100	90
500,000	100	100	100	93

When $Y^{(i\to j)}$ is given by equation (31), Algorithm 2.1 moves from the starting point k_0 toward the set $\mathcal{S}$, but the speed of convergence depends on the size of the feasible region $(N+1)$ and on the parameter λ. The convergence is fastest when $\lambda = 1$ and $N = 10$, and it is slowest when $\lambda = 10$ and $N = 100$. The speed of convergence of Algorithm 2.1 when $Y^{(i\to j)}$ is given by equation (31) is very similar to the speed of convergence when $Y^{(i\to j)}$ is given by equation (29). (Algorithm 2.1 seems to converge slightly faster to the set $\mathcal{S}$ when $Y^{(i\to j)}$ is given by equation (29) than when $Y^{(i\to j)}$ is

given by equation (31).) However, when $Y^{(i\to j)}$ is given by equation (31), then $Y^{(i\to j)}$ can be generated from a single replication of the demand D, whereas when $Y^{(i\to j)}$ is given by equation (29), then two independent replications D_1 and D_2 of the demand D are needed. This shows that when $Y^{(i\to j)}$ is given by equation (31), Algorithm 2.1 requires less computation per iteration than when $Y^{(i\to j)}$ is given by equation (29). This example therefore shows that the performance of Algorithm 2.1 depends on the choice of the random variables $Y^{(i\to j)}$, $i, j \in \mathcal{K}$, both in terms of the computational effort per iteration and in terms of the number of iterations it takes to converge to $\mathcal{S}$, the set of global solutions to the underlying optimization problem.

5. Conclusion. We have presented two variants of a new iterative method for discrete stochastic optimization. Both variants are guaranteed to converge almost surely to a global solution of the underlying optimization problem as the number of iterations goes to infinity. We have shown that one variant of the proposed method can be applied to solve discrete optimization problems in both transient and steady-state simulation and that the other variant can be applied to solve a particular class of discrete stochastic optimization problems. Some numerical results have also been presented.

The method proposed in this paper is in many ways similar to the method proposed earlier by Andradóttir [2, 3]. However, the earlier method uses *local search* to update the sequence $\{k_m\}$ generated by the algorithm (so for all m, k_{m+1} is one of two neighbors of k_m), whereas the current method uses *global search* to update the sequence $\{k_m\}$. Thus we expect the method proposed in this paper to perform better than the earlier method in situations where a natural neighborhood structure does not exist and in situations where the objective function has many local minimizers. Both the method proposed in this paper and the method proposed by Andradóttir [2, 3] have the following features: they converge *almost surely*, they can be applied to solve discrete stochastic optimization problems involving a large number of alternatives, they require no boundedness assumptions in the sense that it is not necessary to assume that the objective function is bounded or that the noise in the evaluations of the objective function is bounded in any sense, the basic variant of these methods requires only two function evaluations per iteration, the generalized variant of these methods can be applied to solve optimization problems in both transient and steady-state simulation, and finally, these methods spend most of the computational effort close to the optimal solution of the underlying optimization problem. (In fact, the estimate of the solution is always the parameter value that these algorithms have visited most often.)

REFERENCES

[1] R. AGRAWAL, *Sample mean based index policies with $O(\log(n))$ regret for the multi-armed bandit problem*, Adv. in Appl. Probab., 27 (1995), pp. 1054–1078.

[2] S. ANDRADÓTTIR, *Discrete optimization in simulation: A method and applications*, in Proc. 1992 Winter Simulation Conf., Arlington, VA, 1992, pp. 483–486.

[3] ———, *A method for discrete stochastic optimization*, Management Sci., 1995, to appear.

[4] S. ASMUSSEN, *Applied Probability and Queues*, Wiley, Chichester, UK, 1987.

[5] M.A. CRANE AND A. J. LEMOINE, *An Introduction to the Regenerative Method for Simulation Analysis*, Springer-Verlag, Berlin, 1977.

[6] L. DEVROYE, *On the convergence of statistical search*, IEEE Trans. Systems Man Cybernet., 6 (1976), pp. 46–56.

[7] ———, *Probabilistic search as a strategy selection procedure*, IEEE Trans. Systems Man Cybernet., 6 (1976), pp. 315–321.

[8] ———, *A class of optimal performance directed probabilistic automata*, IEEE Trans. Systems Man Cybernet., 6 (1976), pp. 777–783.

[9] J. D. GIBBONS, I. OLKIN, AND M. SOBEL, *Selecting and Ordering Populations: A New Statistical Methodology*, Wiley, New York, 1977.

[10] D. GOLDSMAN, B. L. NELSON, AND B. SCHMEISER, *Methods for selecting the best system*, in Proc. 1991 Winter Simulation Conf., Phoenix, AZ, 1991, pp. 177–186.

[11] W. B. GONG, Y. C. HO, AND W. ZHAI, *A Stochastic Comparison Algorithm for Discrete Optimization with Estimation*, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, 1993, preprint.

[12] S. S. GUPTA AND S. PANCHAPAKESAN, *Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations*, Wiley, New York, 1979.

[13] Y. HOCHBERG AND A. C. TAMHANE, *Multiple Comparison Procedures*, Wiley, New York, 1987.

[14] D. L. IGLEHART AND G. S. SHEDLER, *Regenerative Simulation of Response Times in Networks of Queues*, Springer-Verlag, Berlin, 1980.

[15] M. IOSIFESCU AND R. THEODORESCU, *Random Processes and Learning*, Springer-Verlag, New York, 1969.

[16] S. LAKSHMIVARAHAN AND M. A. L. THATHACHAR, *Absolute expediency of Q- and S-model learning algorithms*, IEEE Trans. Systems Man Cybernet., 6 (1976), pp. 222–226.

[17] T. L. LAI AND H. ROBBINS, *Asymptotically efficient adaptive allocation rules*, Adv. in Appl. Math., 6 (1985), pp. 4–22.

[18] N. METROPOLIS, A. W. ROSENBLUTH, M. N. ROSENBLUTH, A. H. TELLER, AND E. TELLER, *Equation of state calculations by fast computing machines*, J. Chem. Phys., 21 (1953), pp. 1087–1092.

[19] H. ROBBINS, *Some aspects of the sequential design of experiments*, Bull. Amer. Math. Soc., 58 (1952), pp. 527–535.

[20] I. J. SHAPIRO AND K. S. NARENDRA, *Use of stochastic automata for parameter self-optimization with multimodal performance criteria*, IEEE Trans. Systems Sci. Cybernet., 5 (1969), pp. 352–360.

[21] E. TSETLIN, *On the behavior of finite automata in random media*, Automat. Remote Control, 22 (1961), pp. 1210–1219.

[22] R. VISWANATHAN AND K. S. NARENDRA, *A note on the linear reinforcement scheme for variable-structure stochastic automata*, IEEE Trans. Systems Man Cybernet., 2 (1972), pp. 292–294.

[23] S. J. YAKOWITZ AND E. LUGOSI, *Random search in the presence of noise, with applications to machine learning*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 702–712.

[24] S. J. YAKOWITZ AND W. LOWE, *Nonparametric bandit methods*, Ann. Oper. Res., 28 (1991), pp. 297–312.

[25] D. YAN AND H. MUKAI, *Stochastic discrete optimization*, SIAM J. Control Optim., 30 (1992), pp. 594–612.

SIAM J. OPTIMIZATION
Vol. 6, No. 2, pp. 531–547, May 1996
© 1996 Society for Industrial and Applied Mathematics
014

LIPSCHITZ STABILITY FOR STOCHASTIC PROGRAMS WITH COMPLETE RECOURSE*

WERNER RÖMISCH† AND RÜDIGER SCHULTZ‡

Abstract. This paper investigates the stability of optimal solution sets to stochastic programs with complete recourse, where the underlying probability measure is understood as a parameter varying in some space of probability measures. In [*Math. Programming*, 67 (1994), pp. 99–108] Shapiro proved Lipschitz upper semicontinuity of the solution set mapping. Inspired by this result, we introduce a subgradient distance for probability distributions and establish the persistence of optimal solutions. For a subclass of recourse models we show that the solution set mapping is (Hausdorff) Lipschitz continuous with respect to the subgradient distance. Moreover, the subgradient distance is estimated above by the Kolmogorov–Smirnov distance of certain distribution functions related to the recourse model. The Lipschitz continuity result is illustrated by verifiable sufficient conditions for stochastic programs to belong to the mentioned subclass and by examples showing its validity and limitations. Finally, the Lipschitz continuity result is used to derive some new results on the asymptotic behavior of optimal solutions when the probability measure underlying the recourse model is estimated via empirical measures (law of iterated logarithm, large deviation estimate, estimate for asymptotic distribution).

Key words. stochastic programs with recourse, Lipschitz stability, empirical distributions, asymptotic analysis

AMS subject classifications. 90C15, 90C31

1. Introduction.

We study quantitative stability and asymptotic properties (of estimates via empirical measures) of optimal solutions to stochastic programs with complete recourse. The latter are given by

$$P(\mu) \qquad \min\{g(x) + Q_\mu(Ax) : x \in C\}, \tag{1.1}$$

where

$$Q_\mu(\zeta) = \int_{\mathbb{R}^s} Q(z - \zeta)\mu(\mathrm{d}z) \tag{1.2}$$

and

$$Q(t) = \min\{q^T v : Wv = t,\ v \geq 0\}. \tag{1.3}$$

For the data we assume that $g : \mathbb{R}^m \to \mathbb{R}$ is a convex function, $C \subset \mathbb{R}^m$ is a nonempty closed convex set, $q \in \mathbb{R}^{\bar{m}}$, and A, W are matrices of proper dimensions. As indicated in (1.1), the integrating probability measure μ is understood as a parameter which we assume will vary in $\mathcal{M}_1(\mathbb{R}^s)$—the space of all Borel probability measures on $\mathbb{R}^s$ with finite first moment, i.e., $\int_{\mathbb{R}^s} \|z\|\mu(\mathrm{d}z) < +\infty$ for all $\mu \in \mathcal{M}_1(\mathbb{R}^s)$. Further assumptions that ensure (1.1)–(1.3) to be well defined will be given in §2.

It is well known that (1.1)–(1.3) models a two-stage decision process under uncertainty with first-stage decision x, random entry z, and second-stage (or recourse)

* Received by the editors April 2, 1993; accepted for publication (in revised form) October 10, 1994. This research was supported by the Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung" of the Deutsche Forschungsgemeinschaft.

† Humboldt-Universität zu Berlin, Institut für Mathematik, Unter den Linden 6, 10099 Berlin, Germany (romisch@mathematik.hu-berlin.de).

‡ Konrad-Zuse-Zentrum für Informationstechnik Berlin, Heilbronner Strasse 10, 10711 Berlin, Germany (schultz@zib-berlin.de).

decision y. For a more detailed introduction to this class of models, including a basic analysis of the function Q_μ in (1.2), we refer to [6], [31]. Here we only mention that Q_μ is convex whenever it is well defined.

In the present paper, we concentrate on studying the impact of changes in the underlying probability measure μ on the problem (1.1). To this end, we assign to $\mu \in \mathcal{M}_1(\mathbb{R}^s)$ the (global) optimal value $\varphi(\mu)$ and the set of (global) optimal solutions $\psi(\mu)$. The mappings φ and ψ are common objects of study in the stability analysis of optimization problems. In the context of stochastic programming, the above setup (i.e., understanding the underlying measure as the quantity subjected to perturbations) has two principal origins: the numerical intractability of the integral in (1.2) and the incomplete information on μ that one is faced with in general. In the first case, approximations of a complicated measure μ by simpler ones give rise to a perturbation analysis. In the second case, perturbations come via attempts to construct some "reasonable" measure μ based on the (statistical) information that is available on the random parameter z. For more details on the stability of stochastic programs we refer to [3], [4], [7], [12], [19], [22], [26], [28], [30], [32], [33].

The subsequent analysis is entirely concerned with quantitative continuity properties of the optimal set mapping ψ. As in our earlier work [22], [23], we dispense with the assumption that the solution set of the unperturbed problem is a singleton.

For the model (1.1)–(1.3), uniqueness of optimal solutions is rather exceptional, as is seen by the following example. Let us first mention that the example does fit the setting of our central stability estimate; in particular, the function Q_μ is here strongly convex on a suitable subset (cf. Theorem 2.4).

Example 1.1. In (1.1)–(1.3) let $m = 2$, $s = 1$, $g(x) \equiv 0$, $A = (1, 0)$, $(0, 0)^T \in C$, $\bar{m} = 2$, $q = (1, 1)^T$, $W = (1, -1)$, and μ be the uniform distribution on the closed interval $[-1/2, 1/2]$. Then it is straightforward to see that $\psi(\mu) = \ker A \cap C = \{(0, \xi)^T \in C,\ \xi \in \mathbb{R}\}$.

One observes that Q_μ in (1.2) is always constant on translates of the null space $\ker A$ of A. Hence, uniqueness of optimal solutions is guaranteed only if the constraint set C picks just one element from the relevant level set of Q_μ.

Our investigations have been stimulated by recent results of Shapiro. In [29] the author proves an upper Lipschitz continuity estimate for ψ under the assumption that, for the unperturbed problem $P(\mu)$, the objective function grows at least quadratically for feasible points near the set of optimal solutions. The right-hand side of the estimate essentially consists of the maximal norm of elements arising in the Clarke subdifferential [2] of the function $Q_\nu - Q_\mu$ (cf. (1.2)) at points belonging to a suitable neighbourhood. Here Q_ν corresponds to the perturbed problem $P(\nu)$, $\nu \in \mathcal{M}_1(\mathbb{R}^s)$.

In the present paper we introduce a "subgradient distance" for $\mu, \nu \in \mathcal{M}_1(\mathbb{R}^s)$ based on the above maximal norm (cf. (2.1)). We focus on the stability of models which fit into (1.1)–(1.3) and obey the additional properties that g is convex quadratic, C is a nonempty polyhedron, and Q_μ is strongly convex on a suitable neighbourhood of $A(\psi(\mu))$. Then the Lipschitz upper semicontinuity extends to the Lipschitz continuity of the Hausdorff distance of solution sets (Theorem 2.4). Since the subgradient distance of $\mu, \nu \in \mathcal{M}_1(\mathbb{R}^s)$ can always be estimated above by the Kolmogorov–Smirnov distance of certain distribution functions related to μ, ν and the algebra in (1.3) (Corollary 2.5, Remark 2.6), this leads to a powerful tool for quantitative statements on the stability of optimal solutions. In §3, one such application is worked out in detail—in the presence of empirical measures we derive some

new results on the asymptotic behaviour of solution sets (law of iterated logarithm, large deviation estimate, estimate for asymptotic distribution). In particular, previous results are extended to the case where solution sets are not necessarily singletons. No additional assumptions on the underlying probability measure μ are required for the large deviation result.

Some further propositions and examples supplement and illustrate the main issue of the paper. In Proposition 2.3 the persistence of optimal solutions under perturbations in the "subgradient distance" is addressed. Proposition 2.15 displays some handy conclusions for the special case of "simple recourse." Examples in §2 show that Shapiro's assumptions in [29] do not guarantee the lower semicontinuity of ψ (Example 2.11), that Theorem 2.4 is lost for general convex g and C (Example 2.9), and that the setting of Theorem 2.4 does not guarantee stability of the optimal value (Example 2.12).

Compared to Shapiro's analysis in [29], we confine ourselves to a specific problem class for which we eventually obtain stronger results (Lipschitz continuity instead of Lipschitz upper semicontinuity). Our stability conditions can be verified comparatively easily (strong convexity of Q_μ, to be verified using Proposition 2.13), and in fact they turn out to be sufficient for Shapiro's second-order growth condition to hold. Moreover, the persistence established in Proposition 2.3 further develops the results in [29].

Compared to [22], [23], where the stability analysis is based on the L_1-Wasserstein distance and where Hölder continuity (with exponent 1/2) is obtained, the present paper leads to Lipschitz continuity. To illustrate our improvements over previous results, let us first mention that there exist sequences of measures where the results from both [22], [23] and the present paper lead to the same convergence rates (cf. the discussion after Proposition 2.13). On the other hand, there are important specific modes of perturbation (contaminated distributions, empirical measures) where the Hölder result in [22], [23] yields the rate 1/2, whereas the present approach leads to the rate 1 (Proposition 2.14, §3).

2. Stability. The following basic assumptions are well known to ensure that the function Q_μ in (1.2) is well defined and convex on $\mathbb{R}^m$ (cf. [6], [31]):

(A1) $\{Wy : y \in \mathbb{R}^{\bar{m}}_+\} = \mathbb{R}^s$ (complete recourse),

(A2) $\{u \in \mathbb{R}^s : W^T u \leq q\} \neq \emptyset$ (dual feasibility),

(A3) $\mu \in \mathcal{M}_1(\mathbb{R}^s)$.

For arbitrary $\mu, \nu \in \mathcal{M}_1(\mathbb{R}^s)$ and some fixed, nonempty, closed, convex set $U \subset \mathbb{R}^m$, we define the following "subgradient distance" d of μ and ν:

$$d(\mu, \nu; U) = \sup\{\|z^*\| : z^* \in \partial(Q_\nu - Q_\mu)(Ax),\ x \in U\}. \tag{2.1}$$

Here "∂" denotes Clarke's subdifferential [2]. Since both Q_ν and Q_μ are convex, their difference is locally Lipschitzian and, hence, the Clarke subdifferential in (2.1) is well defined. Provided that U is bounded, one uses simple properties of the Clarke subdifferential to show that $d(\cdot, \cdot; U)$ is a pseudometric on $\mathcal{M}_1(\mathbb{R}^s)$. Note that $d(\mu, \nu; U) = 0$ is possible for $\mu \neq \nu$. Subsequent considerations involving d will use the fact that, in finite dimension, the Clarke subdifferential may be represented as the convex hull of limits of sequences of gradients collected at differentiability points and possibly avoiding arguments in a set of Lebesgue measure zero (Theorem 2.5.1 in [2]). The following lemma provides some more insight into d.

LEMMA 2.1. *Let $h_1, h_2 : \mathbb{R}^s \to \mathbb{R}$ be locally Lipschitzian. Then it holds for arbitrary $\zeta \in \mathbb{R}^s$ that $d_H(\partial h_1(\zeta), \partial h_2(\zeta)) \leq \sup\{\|z^*\| : z^* \in \partial(h_1 - h_2)(\zeta)\}$, where d_H denotes the Hausdorff distance of sets.*

Proof. For $\partial h_i(\zeta)$, $i = 1, 2$, we have the following representation (Theorem 2.5.1 in [2]):

$$\partial h_i(\zeta) = \operatorname{conv} \mathcal{L}_{h_i}(\zeta),$$

where

$$\mathcal{L}_{h_i}(\zeta) = \{z : \text{ there exist } \zeta_n \in \operatorname{Diff}(h_1) \cap \operatorname{Diff}(h_2) \text{ such that } \zeta_n \to \zeta \text{ and } h_i'(\zeta_n) \to z \text{ as } n \to \infty\}.$$

Here $\operatorname{Diff}(h_i)$ denotes the set of differentiability points of h_i. Clearly, $\operatorname{Diff}(h_1) \cap \operatorname{Diff}(h_2) \subset \operatorname{Diff}(h_i)$ and, by Rademacher's theorem, $\mathbb{R}^s \setminus (\operatorname{Diff}(h_1) \cap \operatorname{Diff}(h_2))$ has Lebesgue measure zero.

Assume that

$$d_H(\partial h_1(\zeta), \partial h_2(\zeta)) > \sup\{\|z^*\| : z^* \in \partial(h_1 - h_2)(\zeta)\}$$

for some $\zeta \in \mathbb{R}^s$.

This implies

$$d_H(\mathcal{L}_{h_1}(\zeta), \mathcal{L}_{h_2}(\zeta)) > \sup\{\|z^*\| : z^* \in \partial(h_1 - h_2)(\zeta)\},$$

and, hence, by the definition of the Hausdorff distance, there exists a $z_{1,0}^* \in \mathcal{L}_{h_1}(\zeta)$ (without loss of generality) such that

$$\|z_{1,0}^* - z_2^*\| > \sup\{\|z^*\| : z^* \in \partial(h_1 - h_2)(\zeta)\} \tag{2.2}$$

for all $z_2^* \in \mathcal{L}_{h_2}(\zeta)$.

Since $z_{1,0}^* \in \mathcal{L}_{h_1}(\zeta)$, there exists a sequence of points $\zeta_n \in \operatorname{Diff}(h_1) \cap \operatorname{Diff}(h_2)$ such that $h_1'(\zeta_n) \to z_{1,0}^*$ as $n \to \infty$.

Now consider the sequence $\{h_2'(\zeta_n)\}$. By the local Lipschitz property of h_2 it has an accumulation point $z_{2,0}^*$ that obviously belongs to $\mathcal{L}_{h_2}(\zeta)$. In view of (2.2),

$$\|z_{1,0}^* - z_{2,0}^*\| > \sup\{\|z^*\| : z^* \in \partial(h_1 - h_2)(\zeta)\},$$

but, on the other hand, $z_{1,0}^* - z_{2,0}^* \in \mathcal{L}_{h_1 - h_2}(\zeta) \subset \partial(h_1 - h_2)(\zeta)$, which is an obvious contradiction. □

In our quantitative stability analysis for optimal solutions of perturbed stochastic programs, d will be the distance that measures "how far" away a perturbation $P(\nu)$ is from the original program $P(\mu)$. In the context of stochastic programming Shapiro [29] has also used information contained in the definition of d to derive quantitative stability properties. Kummer [15] has obtained results on the quantitative stability of general convex programs based on the Hausdorff distance of subgradients, which appears in Lemma 2.1. Our considerations start with the following result by Shapiro [29].

THEOREM 2.2. *Suppose* (A1)–(A3), $\psi(\mu) \neq \emptyset$, *and that*

(2.3) *there exists a convex open set U_o containing $\psi(\mu)$ and a constant $\alpha > 0$ such that $g(x) + Q_\mu(Ax) \geq \varphi(\mu) + \alpha \cdot \operatorname{dist}(x, \psi(\mu))^2$ for all $x \in C \cap U_o$, where* dist *denotes the usual point-to-set distance.*

Then the following estimate is valid for all $\nu \in \mathcal{M}_1(\mathbb{R}^s)$:

$$\sup_{x \in \psi(\nu) \cap U_o} \operatorname{dist}(x, \psi(\mu)) \leq \alpha^{-1} \cdot d(\mu, \nu; \operatorname{cl} U_o),$$

where the left-hand side is defined to be zero if $\psi(\nu) \cap U_o = \emptyset$.

Theorem 2.2 asserts the upper Lipschitz continuity of the solution set mapping ψ with respect to the pseudometric d. It does not contain the persistence of optimal solutions, i.e., it is not clear whether the perturbed program $P(\nu)$ has a nonempty set of optimal solutions if $d(\mu, \nu; \operatorname{cl} U_o)$ is sufficiently small. The next proposition answers this question.

PROPOSITION 2.3. *Suppose* (A1)–(A3) *and that* $\psi(\mu)$ *is nonempty and bounded. Let* $U_o \subset \mathbb{R}^m$ *be an open, convex, bounded set containing* $\psi(\mu)$. *Then there exists a constant* $\delta > 0$ *such that*

$$\emptyset \neq \psi(\nu) \subset U_o$$

for all $\nu \in \mathcal{M}_1(\mathbb{R}^s)$ *such that* $d(\mu, \nu; \operatorname{cl} U_o) < \delta$.

Proof. We introduce the following notation:

$$\begin{aligned} G(x, \nu) &:= g(x) + Q_\nu(Ax), \ \nu \in \mathcal{M}_1(\mathbb{R}^s), \\ \psi_d(\mu) &:= \operatorname{argmin}\{G(x, \mu) + d^T x : x \in C\}, \ d \in \mathbb{R}^m, \text{ and} \\ \mathcal{N}_r(M) &:= \{x \in \mathbb{R}^m : \operatorname{dist}(x, M) < r\}, \ M \subset \mathbb{R}^m, \ r > 0. \end{aligned}$$

Select some $r > 0$ such that $\mathcal{N}_r(\psi(\mu)) \subset U_o$. Since $\psi(\mu)$ is bounded and $G(\cdot, \mu)$ is convex, well-known results on the stability of convex programs apply. In particular, Theorem 4.3.3 and Corollary 4.3.3.2 from [1] imply that there exists a constant $\delta' > 0$ such that

$$\emptyset \neq \psi_d(\mu) \subset \mathcal{N}_r(\psi(\mu)) \quad \text{for all } d \in \mathbb{R}^m \text{ with } \|d\| < \delta'. \tag{2.4}$$

To apply results on the stability of certain generalized equations [15], we introduce the set-valued mappings $\Gamma_\nu : \operatorname{cl} U_o \to \mathbb{R}^m$, $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, given by $\Gamma_\nu(x) = \partial_x G(x, \nu) + N_C(x)$. Here ∂_x denotes the subdifferential of $G(\cdot, \nu)$ and $N_C(x)$ the normal cone to C at x, both in the sense of convex analysis [20].

Of course, $x \in \psi_d(\mu)$ is equivalent to $-d \in \Gamma_\mu(x)$.

The compactness of $\operatorname{cl} U_o$, elementary properties of the convex subdifferential, and the normal cone operator $N_c(\cdot)$ together with relation (2.4) now imply that the assumptions of Proposition 6 in [15] are fulfilled. Proposition 3 in [15] then says that Γ_μ is a regular multifunction, i.e., there exists a constant $\tilde{\delta} > 0$ such that the generalized equation

$$0 \in \tilde{\Gamma}(x), \quad x \in \operatorname{cl} U_o$$

is solvable for any admissible multifunction $\tilde{\Gamma}$ satisfying

$$\Gamma_\mu(x) \subset \tilde{\Gamma}(x) + \tilde{\delta} B_m \quad \text{for all} \quad x \in \operatorname{cl} U_o,$$

where $B_m \subset \mathbb{R}^m$ denotes the closed unit ball.

For the definition of admissibility we refer to [15]. For our purposes it is sufficient to know that upper semicontinuous multifunctions with nonempty, closed, convex image sets (hence, all the mappings Γ_ν) are admissible.

Let $\nu \in \mathcal{M}_1(\mathbb{R}^s)$ such that $d(\mu, \nu; \operatorname{cl} U_o) < \tilde{\delta}$. Lemma 2.1 now implies that

$$\Gamma_\mu(x) \subset \Gamma_\nu(x) + \tilde{\delta} B_m \qquad \text{for all} \quad x \in \operatorname{cl} U_o.$$

By the regularity of Γ_μ, this yields that $\psi(\nu)$ is nonempty whenever $d(\mu, \nu; \operatorname{cl} U_o) < \tilde{\delta}$. Now select $\delta > 0$ such that $\delta < \min\{\delta', \tilde{\delta}\}$. Let $\nu \in \mathcal{M}_1(\mathbb{R}^s)$ such that $d(\mu, \nu; \operatorname{cl} U_o) < \delta$. Then it holds that $\psi(\nu) \cap \operatorname{cl} U_o \neq \emptyset$. Let us assume that $\psi(\nu) \not\subset U_o$. By convexity, this yields $\psi(\nu) \cap \operatorname{bd} U_o \neq \emptyset$. Let $\tilde{x} \in \psi(\nu) \cap \operatorname{bd} U_o$. It holds that

$$\Gamma_\nu(\tilde{x}) \subset \Gamma_\mu(\tilde{x}) + \delta B_m \quad \text{and} \quad 0 \in \Gamma_\nu(\tilde{x}).$$

Hence, there exists a $\tilde{d} \in \mathbb{R}^m$, $\|\tilde{d}\| < \delta'$ such that $\tilde{d} \in \Gamma_\mu(\tilde{x})$.

By (2.4) this implies $\tilde{x} \in \mathcal{N}_r(\psi(\mu))$, contradicting $\mathcal{N}_r(\psi(\mu)) \cap \operatorname{bd} U_o = \emptyset$, and the proof is complete. □

Now we direct our attention to stochastic programs for which Theorem 2.2 extends to the Lipschitz continuity of ψ with respect to the Hausdorff distance of sets and the pseudometric d of probability measures.

While Shapiro derived Theorem 2.2 via a general variational principle, we impose an additional structure which enables us to use more specific techniques leading to Lipschitz continuity instead of Lipschitz upper semicontinuity. Our techniques combine estimates for strongly convex functions with Lipschitz continuity results for optimal solutions to perturbed (convex) quadratic programs.

THEOREM 2.4. *Suppose* (A1)–(A3) *and that* $\psi(\mu)$ *is nonempty and bounded. Let* g *be convex quadratic and* C *be convex polyhedral. Assume that there exists a convex open subset* V *of* $\mathbb{R}^s$ *such that* $A(\psi(\mu)) \subset V$ *and the function* Q_μ *is strongly convex on* V. *Let* $U = \operatorname{cl} U_o$, *where* U_o *is an open, convex, bounded set such that* $\psi(\mu) \subset U_o$ *and* $A(U) \subset V$. *Then there exist constants* $L > 0$, $\delta > 0$ *such that*

$$d_H(\psi(\mu), \psi(\nu)) \leq L \cdot d(\mu, \nu; U)$$

whenever $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, $d(\mu, \nu; U) < \delta$.

Recall that Q_μ is said to be strongly convex on V if there exists a constant $\kappa > 0$ such that for all $\zeta, \tilde{\zeta} \in V$, and $\lambda \in [0, 1]$

$$Q_\mu(\lambda\zeta + (1-\lambda)\tilde{\zeta}) \leq \lambda Q_\mu(\zeta) + (1-\lambda) Q_\mu(\tilde{\zeta}) - \kappa\lambda(1-\lambda)\|\zeta - \tilde{\zeta}\|^2.$$

Proof. Given an open ball B_∞ (with respect to the norm $\|\cdot\|_\infty$ and around zero) such that $\psi(\mu) \subset B_\infty$, we select a $\delta > 0$ such that $\emptyset \neq \psi(\nu) \subset B_\infty$ for all $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, $d(\mu, \nu; U) < \delta$ (Proposition 2.3). We denote $C_o := C \cap \operatorname{cl} B_\infty$. Note that the compact set C_o is again a polyhedron. Then it holds for all $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, $d(\mu, \nu; U) < \delta$ that

$$\psi(\nu) = \operatorname{argmin}\{g(x) + Q_\nu(Ax) : \ x \in C_o\}.$$

Furthermore, the compactness of C_o guarantees

$$\begin{aligned} \min_x \{g(x) + Q_\nu(Ax) : x \in C_o\} &= \min_{x,y} \{g(x) + Q_\nu(y) : Ax = y, \ x \in C_o\} \\ &= \min_y \{Q_\nu(y) + \min_x \{g(x) : Ax = y, \ x \in C_o\} : y \in A(C_o)\}. \end{aligned}$$

Introducing $\pi(y) := \min_x\{g(x) : Ax = y, x \in C_o\}$, $\bar{X}(y) := \operatorname{argmin}\{g(x) : Ax = y, \ x \in C_o\}$, and $\bar{Y}(\nu) := \operatorname{argmin}\{Q_\nu(y) + \pi(y) : y \in A(C_o)\}$, we obtain by verification of the respective inclusions

$$\psi(\nu) = \bar{X}(\bar{Y}(\nu))$$

for all $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, $d(\mu, \nu; U) < \delta$.

The multifunction $\bar{X}(\cdot)$ is Lipschitzian on its effective domain $\operatorname{dom} \bar{X} = \{y \in \mathbb{R}^s : \bar{X}(y) \neq \emptyset\}$ (Satz 4.3.3 in [13], Theorem 4.2 in [14]). Therefore, there exists a constant $L_o > 0$ such that

$$d_H(\psi(\mu), \psi(\nu)) = d_H(\bar{X}(\bar{Y}(\nu)), \bar{X}(\bar{Y}(\mu))) \leq L_o \cdot \sup_{y \in \bar{Y}(\nu)} \|y - y_\mu\| \tag{2.5}$$

whenever $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, $d(\mu, \nu; U) < \delta$.

Since $\pi(\cdot)$ is convex on $A(C_o)$ and Q_μ is strongly convex on $V \supset A(\psi(\mu))$, the set $\bar{Y}(\mu)$ reduces to a singleton $\{y_\mu\}$. Moreover, the function $G(y, \mu) := Q_\mu(y) + \pi(y)$ is strongly convex on V with modulus $\kappa > 0$.

Decrease, if necessary, $\delta > 0$ such that $\psi(\nu) \subset U_o$ whenever $\nu \in \mathcal{M}_1(\mathbb{R}^s)$, $d(\mu, \nu; U) < \delta$ (Proposition 2.3). Let $y \in \bar{Y}(\nu)$. Then $\bar{X}(y) \subset \psi(\nu)$. Since $\psi(\nu) \subset U_o$ and $\{y\} = A\bar{X}(y)$, it follows that $y \in A(U_o) \subset V$. Consider the point $\frac{1}{2}(y + y_\mu)$ belonging to $A(C_o) \cap V$.

Then

$$\begin{aligned} G(y_\mu, \mu) &\leq G\left(\frac{1}{2}(y + y_\mu), \mu\right) \\ &\leq \frac{1}{2}G(y, \mu) + \frac{1}{2}G(y_\mu, \mu) - \frac{\kappa}{4}\|y - y_\mu\|^2 \end{aligned}$$

by the strong convexity of $G(\cdot, \mu)$ on V.

This implies (if $\|y - y_\mu\| \neq 0$)

$$\begin{aligned} \frac{\kappa}{2}\|y - y_\mu\| &\leq \frac{G(y, \mu) - G(y_\mu, \mu)}{\|y - y_\mu\|} \\ &= \frac{(G(y_\mu, \nu) - G(y_\mu, \mu)) - (G(y_\mu, \nu) - G(y, \mu))}{\|y - y_\mu\|} \\ &\leq \frac{(G(y_\mu, \nu) - G(y_\mu, \mu)) - (G(y, \nu) - G(y, \mu))}{\|y - y_\mu\|} \\ &\leq \frac{|(Q_\nu - Q_\mu)(y_\mu) - (Q_\nu - Q_\mu)(y)|}{\|y - y_\mu\|}. \end{aligned}$$

By Lebourg's mean value theorem [2] there exists a point y^* on the line segment $[y_\mu, y]$ (belonging entirely to $A(U)$) such that the above estimate continues

$$\leq \sup \left| \left\langle \partial(Q_\nu - Q_\mu)(y^*), \frac{y_\mu - y}{\|y_\mu - y\|} \right\rangle \right|.$$

Hence,

$$\begin{aligned} \sup_{y \in \bar{Y}(\nu)} \|y - y_\mu\| &\leq \frac{2}{\kappa} \sup\{\|z^*\| : z^* \in \partial(Q_\nu - Q_\mu)(y) : y \in A(U)\} \\ &= \frac{2}{\kappa} d(\mu, \nu; U). \end{aligned}$$

Together with (2.5) this completes the proof. □

COROLLARY 2.5. *Adopt the setting of Theorem* 2.4. *Then there exist nonsingular matrices* B_i $(i = 1, \ldots, \ell)$ *and a constant* $L > 0$ *such that*

$$d_H(\psi(\mu), \psi(\nu)) \leq L \sum_{i=1}^{\ell} \sup_{t \in A(U)} |F_{\mu \circ (-B_i)}(-B_i^{-1}t) - F_{\nu \circ (-B_i)}(-B_i^{-1}t)|$$

whenever $\nu \in \mathcal{M}_1(\mathbb{R}^s)$ is chosen such that the right-hand side is sufficiently small. The notation F refers to the distribution function of the probability measure in the subscript.

Proof. Using Theorem 2.5.1 in [2] we obtain the following representation of $d(\mu, \nu; U)$:

$$d(\mu, \nu; U) = \sup\{\|\nabla(Q_\nu - Q_\mu)(Ax)\| : x \in U \setminus E\}, \tag{2.6}$$

where E contains those $x \in \mathbb{R}^m$ such that $Q_\nu - Q_\mu$ is not differentiable at Ax and $A(E)$ has Lebesgue measure zero.

Recall that the integrand Q in (1.2) is a piecewise linear convex function on $\mathbb{R}^s$ and that there exist basis submatrices $B_1, \ldots, B_\ell$ of W such that the simplicial cones $B_1(\mathbb{R}^s_+); \ldots, B_\ell(\mathbb{R}^s_+)$ are linearity regions of Q (in general not the maximal ones) (cf. [6], [16], Satz 6.7). Of course, $\bigcup_{i=1}^{\ell} B_i(\mathbb{R}^s_+) = \mathbb{R}^s$, and B_i $(i = 1, \ldots, \ell)$ can be chosen in such a way that $B_i(\mathbb{R}^s_+) \cap B_j(\mathbb{R}^s_+)$, $i \neq j$, is always contained in a hyperplane in $\mathbb{R}^s$. Thus, $\mathcal{H} := \mathbb{R}^s \setminus \bigcup_{i=1}^{\ell} \operatorname{int} B_i(\mathbb{R}^s_+)$ is contained in a finite union of hyperplanes in $\mathbb{R}^s$.

Let us now confirm that, for some single hyperplane $\mathcal{H}_o \subset \mathbb{R}^s$, the set $\mathcal{Z}_{\nu,o} := \{\zeta \in \mathbb{R}^s : \nu(\zeta + \mathcal{H}_o) > 0\}$ has Lebesgue measure zero. It holds that

$$\begin{aligned} \zeta + \mathcal{H}_o &= \zeta + \{t \in \mathbb{R}^s : a^T t = 0\} \\ &= \{t \in \mathbb{R}^s : a^T(t - \zeta) = 0\} = a^{-1}(\{a^T \zeta\}), \end{aligned}$$

where $a : \mathbb{R}^m \to \mathbb{R}$ denotes the linear transformation induced by a^T and a^{-1} is the preimage.

Hence, $\mathcal{Z}_{\nu,o} = \{\zeta \in \mathbb{R}^s : \nu \circ a^{-1}(\{a^T\zeta\}) > 0\}$. Now $\nu \circ a^{-1}$ is a probability measure on $\mathbb{R}$ and $a^T\zeta$ is an atom of $\nu \circ a^{-1}$. Since $\nu \circ a^{-1}$ has at most countably many atoms, $\mathcal{Z}_{\nu,o}$ is contained in a countable union of hyperplanes and has Lebesgue measure zero.

Therefore the sets

$$\mathcal{Z}_\mu := \{\zeta \in \mathbb{R}^s : \mu(\zeta + \mathcal{H}) > 0\} \text{ and } \mathcal{Z}_\nu := \{\zeta \in \mathbb{R}^s : \nu(\zeta + \mathcal{H}) > 0\}$$

have Lebesgue measure zero.

Now select E in (2.6) as the preimage $A^{-1}(\mathcal{Z}_\mu \cup \mathcal{Z}_\nu)$. Then both Q_ν and Q_μ are differentiable at Ax for all $x \in U \setminus E$ and we have for those x

$$\begin{aligned} \nabla(Q_\nu - Q_\mu)(Ax) &= \nabla\left(\int_{\mathbb{R}^s \setminus \{Ax + \mathcal{H}\}} Q(z - Ax)\nu(\mathrm{d}z) - \int_{\mathbb{R}^s \setminus \{Ax+\mathcal{H}\}} Q(z - Ax)\mu(\mathrm{d}z) \right) \\ &= \int_{\mathbb{R}^s \setminus \{Ax+\mathcal{H}\}} \nabla Q(z - Ax)(\nu - \mu)\mathrm{d}z, \end{aligned}$$

where the first identity is valid because $\nu(Ax + \mathcal{H}) = \mu(Ax + \mathcal{H}) = 0$. The second identity follows from Lebesgue's theorem on dominated convergence. Indeed, since Q is globally Lipschitzian on $\mathbb{R}^s$, the difference quotients related to ∇Q are bounded above by a uniform constant yielding the integrable majorant making Lebesgue's theorem work.

We continue the above identity as follows:

$$\nabla(Q_\nu - Q_\mu)(Ax) = \int_{Ax + \bigcup_{i=1}^{\ell} \operatorname{int} B_i(\mathbb{R}^s_+)} \nabla Q(z - Ax)(\nu - \mu)(\mathrm{d}z)$$

$$= \sum_{i=1}^{\ell} d_i(\nu - \mu)(Ax + B_i(\mathbb{R}^s_+))$$
$$= \sum_{i=1}^{\ell} d_i(F_{\nu\circ(-B_i)}(-B_i^{-1}Ax) - F_{\mu\circ(-B_i)}(-B_i^{-1}Ax)),$$

where $-d_i$ is the gradient of Q on int $B_i(\mathbb{R}^s_+)$, $i = 1, \ldots, \ell$. Note that for the second identity we have also used that $\mu(Ax + \text{int } B_i(\mathbb{R}^s_+)) = \mu(Ax + B_i(\mathbb{R}^s_+))$, $\nu(Ax + \text{int } B_i(\mathbb{R}^s_+)) = \nu(Ax + B_i(\mathbb{R}^s_+))$ for all $x \in U \setminus E$.

The assertion now immediately follows from Theorem 2.4, (2.6), and the above identity. Since $A(U \setminus E)$ is less explicitly known than $A(U)$, we finally take the supremum over the larger set $A(U)$. □

Remark 2.6. The above estimate is closely related to Theorem 2.1 in [29], where the author uses the normal cones $\bar{C}_j$ $(j = 1, \ldots, k)$ to the set $\{u \in \mathbb{R}^s : W^T u \leq q\}$ at its vertices v_j $(j = 1, \ldots, k)$. From linear parametric programming it is known ([16], Satz 6.7) that each of the cones $\bar{C}_j$ $(j = 1, \ldots, k)$ is the union of certain cones $B_i(\mathbb{R}^s_+)$ $(i \in \{1, \ldots, \ell\})$ arising in Corollary 2.5.

We prefer to use the cones $B_i(\mathbb{R}^s_+)$ since these are simplicial cones, which allows a direct relation to distribution functions.

Remark 2.7. Consider the right-hand side of the estimate in Corollary 2.5 and take the suprema with respect to $t \in \mathbb{R}^s$ instead of $t \in A(U)$. In this way we obtain a Lipschitz estimate with respect to the uniform (or Kolmogorov–Smirnov) distance of the distribution functions $F_{\mu\circ(-B_i)}$ and $F_{\nu\circ(-B_i)}$ $(i = 1, \ldots, \ell)$.

Remark 2.8. Theorem 2.4 remains valid under any hypothesis on g and C leading to Lipschitz continuity of the multifunction $\bar{X}(y) := \text{argmin}\{g(x) : Ax = y,\ x \in C\}$.

The next example shows that (already for contaminated distributions) Theorem 2.4 is lost for a general closed convex set $C \subset \mathbb{R}^m$. Another counterexample involving the function g can be constructed following the guidelines of Remark 2.9 in [22].

Example 2.9. In (1.1)–(1.3) let $m = 2$, $s = 1$, $g(x) \equiv 0$, $A = (1, 0)$, $C = \{x \in \mathbb{R}^2 : (x_2)^2 \leq x_1\}$, $q = (1, 1)^T$, $W = (1, -1)$, and μ be the uniform distribution on $[-1/2, 1/2]$. Let δ_1 denote the probability measure on $\mathbb{R}$ having unit mass at 1 and construct perturbations μ_t of μ by setting $\mu_t = (1 - t)\mu + t\delta_1$, $t \in [0, 1]$. Then $\psi(\mu) = \{0\}$ and the strong convexity assumption for Q_μ holds for $V = (-1/2, 1/2)$. Furthermore, one computes that $(x_{1,t}, \sqrt{x_{1,t}})^T \in \psi(\mu_t)$ for $0 < t < 2/3$, where $x_{1,t} = t/2(1 - t)$. Hence, $x_{1,t} > \frac{1}{2}t$ and $d_H(\psi(\mu), \psi(\mu_t)) \geq \frac{1}{\sqrt{2}}\sqrt{t}$ for all $t \in (0, 2/3)$. With $U \subset \mathbb{R}^2$ taken as the closed ball around zero with radius 1/2 (for instance), one confirms that $d(\mu, \mu_t; U) = \text{const} \cdot t$, i.e., the assertion of Theorem 2.4 does not hold.

Note that in Example 2.9 there is even no upper Lipschitz continuity of ψ. This indicates that, in general, one cannot hope to obtain the second-order growth condition (2.3) in Theorem 2.2 without adding assumptions on g and C.

Remark 2.10. Using similar techniques as in the proof of Theorem 2.7 in [22], it can be shown that the assumptions from Theorem 2.4 imply that the second-order growth condition (2.3) in Theorem 2.2 holds.

The following example concludes the comparison between Theorems 2.2 and 2.4. This example demonstrates that the setting in Theorem 2.2 is the more general one. Indeed, Theorem 2.2 does not guarantee the lower semicontinuity of the mapping ψ which, of course, is a special implication of the Hausdorff–Lipschitz result in Theorem 2.4.

Example 2.11. In (1.1)–(1.3) let $m = s = 1$, $g(x) \equiv 0$, $A = 1$, $C = [-1, 1]$, $q = (1, 1)^T$, $W := (1, -1)$ ("simple recourse"), and μ be the uniform distribution on $[-1, -1/2] \cup [1/2, 1]$. Let $\bar{\mu}$ be the uniform distribution on $[-1/2, 1/2]$ and construct perturbations μ_t of μ by setting $\mu_t = \mu + t(\bar{\mu} - \mu)$, $t \in [0, 1]$ ("contaminated distributions"). Then one computes that $\psi(\mu) = [-1/2, 1/2]$ and that

$$Q_\mu(x) \geq Q_\mu(x_\mu) + [\mathrm{dist}(x, \psi(\mu))]^2$$

for all $x \in (-1, 1)$ and all $x_\mu \in \psi(\mu)$. Hence (2.3) is fulfilled and Theorem 2.2 applies. On the other hand, $\psi(\mu_t) = \{0\}$ for all $t \in (0, 1]$ and $d(\mu, \mu_t; C) = \mathrm{const} \cdot t$. Thus, ψ does not share the Lipschitz property from Theorem 2.4; moreover, ψ is not lower semicontinuous at μ.

The next example is interesting because it shows that Theorem 2.4 does not ensure the stability of the optimal value.

Example 2.12. In (1.1)–(1.3) let $m = s = 1$, $g(x) \equiv 0$, $C = [-1, 1]$, $A = 1$, $q = (1, 1)$, $W = (1, -1)$, $\mu = \delta_o$, and construct pertrubations μ_n of μ by setting $\mu_n = (1 - \frac{1}{n})\delta_o + \frac{1}{n}\delta_{n^2}$ ($n \in \mathbb{N}$). Then we have $Q_\mu(x) = |x|$ and, thus, $\varphi(\mu) = 0$, $\psi(\mu) = \{0\}$. Furthermore, $Q_{\mu_n}(x) = (1 - \frac{1}{n})|x| + \frac{1}{n}(n^2 - x)$. Therefore, $\psi(\mu_n) = \{0\}$, $\varphi(\mu_n) = n$. The assumptions of Proposition 2.3 and Theorem 2.4 are fulfilled, but $\varphi(\mu_n) \to \infty$ as $n \to \infty$.

The following result (established in [24]) provides a handy tool for checking the strong convexity of Q_μ needed in Theorem 2.4.

PROPOSITION 2.13 ([24], Theorem 2.2). *Let $V \subset \mathbb{R}^s$ be open and convex. Assume*

(a) (A1),
(b) *there exists a $\bar{u} \in \mathbb{R}^s$ such that $W^T\bar{u} < q$ componentwise,*
(c) (A3),
(d) *μ has a density Θ_μ on $\mathbb{R}^s$,*
(e) *there exist constants $r > 0$, $\varrho > 0$ such that $\Theta_\mu(t) \geq r$ for all $t \in \mathbb{R}^s$ such that* $\mathrm{dist}(t, V) \leq \varrho$.

Then Q_μ is strongly convex on V.

In [22], [23] the quantitative continuity of the mapping ψ is studied with respect to the L_1-Wasserstein distance $W_1(\mu, \nu)$ for measures μ, ν in $\mathcal{M}_1(\mathbb{R}^s)$ [17]. In fact, Theorem 2.7 in [22] states the Hölder continuity (with exponent 1/2) of $d_H(\psi(\mu), \psi(\nu))$ with respect to $W_1(\mu, \nu)$ under precisely the same assumptions as in Theorem 2.4 in the present paper. Furthermore, [22] contains an example (Remark 2.9) showing the optimality of the convergence rate 1/2. We now analyze this example with respect to the pseudometric $d(\mu, \nu; U)$. The setting is as in Examples 2.11 and 2.12, but μ is taken as the uniform distribution on $[-1/2, 1/2]$ and the perturbations μ_n are given by the distribution functions

$$F_{\mu_n}(t) = \begin{cases} \frac{1}{2} & t \in [-\varepsilon_n, \varepsilon_n), \\ F_\mu(t) & \text{otherwise}, \end{cases}$$

where (ε_n) is arbitrary and tending to zero from above ($n \to \infty$). It holds that $\psi(\mu) = \{0\}$ and $\psi(\mu_n) = [-\varepsilon_n, \varepsilon_n]$. The assumptions of Theorem 2.4 are fulfilled. In [22] we computed $W_1(\mu, \mu_n) = \varepsilon_n^2$, which showed the optimality of the Hölder exponent 1/2. Here we obtain that, with $U = [-1/4, 1/4]$, $d(\mu, \mu_n; U) = \mathrm{const} \cdot \varepsilon_n$. Hence, in the worst case, Theorem 2.4 does not outperform Theorem 2.7 in [22]. However, for certain specific modes of perturbation, Theorem 2.4 yields stronger estimates than Theorem 2.7 in [22] (contaminated distributions, asymptotic properties of nonparametric estimators; see the analysis that follows).

To further explain the essence of Theorem 2.4, let us mention that Theorem 2.7 in [22] also yields the convergence rate 1/2 when one replaces the Wasserstein distance W_1 by

$$d^*(\mu,\nu;U) = \sup\{|(Q_\nu - Q_\mu)(Ax)| : x \in U\},$$

where $U \subset \mathbb{R}^m$ is a suitable nonempty, convex, compact set. Therefore, the approach taken here differs from older ones because it measures the distance of the objectives in the original and the perturbed programs in terms of their subgradients rather than in terms of their function values. Of course $d(\mu,\nu;U)$ may tend to zero while $d^*(\mu,\nu;U)$ does not, which explains the collapse of optimal-value convergence observed in Example 2.12. Hence, when aiming at the stability of the optimal value one should resort to a distance like d^*. In [15], Proposition 8, it is shown that convergence to zero of d^* does imply the same for d, provided that the original function Q_μ is differentiable.

PROPOSITION 2.14 (contaminated distributions). *Suppose* (A1)–(A3) *and that* $\psi(\mu)$ *is nonempty and bounded. Let* g *be convex quadratic and* C *be polyhedral. Assume that there exists a convex open subset* V *of* $\mathbb{R}^s$ *such that* $A(\psi(\mu)) \subset V$ *and the function* Q_μ *is strongly convex on* V. *Let* $\bar{\mu} \in \mathcal{M}_1(\mathbb{R}^s)$ *be arbitrarily fixed and define* $\mu_t = (1-t)\mu + t\bar{\mu}$, $t \in [0,1]$. *Then there exist constants* $L > 0$ *and* $t_o > 0$ *such that*

$$d_H(\psi(\mu), \psi(\mu_t)) \le Lt$$

for all $t \in [0, t_o]$.

Proof. Note that $Q_{\mu_t} - Q_\mu = t(Q_{\bar{\mu}} - Q_\mu)$. Calculus rules for the Clarke subdifferential thus imply that $d(\mu;\mu_t;U) = t \cdot d(\mu,\bar{\mu};U)$, where, of course, $d(\mu,\bar{\mu};U)$ is a constant. The result now immediately follows from Theorem 2.4. □

The following result refers to the special case of simple recourse, i.e., Q in (1.3) is given by

$$Q(t) := \min\{q^{+^T} y^+ + q^{-^T} y^- : y^+ - y^- = t,\ y^+ \ge 0,\ y^- \ge 0\}, \tag{2.7}$$

where $\bar{m} = 2s$ and $q^+, q^- \in \mathbb{R}^s$. Shapiro has derived a similar result (Theorem 3.1 in [29]) by a direct estimate from Theorem 2.2.

PROPOSITION 2.15. *Let* $P(\mu)$ *be a simple recourse model,* $\psi(\mu)$ *be nonempty and bounded,* g *be convex quadratic, and* C *be a nonempty polyhedron. Assume that* $q^+ + q^- > 0$ *(componentwise) and that all the one-dimensional marginal distributions* μ_j *of* μ $(j = 1,\dots,s)$ *have finite first moments and densities that are positively bounded below on some open neighbourhoods of the orthogonal projections of* $\psi(\mu)$ *to the coordinate axes. Then there exists a constant* $L > 0$ *such that*

$$d_H(\psi(\mu), \psi(\nu)) \le L \cdot \sum_{j=1}^{s} \sup_{t_j \in \mathrm{proj}_j(A(U))} |F_{\mu_j}(t_j) - F_{\nu_j}(t_j)|$$

whenever $\nu \in \mathcal{M}_1(\mathbb{R}^s)$ *is chosen such that the right-hand side is sufficiently small.*

Proof. First note that the function Q in (2.7) is separable with respect to the components of t. Therefore, the functions Q_ν and Q_μ here depend only on the one-dimensional marginal distributions of ν and μ (cf. also [6], [31]), and we can assume without loss of generality that ν, μ are probability measures with independent one-dimensional marginals. Then our assumptions and Proposition 2.13 yield that Q_μ is strongly convex on some convex open set $V \supset A(\psi(\mu))$.

Here the cones $B_i(\mathbb{R}^s_+)$ $(i = 1, \dots, 2^s)$ are orthants. To estimate $(\nu - \mu)(Ax + B_i(\mathbb{R}^s_+))$, let us fix some $B_i(\mathbb{R}^s_+)$ for which we assume without loss of generality that

$$B_i(\mathbb{R}^s_+) = \bigtimes_{j=1}^{s_i}(-\infty, 0] \times \bigtimes_{j=s_i+1}^{s}[0, +\infty).$$

Our independence assumption then yields

$$\begin{aligned}(\nu - \mu)(Ax + B_i(\mathbb{R}^s_+)) = &\prod_{j=1}^{s_i}\nu_j((-\infty, (Ax)_j]) \cdot \prod_{j=s_i+1}^{s}\nu_j([(Ax)_j, +\infty)) \\ &- \prod_{j=1}^{s_i}\mu_j((-\infty, (Ax)_j]) \cdot \prod_{j=s_i+1}^{s}\mu_j([(Ax)_j, +\infty)).\end{aligned}$$

Using the inequality

$$\left|\prod_{j=1}^{s}\alpha_j - \prod_{j=1}^{s}\beta_j\right| \le \sum_{j=1}^{s}|\alpha_j - \beta_j| \quad \text{for } 0 \le \alpha_j, \beta_j \le 1,\ j = 1, \dots, s$$

(which can be shown by induction), we obtain

$$\begin{aligned}&|(\nu - \mu)(Ax + B_i(\mathbb{R}^s_+))| \\ &\le \sum_{j=1}^{s_i}|F_{\nu_j}((Ax)_j) - F_{\mu_j}((Ax)_j)| + \sum_{j=s_i+1}^{s}|F^-_{\nu_j}((Ax)_j) - F^-_{\mu_j}((Ax)_j)|,\end{aligned}$$

where the superscripts in the last term indicate limits from the left.

For $x \in U \setminus E$ (with E as in the proof of Corollary 2.5) the superscripts can be dropped and we obtain

$$|(\nu - \mu)(Ax + B_i(\mathbb{R}^s_+))| \le \sum_{j=1}^{s}|F_{\nu_j}((Ax)_j) - F_{\mu_j}((Ax)_j)|,$$

and the proof is completed as with Corollary 2.5. □

3. Applications to asymptotic analysis. In this section, we show how to employ the Lipschitz stability result of §2 to derive asymptotic properties of optimal solutions when estimating μ in $P(\mu)$ by empirical measures. We obtain a law of iterated logarithm, a large deviation estimate, and an estimate for the asymptotic distribution of the optimal solution sets without imposing that $\psi(\mu)$ must be a singleton. The basic tools are known limit theorems for the Kolmogorov–Smirnov distance of the empirical distribution function. Let $\xi_1, \xi_2, \dots, \xi_n, \dots$ be independent $\mathbb{R}^s$-valued random variables on a probability space $(\Omega, \mathfrak{A}, \mathbf{P})$ having joint distribution μ. Let δ_z denote the probability measure assigning unit mass to $z \in \mathbb{R}^s$. We consider the empirical measures

$$\mu_n(\omega) := \frac{1}{n}\sum_{i=1}^{n}\delta_{\xi_i(\omega)} \qquad (\omega \in \Omega;\ n \in \mathbb{N}),$$

and we are interested in the asymptotic behaviour of the solution set $\psi(\mu_n(\cdot))$ of $P(\mu_n(\cdot))$ as n tends to infinity. Our results are put in terms of the Hausdorff distance $d_H(\psi(\mu), \psi(\mu_n(\cdot)))$, which is a $\mathfrak{A}$-measurable mapping due to Theorem 2K in [21].

PROPOSITION 3.1. *Under the assumptions of Theorem* 2.4 *it holds that*

$$\limsup_{n\to\infty}\Big(\frac{2n}{\log\log n}\Big)^{\frac{1}{2}}\cdot d_H(\psi(\mu),\psi(\mu_n(\omega)))\le L\ell \quad \mathbf{P}\text{-almost surely,}$$

where L *and* ℓ *denote the Lipschitz modulus and the number of basis matrices, respectively, arising in Corollary* 2.5.

Proof. Let B_j, $j=1,\dots,\ell$, denote the relevant basis submatrices of W. Then $\mu_n(\omega)\circ(-B_j)$ coincides with the empirical measure of $\mu\circ(-B_j)$ and the following law of iterated logarithm holds ([18], p. 302, [25]):

$$\limsup_{n\to\infty}\Big(\frac{n}{2\log\log n}\Big)^{\frac{1}{2}}\sup_{t\in\mathbb{R}^s}|F_{\mu\circ(-B_j)}(t)-F_{\mu_n(\omega)\circ(-B_j)}(t)|\le\frac{1}{2}$$

$\mathbf{P}$-almost surely for all $j=1,\dots,\ell$.

Hence, the estimate from Corollary 2.5 is valid for $\mathbf{P}$-almost all $\omega\in\Omega$ with $\nu:=\mu_n(\omega)$, provided that $n=n(\omega)\in\mathbb{N}$ is sufficiently large.

Thus we have for $\mathbf{P}$-almost all $\omega\in\Omega$

$$\limsup_{n\to\infty}\Big(\frac{n}{2\log\log n}\Big)^{\frac{1}{2}}d_H(\psi(\mu),\psi(\mu_n(\omega)))\le\frac{L\ell}{2}. \qquad \square$$

In [4], [12] the authors obtain consistency results under weak hypotheses on the optimization problems involved (based on the theory of epi-convergence). Proposition 3.1 supplements these results by giving, under stronger assumptions, the (optimal) rate of convergence for the solution sets. Compared to considerations in [29], we can dispense with the linear-independence assumptions imposed there. This became possible because we used simplical cones instead of more general ones (cf. Remark 2.6). Compared to [4], [33] we do not need the unique solvability of $P(\mu)$.

PROPOSITION 3.2. *Under the assumptions of Theorem* 2.4 *there exists a constant* $\varepsilon_o>0$ *such that it holds for all* $\varepsilon\in(0,\varepsilon_o]$ *that*

$$\limsup_{n\to\infty}\frac{1}{n}\log\mathbf{P}(\{\omega: d_H(\psi(\mu),\psi(\mu_n(\omega)))\ge\varepsilon\})\le-2\Big(\frac{\varepsilon}{L\ell}\Big)^2.$$

Proof. For brevity, we introduce the following notation:

$$\Phi_n(\omega):=\max_{j=1,\dots,\ell}\eta_{n,j}(\omega),$$
$$\eta_{n,j}(\omega):=\sup_{t\in\mathbb{R}^s}|F_{\mu\circ(-B_j)}(t)-F_{\mu_n(\omega)\circ(-B_j)}(t)| \quad (\omega\in\Omega).$$

Now select $\varepsilon_o>0$ in such a way that $L\ell\Phi_n(\omega)<\varepsilon_o$ and Corollary 2.5 imply

$$d_H(\psi(\mu),\psi(\mu_n(\omega)))\le L\ell\Phi_n(\omega).$$

Then we have for each $\varepsilon\in(0,\varepsilon_o]$ and all $n\in\mathbb{N}$

$$\{\omega: d_H(\psi(\mu),\psi(\mu_n(\omega)))\ge\varepsilon\}\subseteq\Big\{\omega:\Phi_n(\omega)\ge\frac{\varepsilon_o}{L\ell}\Big\}\cup\Big\{\omega:\Phi_n(\omega)\ge\frac{\varepsilon}{L\ell}\Big\}$$
$$=\Big\{\omega:\Phi_n(\omega)\ge\frac{\varepsilon}{L\ell}\Big\}=\bigcup_{j=1}^{\ell}\Big\{\omega:\eta_{n,j}(\omega)\ge\frac{\varepsilon}{L\ell}\Big\}$$

and, hence,

$$\mathbf{P}(\{\omega : d_H(\psi(\mu), \psi(\mu_n(\omega))) \geq \varepsilon\}) \leq \sum_{j=1}^{\ell} \mathbf{P}\Big(\Big\{\omega : \eta_{n,j}(\omega) \geq \frac{\varepsilon}{L\ell}\Big\}\Big).$$

The multivariate version of the Dvoretzky–Kiefer–Wolfowitz inequality (cf. [9], [25]) then implies that, for each $\delta > 0$ and $j \in \{1, \ldots, \ell\}$, there exist constants $C_j > 0$ such that

$$\mathbf{P}\Big(\Big\{\omega : \eta_{n,j}(\omega) \geq \frac{\varepsilon}{L\ell}\Big\}\Big) \leq C_j \exp\Big(-(2-\delta)n\Big(\frac{\varepsilon}{L\ell}\Big)^2\Big) \quad \text{for all } n \in \mathbb{N}.$$

Hence, we obtain for any $n \in \mathbb{N}$ and any $\delta > 0$,

$$\frac{1}{n} \log \mathbf{P}(\{\omega : d_H(\psi(\mu); \psi(\mu_n(\omega)) \geq \varepsilon\}) \leq \frac{1}{n} \log\Big(\sum_{j=1}^{\ell} C_j\Big) - (2-\delta)\Big(\frac{\varepsilon}{L\ell}\Big)^2,$$

and the proof is complete. □

Compared to Theorem 4.6 in [8], which represents a large deviation result for more general stochastic programs, Proposition 3.2 requires only the weak moment condition (A3) and yields an explicit estimate instead of an implicit one involving a conditioning function that is often hard to quantify. We also refer to the exponential bound in Theorem 2 in [30] which, in the context of two-stage stochastic programming, works for nonunique solutions but applies only to measures μ with bounded support.

Another substantial step in the asymptotic analysis of optimal solutions involves obtaining asymptotic distributions of the sequence of closed random sets

$$(n^{1/2}(\psi(\mu_n(\cdot)) - x))_{n \in \mathbb{N}} \qquad (\text{for each } x \in \psi(\mu))$$

on the hyperspace of closed subsets of $\mathbb{R}^m$. In [11], [27] this problem was tackled for stochastic programs involving expectation functions with smooth integrands. Moreover, it was assumed that the unperturbed problem has a unique optimal solution. For stochastic programs with complete recourse the relevant integrands are typically nonsmooth (cf. (1.2), (1.3)) and uniqueness of optimal solutions is rather exceptional (cf. Example 1.1) such that the results from [11], [27] do not apply.

From Theorem 2.4, however, a lower estimate for the asymptotic distribution of

$$(n^{1/2} d_H(\psi(\mu), \psi(\mu_n(\cdot))))_{n \in \mathbb{N}}$$

can be derived. This is done next. The result is inspired by the concept of normalized convergence and the corresponding techniques in [5]. For simple recourse models the lower estimate becomes more detailed (Remark 3.4).

PROPOSITION 3.3. *Under the assumptions of Theorem* 2.4 *there exist probability distribution functions* G_j, $j = 1, \ldots, \ell$, *on* $\mathbb{R}$ *such that it holds that*

$$\liminf_{n \to \infty} \mathbf{P}(\{\omega : n^{\frac{1}{2}} d_H(\psi(\mu), \psi(\mu_n(\omega))) < t\}) \geq 1 + \sum_{j=1}^{\ell} \Big(G_j\Big(\frac{t}{L\ell}\Big) - 1\Big)$$

for all $t \geq 0$, *where* L *and* ℓ *denote the Lipschitz modulus and the number of basis matrices, respectively, arising in Corollary* 2.5.

Proof. Let $\eta_{n,j}(\omega)$ be given as in the proof of Proposition 3.2.

From the asymptotic distribution theory for the Kolmogorov–Smirnov distance it is known that for each $j = 1, \dots, \ell$ the sequence

$$(n^{\frac{1}{2}} \eta_{n,j}(\cdot))_{n \in \mathbb{N}}$$

converges in distribution to some real random variable ζ_j (Theorem 2 in [10], Chap. 2.1.5 in [25]).

Let $t \geq 0$, $n \in \mathbb{N}$ and consider the following events in $\mathfrak{A}$:

$$A := \left\{ \omega : Ln^{\frac{1}{2}} \sum_{j=1}^{\ell} \eta_{n,j}(\omega) < t \right\},$$

$$A_j := \left\{ \omega : n^{\frac{1}{2}} \eta_{n,j}(\omega) < \frac{t}{L\ell} \right\} \qquad (j = 1, \dots, \ell),$$

$$B_\delta := \left\{ \omega : L \sum_{j=1}^{\ell} \eta_{n,j}(\omega) < \delta \right\},$$

where $\delta > 0$ is selected according to Corollary 2.5 such that, for all $\omega \in B_\delta$, $d_H(\psi(\mu), \psi(\mu_n(\omega)))$ can be estimated by the expression defining B_δ.

Corollary 2.5 then yields the following chain of inequalities:

$$\begin{aligned}
&\mathbf{P}(\{\omega : n^{\frac{1}{2}} d_H(\psi(\mu), \psi(\mu_n(\omega))) < t\}) \\
&\geq \mathbf{P}(\{\omega : n^{\frac{1}{2}} d_H(\psi(\mu), \psi(\mu_n(\omega))) < t\} \cap B_\delta) \\
&\geq \mathbf{P}(A \cap B_\delta) \geq \mathbf{P}\left(\bigcap_{j=1}^{\ell} A_j \cap B_\delta \right) \\
&= \mathbf{P}\left(\bigcap_{j=1}^{\ell} A_j \right) - \mathbf{P}\left(\bigcap_{j=1}^{\ell} A_j \cap \bar{B}_\delta \right) \\
&\geq \mathbf{P}\left(\bigcap_{j=1}^{\ell} A_j \right) - \mathbf{P}(\bar{B}_\delta) \\
&= 1 - \mathbf{P}\left(\bigcup_{j=1}^{\ell} \bar{A}_j \right) - \mathbf{P}(\bar{B}_\delta) \\
&\geq 1 - \sum_{j=1}^{\ell} \mathbf{P}(\bar{A}_j) - \mathbf{P}(\bar{B}_\delta) = 1 + \sum_{j=1}^{\ell} (\mathbf{P}(A_j) - 1) - \mathbf{P}(\bar{B}_\delta).
\end{aligned}$$

Hence we obtain the following estimate for all $t \geq 0$ and $n \in \mathbb{N}$:

$$\begin{aligned}
&\mathbf{P}(\{\omega : n^{\frac{1}{2}} d_H(\psi(\mu), \psi(\mu_n(\omega))) < t\}) \\
&\geq 1 + \sum_{j=1}^{\ell} \left(\mathbf{P}\left(\left\{ \omega : n^{\frac{1}{2}} \eta_{n,j}(\omega) < \frac{t}{L\ell} \right\} \right) - 1 \right) - \mathbf{P}\left(\left\{ \omega : L \sum_{j=1}^{\ell} \eta_{n,j}(\omega) \geq \delta \right\} \right).
\end{aligned}$$

The latter probability tends to zero as $n \to \infty$ because of the Glivenko–Cantelli theorem, and we finally obtain the following via the Portmanteau theorem for each $t \geq 0$:

$$\liminf_{n \to \infty} \mathbf{P}(\{\omega : n^{\frac{1}{2}} d_H(\psi(\mu), \psi(\mu_n(\omega))) < t\})$$

$$\geq 1 + \sum_{j=1}^{\ell} \Big(\liminf_{n\to\infty} \mathbf{P}\Big(\Big\{ \omega : n^{\frac{1}{2}} \eta_{n,j}(\omega) < \frac{t}{L\ell} \Big\} - 1 \Big) \Big)$$

$$\geq 1 + \sum_{j=1}^{\ell} \Big(\mathbf{P}\Big(\Big\{ \omega : \zeta_j(\omega) < \frac{t}{L\ell} \Big\} \Big) - 1 \Big) = 1 + \sum_{j=1}^{\ell} \Big(G_j\Big(\frac{t}{L\ell} \Big) - 1 \Big),$$

where $G_j(u) := \mathbf{P}(\{\omega : \zeta_j(\omega) < u\})$ for all $u \in \mathbb{R}$. □

Remark 3.4. Unfortunately, the limit distributions G_j $(j = 1, \ldots, \ell)$ cannot be characterized in general for multidimensional distributions $F_{\mu \circ (-B_j)}$ (see [10]). However, in the case of simple recourse, we obtain the following estimate by using Proposition 2.15 instead of Corollary 2.5 and under the assumption that all one-dimensional marginal distributions of μ are continuous:

$$\liminf_{n\to\infty} \mathbf{P}(\{\omega : n^{\frac{1}{2}} d_H(\psi(\mu), \psi(\mu_n(\omega))) < t\}) \geq 1 + s\Big(H\Big(\frac{t}{Ls} \Big) - 1 \Big)$$

for all $t \geq 0$, where $H(u) := 1 - 2\sum_{k=1}^{\infty} (-1)^{k-1} e^{-2k^2u^2}$ $(u \geq 0)$ is the asymptotic distribution in the Kolmogorov limit theorem.

Acknowledgments. This paper has benefited from several discussions with Alexander Shapiro (Georgia Institute of Technology, Atlanta) and Diethard Klatte (University of Zurich). Moreover, the suggestions of a referee that led to a more transparent proof of Theorem 2.4 are gratefully acknowledged.

REFERENCES

[1] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Non-Linear Parametric Optimization*, Akademie-Verlag, Berlin, 1982.

[2] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[3] J. Dupačová, *Stability and sensitivity analysis for stochastic programming*, Ann. Oper. Res., 27 (1990), pp. 115–142.

[4] J. Dupačová and R. J.-B. Wets, *Asymptotic behaviour of statistical estimators and of optimal solutions of stochastic optimization problems*, Ann. Statist., 16 (1988), pp. 1517–1549.

[5] Y. M. Ermoliev and V. I. Norkin, *Normalized convergence in stochastic optimization*, Ann. Oper. Res., 30 (1991), pp. 187–198.

[6] P. Kall, *Stochastic linear programming*, Springer-Verlag, Berlin, 1976.

[7] ———, *On approximations and stability in stochastic programming*, in Parametric Optimization and Related Topics, J. Guddat, H. Th. Jongen, B. Kummer, and F. Nožička, eds., Akademie Verlag, Berlin, 1987, pp. 387–407.

[8] Y. M. Kaniovski, A. J. King, and R. J.-B. Wets, *Probabilistic bounds (via large deviations) for the solutions of stochastic programming problems*, Ann. Oper. Res., 56 (1995), pp. 189–208.

[9] J. Kiefer, *On large deviations of the empiric D.F. of vector chance variables and a law of the iterated logarithm*, Pacific J. Math., 11 (1961), pp. 649–660.

[10] J. Kiefer and J. Wolfowitz, *On the deviations of the empiric distribution function of vector chance variables*, Trans. Amer. Math. Soc., 87 (1958), pp. 173–186.

[11] A. J. King and R. T. Rockafellar, *Asymptotic theory for solutions in statistical estimation and stochastic programming*, Math. Oper. Res., 18 (1993), pp. 148–162.

[12] A. J. King and R. J.-B. Wets, *Epi-consistency of convex stochastic programs*, Stochastics Stochastics Rep., 34 (1991), pp. 83–92.

[13] D. Klatte, *Beiträge zur Stabilitätsanalyse nichtlinearer Optimierungsprobleme*, Dissertation B, Humboldt-Universität Berlin, Sektion Mathematik, Berlin, 1984.

[14] D. Klatte and G. Thiere, *Error bounds for solutions of linear equations and inequalities*, Math. Methods Oper. Res., 41 (1995), pp. 191–214.

[15] B. Kummer, *Generalized equations: Solvability and regularity*, Math. Programming Stud., 21 (1984), pp. 199–212.

[16] F. NOŽIČKA, J. GUDDAT, H. HOLLATZ, AND B. BANK, *Theorie der linearen parametrischen Optimierung*, Akademie-Verlag, Berlin, 1974.
[17] S. T. RACHEV, *Probability Metrics and the Stability of Stochastic Models*, Wiley, Chichester, UK, 1991.
[18] H. RICHTER, *Das Gesetz vom iterierten Logarithmus für empirische Verteilungsfunktionen im $\mathbb{R}^k$*, Manuscripta Math., 11 (1974), pp. 291–303.
[19] S. M. ROBINSON AND R. J.-B. WETS, *Stability in two-stage stochastic programming*, SIAM J. Control Optim., 25 (1987), pp. 1409–1416.
[20] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[21] ———, *Integral functionals, normal integrands and measurable selections*, in Nonlinear Operators and the Calculus of Variations, G. P. Gossez et al., eds., Lecture Notes in Math., Vol. 543, Springer-Verlag, New York, 1976, pp. 157–207.
[22] W. RÖMISCH AND R. SCHULTZ, *Stability analysis for stochastic programs*, Ann. Oper. Res., 30 (1991), pp. 241–266.
[23] ———, *Stability of solutions for stochastic programs with complete recourse*, Math. Oper. Res., 18 (1993), pp. 590–609.
[24] R. SCHULTZ, *Strong convexity in stochastic programs with complete recourse*, J. Comput. Appl. Math., 56 (1994), pp. 3–22.
[25] R. J. SERFLING, *Approximation Theorems of Mathematical Statistics*, Wiley, New York, 1980.
[26] A. SHAPIRO, *Asymptotic properties of statistical estimators in stochastic programming*, Ann. Statist., 17 (1989), pp. 841–858.
[27] ———, *On differential stability in stochastic programming*, Math. Programming, 47 (1990), pp. 107–116.
[28] ———, *Asymptotic analysis of stochastic programs*, Ann. Oper. Res., 30 (1991), pp. 169–186.
[29] ———, *Quantitative stability in stochastic programming*, Math. Programming, 67 (1994), pp. 99–108.
[30] S. VOGEL, *Stability results for stochastic programming problems*, Optimization, 19 (1988), pp. 269–288.
[31] R. J.-B. WETS, *Stochastic programs with fixed recourse: The equivalent deterministic program*, SIAM Rev., 16 (1974), pp. 309–339.
[32] ———, *Stochastic Programming*, in Handbooks in Operations Research and Management Science, Vol. 1, Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North-Holland, Amsterdam, 1989, pp. 573–629.
[33] ———, *Constrained estimation: Consistency and asymptotics*, Appl. Stochastic Models Data Anal., 7 (1991), pp. 17–32.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 549–569, August 1996
© 1996 Society for Industrial and Applied Mathematics

ILL-POSEDNESS AND THE COMPLEXITY OF DECIDING EXISTENCE OF SOLUTIONS TO LINEAR PROGRAMS*

JORGE R. VERA†

Abstract. We discuss efficient algorithms for deciding existence of solutions to linear programs specified with approximate data. This is important in applications where only an approximation to the real data of the problem is available for computation, or where rounding errors prevent the use of exact numbers. The algorithms are efficient from the point of view of computation and needed data, requiring excessive computation and an excessively precise approximation only for nearly ill-posed instances. We illustrate how the proximity to ill-posedness measures the "conditioning" of the problem and plays an important role in the complexity analysis. This work is one step toward the understanding of ill-posedness in optimization problems and the development of a general complexity theory of problem solving with approximate data.

Key words. theory of linear programming, complexity of linear programming, conditioning

AMS subject classifications. 90C, 90C05, 90C60

1. Introduction.

In this paper we analyze problem solving with approximate data. This is not a new topic in applied mathematics or numerical analysis. It is the normal setup in scientific computation, where uncertainty in the data and rounding errors in the computer prevent us from using the actual values for the data. Renegar [8] considered the problem of deciding a certain property of a problem when only an approximation to the actual instance is known. He showed that there exist efficient algorithms for deciding existence of solutions to the linear programming problem $\max\{c^T x : Ax \leq b, x \geq 0\}$, as well as for approximating an optimal solution to the problem. The efficiency of these algorithms is understood from the computational point of view, as well as from the point of view of the precision with which we must know the data. For that problem, a certain collection of instances are identified as the "ill-posed" instances. By this we mean instances for which the problem cannot be solved relying only on approximate data. It turns out that the computational effort and the precision needed in the data are related to a measure of "intrinsic difficulty" of the problem. This condition measure is proportional to the reciprocal of the distance from the instance to the set of ill-posed instances. This is somehow analogous to known properties of the condition number of a matrix, as defined and studied in numerical linear algebra. (See, for example, Demmel [2, 3] for a treatment of conditioning in this case and in other problems in numerical analysis.)

Here we develop these ideas for the linear programming problems $\max\{c^T x : Ax \leq b\}$ and $\max\{c^T x : Ax = b, x \geq 0\}$. We show that for these formulations of linear programming there are efficient algorithms that will decide existence or nonexistence of solutions using only an approximation to the actual instance, as long as the instance is not ill posed. However, unlike the problem analyzed by Renegar [8], these decisions cannot be achieved efficiently using only the minimal possible knowledge of the instance. But the algorithms are efficient in a wider sense: they require only linearly more precision than the minimum. The problems analyzed here illustrate the following dilemma: if we want to make decisions with the minimal

* Received by the editors June 30, 1992; accepted for publication (in revised form) January 10, 1995. This research was supported by a graduate fellowship from the Research Division of IBM Corporation while the author was a graduate student at Cornell University.

† Department of Industrial Engineering, University of Chile, República 701, Santiago, Chile (jvera@dii.uchile.cl).

possible knowledge of the data, we might be required to use excessive computation. On the other hand, if we impose as a requirement the use of an efficient algorithm (in the sense of some computational complexity model), we might need to use more precision than the minimum. The central objective is to try to keep the balance between data efficiency and computational efficiency. We achieve this balance for the problems analyzed in this paper.

These results should help in the development of a computational complexity theory with approximate data, a theory where the computational effort required to solve a problem is bounded not by a function of the bit size of the instance (as in the classical bit complexity theory based on the Turing machine model of computer), but rather by a function of the "intrinsic difficulty" of the problem, a measure of ill-posedness of the instance. This more or less follows the suggestion of Smale [10].

The setup in this work is similar to the one in Renegar [8]. We want to decide whether the linear programs $\max\{c^T x : Ax \leq b\}$ and $\max\{c^T x : Ax = b, x \geq 0\}$ have solutions or not, but instead of the actual instance $\alpha = (A, b, c) \in \mathbf{R}^{m \times n+m+n}$, we are given an approximation $\bar{\alpha} = (\bar{A}, \bar{b}, \bar{c})$ such that $\|\alpha - \bar{\alpha}\|_\infty \leq \delta$, where $\delta > 0$. We further assume that $(\bar{A}, \bar{b}, \bar{c})$ and δ are given in rational numbers. In this way, our knowledge about the instance is represented only by the ball $B_\infty(\bar{\alpha}, \delta)$, which is actually an interval in the data space. The question reduces to deciding whether or not all instances in the ball have solutions.

The algorithms we show in this work are "fully efficient," as defined by Renegar [8]. This means that they are computationally efficient and data efficient. Computational efficiency is understood in the sense of the classical bit complexity theory: the algorithms run in time bounded by a polynomial in the bit length of the rational input $(\bar{\alpha}, \delta)$. Data efficiency means that the algorithm is guaranteed to identify the property (in this case existence or nonexistence of solutions) whenever $\delta \leq C(\alpha)^{-E}/p(m, n)$, where $C(d) = \|\alpha\|_\infty/\rho$, p is a polynomial, E is an integer independent of α, and ρ is the distance in the ∞-norm from α to the set of ill-posed instances. $C(\alpha)$ is Renegar's condition measure. We can think of $\log C(\alpha)$ as the minimal precision, in number of digits, needed by any algorithm to decide the problem. This interpretation follows from the fact that if $\delta > \rho$, $B_\infty(\bar{\alpha}, \delta)$ might contain instances with different properties from those of α, while if $\delta < \rho/2$, for any approximate data $(\bar{\alpha}, \delta)$ we will have $B_\infty(\bar{\alpha}, \delta) \subset B_\infty(\alpha, \rho)$, and hence all instances in $B_\infty(\bar{\alpha}, \delta)$ will have the same properties as α.

Once we have been able to decide that the instance has solutions, it makes sense to approximate one of these. That is the subject of an upcoming paper, where we consider approximating optimal solutions and the optimal value of linear programs with approximate data. Extensive coverage of these topics can also be found in Vera [12], where similar questions for quadratic programming are analyzed.

This paper is divided into several sections. In the first section some notation is defined and the main results are stated. The next sections include the proofs and technicalities.

2. Preliminaries and statement of the main results.

2.1. Notational conventions. Most of the mathematics used in this work is linear algebra and we use the standard notation used there. In particular, we will denote vectors with small letters and matrices with capital letters. Norms are used throughout. If v is a vector, $\|v\|_p$ will denote, as usual, the p-norm of v, and if A is a matrix, $\|A\|_p$ is the p-norm of A (operator norm). In particular, $\|A\|_2$ is

the spectral norm of A. If we have a combination of matrices and vectors in an n-tuple, we will refer to the ∞-norm of the whole vector of all numbers. For example, when we write $\|(A,b)\|_\infty$ we mean $\max\{|A_{ij}|, |b_i|\}$. If $x \in \mathbf{R}^N$ and $C \subset \mathbf{R}^N$, then $dist_p(x,C) = \min\{\|x-y\|_p : y \in C\}$. Also, in most places $\|A\|_\infty$ means $\max\{|A_{ij}|\}$ and not the ∞ operator norm of A, as will be clear from context. In a few places we will also use the Frobenius norm of a matrix, $\|A\|_F = (\sum a_{ij}^2)^{1/2}$. $B_p(x,\delta)$ will always refer to the closed ball in the p-norm centered at x with radius δ. Also, if x is a vector, $|x|$ will denote the vector whose components are the absolute value of the components of x. If $x \in \mathbf{R}^N$, $sgn(x)$ is the vector in $\mathbf{R}^N$ whose ith component is given by $sgn(x)_i = 1$ if $x_i > 0$, $sgn(x)_i = -1$ if $x_i < 0$, and $sgn(x)_i = 0$ if $x_i = 0$. Regarding vector comparisons, if $x \in \mathbf{R}^n$, then $x >> 0$ means $x_i > 0 \; \forall i = 1, \ldots, n$, and $x > 0$ means $x \geq 0$ but $x \neq 0$. Finally, e_m will denote the m-dimensional vector whose components are all equal to one, unless stated otherwise.

2.2. The main results. We give here an overview of the main results in the paper. We will concentrate on two of the standard formulations of linear programming:

$$\begin{aligned} P_1) \quad & \max\{c^T x : Ax \leq b\}, \\ P_2) \quad & \max\{c^T x : Ax = b, x \geq 0\}. \end{aligned}$$

In both cases, $A \in \mathbf{R}^{m \times n}, b \in \mathbf{R}^m, c \in \mathbf{R}^n$.

The data (A,b,c) will be considered an element of the data space $\mathbf{R}^{m \times n+m+n}$. We will assume that the data has been normalized so that $\|(A,b,c)\|_\infty = 1$. One advantage of this normalization is that distances to ill-posedness are always less than or equal to one. This follows by noticing that if $\delta > 1$, then the zero vector is in the interior of $B_\infty((A,b,c),\delta)$, and hence any problem in the data space, via an appropriate scaling, will be included in $B_\infty((A,b,c),\delta)$. With this assumption, the condition measure is simplified to $C(A,b,c) = 1/\rho$. We will further assume that the approximate data also satisfies $\|(\bar{A},\bar{b},\bar{c})\|_\infty \leq 1$. That is the case, for example, if the approximate data originates from a truncated normalized numerical expansion of the actual data.

For the two formulations described above we want to decide whether there is an optimal solution based only on the knowledge of approximate data. To be precise, we know only that $(A,b,c) \in \mathcal{D} = B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$, $0 < \delta < 1$.

Existence of a solution is equivalent to the problem being both primal and dual feasible. Let

$$\begin{aligned} \mathcal{F}_1^P &= \{(A,b,c) \; : \; \exists x \in \mathbf{R}^n \text{ such that } Ax \leq b\}, \\ \mathcal{F}_1^D &= \{(A,b,c) \; : \; \exists y \in \mathbf{R}^m \text{ such that } A^T y = c, y \geq 0\}; \end{aligned}$$

$\mathcal{F}_2^P, \mathcal{F}_2^D$ are defined in a similar way for the second formulation. Now let

$$\mathcal{F}_1 = \mathcal{F}_1^P \cap \mathcal{F}_1^D, \quad \mathcal{F}_2 = \mathcal{F}_2^P \cap \mathcal{F}_2^D$$

and

$$\begin{aligned} \mathcal{B}_1 &= (\text{int}\mathcal{F}_1 \cup \text{int}\mathcal{F}_1^C)^C, \\ \mathcal{B}_2 &= (\text{int}\mathcal{F}_2 \cup \text{int}\mathcal{F}_2^C)^C, \end{aligned}$$

where $(\cdot)^C$ denotes the complement of sets. Notice that if an instance $\alpha \in \mathcal{B}_1$, then for any $\delta > 0$ and $\bar{\alpha}$ such that $\alpha \in B_\infty(\bar{\alpha},\delta)$, the ball contains instances in $\mathcal{F}_1$ as well

as in $\mathcal{F}_1^C$. This means that no approximate data will suffice to decide whether or not α corresponds to a linear program with optimal solutions. A similar argument can be applied to $\mathcal{B}_2$. Therefore, define $\mathcal{B}_1$ and $\mathcal{B}_2$ to be the sets of ill-posed instances.

It follows that for the first formulation, if our uncertainty interval is such that $\mathcal{D} \subset \mathcal{F}_1$, then P_1 has an optimal solution. If $\mathcal{D} \subset \mathcal{F}_1^C$ no optimal solution exists: the problem is either infeasible or unbounded.

We are interested in developing efficient algorithms, from both the computational point of view and the data point of view, capable of deciding whether $\mathcal{D} \subset \mathcal{F}_1$ or $\mathcal{D} \subset \mathcal{F}_1^C$ (and similar relations for the second formulation). Since $\mathcal{D}$ is represented as an interval, the center and radius will be the input data for the algorithm. The algorithms will have the following structure:

INPUT : $(\bar{A}, \bar{b}, \bar{c})$, δ, rationals such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$.
OUTPUT : One of the following:
A : $\mathcal{D} \subset \mathcal{F}_1$ (or $\mathcal{D} \subset \mathcal{F}_2$),
B : $\mathcal{D} \subset \mathcal{F}_1^C$ (or $\mathcal{D} \subset \mathcal{F}_2^C$),
C : Cannot decide. Please provide a smaller δ.

The basic observation is that if the instance (A, b, c) is not ill posed, the algorithm will always answer either alternative whenever δ is small enough. The main results of this paper are as follows.

THEOREM 2.1. *Let $(A, b, c) \notin \mathcal{B}_1$. For problem P_1, there is a polynomial time algorithm which, given $(\bar{A}, \bar{b}, \bar{c}), \delta$, rational data, such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$, will either tell us that the problem has or does not have a solution, or will request more precision. The algorithm will always identify whether or not (A, b, c) has solutions when $\delta = \rho^3 / f_1(n, m)$ or less, where f_1 is a polynomial and $\rho = dist_\infty((A, b, c), \mathcal{B}_1)$.*

And for the second formulation, the result is the following theorem.

THEOREM 2.2. *Let $(A, b, c) \notin \mathcal{B}_2$. For problem P_2, there is a polynomial time algorithm which, given $(\bar{A}, \bar{b}, \bar{c}), \delta$, rational data, such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$, will either tell us that the problem has or does not have a solution, or will request more precision. The algorithm will always identify whether or not (A, b, c) has solutions when $\delta = \rho^3 / f_2(n, m)$ or less, where f_2 is a polynomial and $\rho = dist_\infty((A, b, c), \mathcal{B}_2)$.*

The results of this paper will show fully efficient algorithms for deciding the existence of solutions for two of the standard formulations of linear programming. The algorithms are computationally efficient, as they run in polynomial time, and are also data efficient, as they do not require much more precision than the minimum needed by any algorithm. If we think of $\log(1/\delta)$ as the number of digits of accuracy with which we know the data, it follows from the theorems that the precision needed to answer the question of existence of solutions is dominated by $3\log(1/\rho)$, so we need at most three times more precision than the minimum in order to decide the problem; moreover, these decisions are achieved using computationally efficient algorithms.

3. Deciding feasibility of linear programs with approximate data. In this section we will present efficient algorithms that will allow us to decide whether $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1^P$ or $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$ (and the analogous relations for the second formulation). We develop two computable conditions depending on whether $(\bar{A}, \bar{b}, \bar{c}) \in \mathcal{F}_1^P$ or $(\bar{A}, \bar{b}, \bar{c}) \in (\mathcal{F}_1^P)^C$. Proposition 3.1 will give a necessary and sufficient condition for $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1^P$ while Proposition 3.2 will establish a similar result for $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$. We will see that the first condition can be easily checked using linear programming. The second condition, however, is harder to check because it involves a combinatorial structure. Instead we develop a sufficient condition that

will require more precision than the minimum but can be checked in time polynomially bounded in the length of the input data. Then in §3.2 we will consider how to decide whether $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^D$ or $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^D)^C$.

3.1. Deciding feasibility of the primal problem. Observe that by Farkas' lemma, feasibility of $Ax \leq b$ is equivalent to infeasibility of the system $A^T y = 0, b^T y < 0, y \geq 0$. Now, $A^T y = 0$ is always feasible, so we only need to know whether or not points satisfying $A^T y = 0$ are in $\{y \geq 0 \ : \ b^T y < 0\}$. Of course, we should do this for all instances $(A,b,c) \in B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$. We will show that it is possible to construct an "enclosure" of all points y such that $A^T y = 0$, for some instance $(A,b,c) \in B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$, and make a test for that set. We begin with a definition.

DEFINITION 3.1. *For $t \geq 0$ define the set*

$$S(t) = \{y \in \mathbf{R}^m \ : \ |\bar{A}^T y| \leq t e_n\}.$$

This set is the enclosure mentioned above. The following two propositions give necessary and sufficient conditions for all problems in the ball to be feasible or infeasible.

PROPOSITION 3.1. *Let $(\bar{A},\bar{b},\bar{c}) \in \mathcal{F}_1^P$ and let $\delta > 0$. Then the following holds:*

$$B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^P \iff S(\delta) \cap \{y \geq 0 : e_m^T y = 1, (\bar{b} - \delta e_m)^T y < 0\} = \emptyset.$$

Proof. Suppose that $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^P$. Then if $(A,b,c) \in B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$ the system $Ax \leq b$ is feasible, which, by Farkas' lemma, implies that the system $A^T y = 0, b^T y < 0, y \geq 0$ is infeasible. Now suppose that

$$S(\delta) \cap \{y \geq 0 \ : \ e_m^T y = 1, (\bar{b} - \delta e_m)^T y < 0\} \neq \emptyset$$

and assume $\bar{y}$ lies in the intersection. Let $d = \bar{A}^T \bar{y}$. Then $\|d\|_\infty \leq \delta$. Let $\Delta A = -e_m d^T$. Then $(\bar{A}+\Delta A)^T \bar{y} = 0$ and $(\bar{b}-\delta e_m)^T \bar{y} < 0$. So, we have found $\tilde{A} = (\bar{A}+\Delta A)$, $\tilde{b} = (\bar{b} - \delta e_m)$ such that $\tilde{A}^T \bar{y} = 0, \tilde{b}^T \bar{y} < 0, \bar{y} \geq 0$, which implies that $\tilde{A}x \leq \tilde{b}$ is infeasible. However, this is not possible because $\|(\bar{A},\bar{b}) - (\tilde{A},\tilde{b})\|_\infty \leq \delta$. We conclude that

$$S(\delta) \cap \{y \geq 0 \ : \ e^T y = 1, (\bar{b} - \delta e_m)^T y < 0\} = \emptyset. \tag{1}$$

Conversely, suppose that (1) holds and there exists an instance (A,b,c) in $B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$ such that $Ax \leq b$ is infeasible or, equivalently, that $A^T y = 0, b^T y < 0, y \geq 0$ is feasible. Then we can assume that there exists $\bar{y} \geq 0, e_m^T \bar{y} = 1$ such that $A^T \bar{y} = 0, b^T \bar{y} < 0$. Now, if $A = \bar{A} + \Delta A$, $b = \bar{b} + \Delta b$ we have that

$$A^T \bar{y} = 0 \Longrightarrow \bar{A}^T \bar{y} = -(\Delta A)^T \bar{y} \Longrightarrow |\bar{A}^T \bar{y}| \leq \delta e_m$$

and

$$(\bar{b} - \delta e_m)^T \bar{y} \leq \bar{b}^T \bar{y} < 0.$$

We have that

$$S(\delta) \cap \{y \geq 0 \ : \ e_m^T y = 1, (\bar{b} - \delta e_m)^T y < 0\} \neq \emptyset,$$

which is a contradiction. This proves the proposition. □

PROPOSITION 3.2. *Let* $(\bar{A}, \bar{b}, \bar{c}) \in (\mathcal{F}_1^P)^C$ *and* $\delta > 0$. *Then the following holds:*

$$B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C \iff \left[\begin{array}{l} \bar{A} \text{ has full column rank, } e_m \text{ is not in the column} \\ \text{space of } \bar{A}, \text{ and every minimal face of } S(\delta) \cap \\ \{y : e_m^T y = 1\} \text{ has a point in } \{y \in \mathbf{R}^m : y \geq 0, \\ (\bar{b} + \delta e_m)^T y < 0\}. \end{array} \right.$$

Proof. We will begin with an explicit representation for the minimal faces. Let $F_1, \ldots, F_p$ denote the minimal faces of $S(\delta) \cap \{y : e_m^T y = 1\}$. Notice that if $\bar{A}$ has full column rank and e_m is not in the column space of $\bar{A}$, then every F_i has the form

$$F_i = \{y : \bar{A}^T y = d^i, e_m^T y = 1\},\ d_j^i \in \{-\delta, \delta\},\ j = 1, \ldots, n, i = 1, \ldots, p. \tag{2}$$

Also notice that all combinations of values for d_j^i will generate all possible minimal faces (there is an exponential number of them). Now let $\Delta A^i = -e_m(d^i)^T$. If $y \in F_i$ it follows that $(\bar{A} + \Delta A^i)^T y = \bar{A}^T y + (\Delta A^i)^T y = d^i - d^i e_m^T y = 0$. Conversely, if $(\bar{A} + \Delta A^i)^T y = 0, e_m^T y = 1$ then $\bar{A}^T y = -(\Delta A^i)^T y$, which implies that $\bar{A}^T y = d^i$. It follows that

$$F_i = \{y : (\bar{A} + \Delta A^i)^T y = 0, e_m^T y = 1\},\ \Delta A^i = -e_m(d^i)^T. \tag{3}$$

The previous relation gives a characterization of the minimal faces under the assumption that $\bar{A}$ has full rank and e_m is not in the column space of $\bar{A}$. We will use this in what follows.

Now suppose that $(\bar{A}, \bar{b}, \bar{c}) \in \text{int}(\mathcal{F}_1^P)^C$. Let us first argue that $\bar{A}$ must have full column rank. Suppose it is not the case and $\bar{A}$ does not have full column rank. Then there exists $u \neq 0$ such that $\bar{A}u = 0$. Suppose, in particular, that the ith component of u is nonzero. Then

$$\bar{A}u = \bar{A}_i u_i + \sum_{j \neq i} \bar{A}_j u_j = 0.$$

So, we can make an arbitrarily small perturbation ΔA_i in A_i, where $\bar{A}_i$ denotes the ith column of $\bar{A}$, to obtain

$$(\bar{A}_i + \Delta A_i)u_i + \sum_{j \neq i} \bar{A}_j u_j << 0.$$

This implies that $(\bar{A} + \Delta A)x \leq \bar{b}$ is feasible, which is a contradiction. We conclude that $\bar{A}$ must have full column rank. Also notice that e_m cannot be in the column space of $\bar{A}$, for otherwise there will exist $u \neq 0$ such that $-e_m = \bar{A}u$, implying that the system $\bar{A}x \leq \bar{b}$ is feasible, which is a contradiction.

Now let us go back to the body of the proof. Suppose first that $B((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$. Then we have immediately that $(\bar{A}, \bar{b}, \bar{c}) \in \text{int}(\mathcal{F}_1^P)^C$, $\bar{A}$ has full rank, and e_m is not in the column space of $\bar{A}$. Hence, representation (3) is valid. Now we must prove that every minimal face of $S(\delta) \cap \{y : e_m^T y = 1\}$ has a point y such that $y \geq 0, (\bar{b} + \delta e_m)^T y < 0$. Suppose it is not the case and there exists a face $F_i = \{y : (\bar{A} + \Delta A^i)^T y = 0, e^T y = 1\}$ such that $F_i \cap \{y : y \geq 0, (\bar{b} + \delta e_m)^T y < 0\} = \emptyset$. However, $\|\Delta A^i\|_\infty \leq \delta$, which implies that $(\bar{A} + \Delta A^i, \bar{b} + \delta e_m, \bar{c}) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$. Then $(\bar{A} + \Delta A^i)x \leq (\bar{b} + \delta e_m)$ is infeasible, which implies that the system $(\bar{A} + \Delta A^i)^T y = 0, (\bar{b} + \delta e_m)^T y < 0, y \geq 0$ is feasible, which is a contradiction. So, the direct implication is true.

Reciprocally, suppose that $\bar{A}$ has full rank, e_m is not in the column space of $\bar{A}$, and every minimal face of $S(\delta) \cap \{y : e_m^T y = 1\}$ has a point in $\{y : y \geq 0, e_m^T y = 1\}$. Then, by hypothesis, for each $i = 1, \ldots, p$ there exists $y \geq 0$ such that $y \in F_i, (\bar{b} + \delta e_m)^T y < 0$. Since $\bar{A}$ has full rank and e_m is not in the column space of $\bar{A}$, we can use representation (3) and conclude that the systems

$$(\bar{A} + \Delta A^i)^T y = 0,\ (\bar{b} + \delta e_m)^T y < 0,\ y \geq 0,\ i = 1, \ldots, p$$

are all feasible, which implies that $(\bar{A}+\Delta A^i)x \leq \bar{b}+\delta e_m$ is infeasible for all $i = 1, \ldots, p$.

Now suppose that $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$ is not a subset of $(\mathcal{F}_1^P)^C$ and there exists a problem instance (A, b, c) such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$ and $Ax \leq b$ is feasible. Then it follows that $Ax \leq b+\delta e_m$ is feasible. Let $\bar{x}$ be such a feasible point and define

$$(\Delta A)_j = \begin{cases} -\delta e_m, & \bar{x}_j \geq 0, \\ \delta e_m, & \bar{x}_j < 0. \end{cases}$$

Then

$$(\bar{A} + \Delta A)\bar{x} = \bar{A}\bar{x} + \sum_{j=1}^{m} (\Delta A)_j \bar{x}_j \leq \bar{b} + \delta e_m.$$

But observe that ΔA has precisely the form of one of the ΔA^i. So, we have found i such that the system $(\bar{A} + \Delta A^i)x \leq \bar{b} + \delta e_m$ is feasible, which is a contradiction to our previous conclusions. This proves that $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$ and completes the proof of the proposition. □

These two propositions give necessary and sufficient *computable* conditions for deciding whether $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$ is contained in $\mathcal{F}_1^P$ or in $(\mathcal{F}_1^P)^C$. We can see immediately that the condition in Proposition 3.1 can be easily checked using linear programming. This will lead to our first polynomial time decision algorithm for this problem, as the following proposition shows.

PROPOSITION 3.3. *There exists a polynomial time algorithm which, given rational data* $(\bar{A}, \bar{b}, \bar{c}) \in \mathcal{F}_1^P$, $\delta > 0$, *will either tell us that* $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1^P$ *or will ask for a smaller* δ. *The algorithm will answer positively whenever* $\delta < \bar{\rho}$, *where* $\bar{\rho} = dist_\infty((\bar{A}, \bar{b}, \bar{c}), (\mathcal{F}_1^P)^C)$.

Proof. Consider the following linear programming problem:

$$C_1) \qquad \begin{array}{rcl} v = \min & & (\bar{b} - \delta e_m)^T y \\ \text{s.t.} & & \\ |\bar{A}^T y| & \leq & \delta e_n, \\ e_m^T y & = & 1, \\ y & \geq & 0. \end{array}$$

It is immediate to see that $S(\delta) \cap \{y : y \geq 0, e^T y = 1, (\bar{b} - \delta e_m)^T y < 0\} \neq \emptyset$ if and only if problem C_1 is feasible and $v < 0$. The validity of the following algorithm follows immediately from this observation.

FUNCTION CHECK-CONDITION1$(\bar{A}, \bar{b}, \bar{c}, \delta)$
Solve problem C_1;
If C_1 is infeasible **then**
  RETURN$(B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \in \mathcal{F}_1^P)$
else
  If $v \geq 0$ **then**

RETURN$(B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \in \mathcal{F}_1^P)$
else
RETURN(Please provide a smaller δ)
end
end

If $\bar{A},\bar{b},\bar{c},\delta$ are rational numbers, then this algorithm can be assumed to run in time bounded by a polynomial in the bit length of the input data. It also follows from Proposition 3.1 that $S(\delta) \cap \{y \geq 0 \ : \ e^T y = 1, (\bar{b} - \delta e_m)^T y < 0\} = \emptyset$ if and only if $\delta < \bar{\rho}$, and hence problem C_1 will be infeasible, or feasible with $v \geq 0$, if and only if $\delta < \bar{\rho}$. □

Observe that the result of the previous proposition tells us that we can check feasibility of a polyhedra of the form $Ax \leq b$ with the *minimal possible knowledge of the data*. It is not the case for checking infeasibility, as there is no clear way to check the condition in Proposition 3.2 efficiently. We will develop an alternative sufficient condition such that if $(\bar{A},\bar{b},\bar{c}) \in (\mathcal{F}_1^P)^C$, it will be satisfied for a small enough δ, telling us that $B((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^P)^C$. However, this δ will not be much smaller than the minimum required by any algorithm.

For this consider the following analytical center problem:

$$\max\{f(y,\lambda) \ : \ \bar{A}^T y = 0, e_m^T y = 1, \bar{b}^T y + \lambda = 0, y > 0, \lambda > 0\}, \tag{4}$$

where

$$f(y,\lambda) = \sum_{i=1}^{m} \log y_i + \log \lambda.$$

As is well known, this convex problem in a bounded region has a unique solution. Let $(\hat{y},\hat{\lambda})$ be the analytical center obtained. Let

$$\hat{\sigma} = \min\{\hat{y}_1,\ldots,\hat{y}_m,\hat{\lambda}/\|\bar{b}\|_2\}$$

and

$$H(\delta) = \{y \ : \ y \in S(\delta), y = \hat{y} + h, h = \bar{A}u, u \in \mathbf{R}^n\}.$$

Notice that $\hat{\sigma}$ is the radius of the maximal sphere centered at $\hat{y}$ and contained in $\{y \ : \ y \geq 0, \bar{b}^T y \leq 0\}$. $H(\delta)$ is an orthogonal section at $\hat{y}$ and contained in $S(\delta)$. Notice that by construction $H(\delta)$ intersects every minimal face of $S(\delta)$. This follows by noticing that for a given d^i defining a minimal face F_i of $S(\delta)$, there exists u^i such that $d^i = \bar{A}^T \bar{A} u^i$. So, taking $h^i = \bar{A}u^i$, $y = \hat{y} + h^i$ is in the face F_i.

The following lemma provides some useful relations between distances used in the following results.

LEMMA 3.1. *Let $A \in \mathbf{R}^{m\times n}$, $m \geq n$. Then*

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2 \ .$$

Moreover, if

$$\rho_{sing} = dist_2(A, \{B \in \mathbf{R}^{m\times n} \ : \ \text{rank}(B) < n\})$$

and

$$\rho_F = dist_F(A, \{B \in \mathbf{R}^{m\times n} \ : \ \text{rank}(B) < n\}),$$

we have

$$\rho_{sing} \leq \rho_F \leq \sqrt{n}\rho_{sing}.$$

Proof. The first relation is well known in linear algebra. To prove the second one, let A' be such that

$$\|A - A'\|_2 = \rho_{sing} = dist_2(A, \{B \in \mathbf{R}^{m\times n} \ : \ \text{rank}(B) < n\})$$

and A'' be such that

$$\|A - A''\|_F = \rho_F = dist_F(A, \{B \in \mathbf{R}^{m\times n} \ : \ \text{rank}(B) < n\}).$$

Then we have

$$\|A - A'\|_2 \leq \|A - A'\|_F \leq \sqrt{n}\|A - A'\|_2 \ .$$

Then $\rho_{sing} \leq \|A - A'\|_F \leq \sqrt{n}\rho_{sing}$. Now, $\rho_F \leq \|A - A'\|_F$, which implies that $\rho_F \leq \sqrt{n}\rho_{sing}$. Now,

$$\|A - A''\|_2 \leq \rho_F \leq \sqrt{n}\|A - A''\|_2$$

and this implies that $\rho_{sing} \leq \rho_F$, proving the result. □

PROPOSITION 3.4. *Let* $(\bar{A}, \bar{b}, \bar{c}) \in (\mathcal{F}_1^P)^C$ *and let* $\bar{\rho} = dist_\infty((\bar{A}, \bar{b}, \bar{c}), \mathcal{F}_1^P)$. *Then if* $\bar{A}$ *has full rank,* e_m *is not in the column space of* $\bar{A}$, *and* $\delta \leq \hat{\sigma}\bar{\rho}^2/(8n\sqrt{m})$, *we have*

$$H(\delta) \subset \{y \ : \ y \geq 0, (\bar{b} + \delta e_m)^T y < 0\}.$$

Proof. Let us define

$$\eta_i(\delta) = dist_2(\{y \ : \ \bar{A}^T y = 0\}, \{y \ : \ \bar{A}^T y = d^i\}),$$
$$\eta(\delta) = \max_i \eta_i(\delta).$$

Observe now that

$$\eta_i(\delta) = \min\{\|y - y'\|_2 \ : \ \bar{A}^T y' = d^i, \bar{A}^T y = 0\}.$$

This optimization problem is solved by vectors y, y', h such that $\|h\|_2 = \eta_i(\delta)$, $h = \bar{A}u, u \in \mathbf{R}^n, y' = y + h$. Now, $\bar{A}^T y' = \bar{A}^T y + \bar{A}^T h$ and hence $d^i = \bar{A}^T h = \bar{A}^T \bar{A}u$. So, $u = (\bar{A}^T \bar{A})^{-1} d^i$ and $h = \bar{A}(\bar{A}^T \bar{A})^{-1} d^i$. This implies that

$$\|h\|_2^2 = \|\bar{A}(\bar{A}^T \bar{A})^{-1} d^i\|_2^2 \leq \|(\bar{A}^T \bar{A})^{-1}\|_2 \|d^i\|_2^2 \ .$$

Now let

$$\rho_{sing} = dist_2(\bar{A}, \{A \in \mathbf{R}^{m\times n} \ : \ \text{rank}(A) < n\}).$$

Now, it is well known that ρ_{sing} equals the smallest singular value of $\bar{A}$ (see, for example, Golub and Van Loan [5]). Hence, ρ_{sing}^2 equals the smallest eigenvalue of $\bar{A}^T \bar{A}$. Since the eigenvalues of a regular square matrix are the reciprocals of the eigenvalues of its inverse it follows that $\|(\bar{A}^T \bar{A})^{-1}\|_2 = 1/\rho_{sing}^2$. Based on Lemma 3.1 we have that $\rho_F \leq \sqrt{n}\rho_{sing}$. But if A' is such that $\rho_F = \|\bar{A} - A'\|_F$, we have

$$\sqrt{n}\rho_{sing} \geq \|\bar{A} - A'\|_F \geq \|\bar{A} - A'\|_\infty = \|(\bar{A}, \bar{b}, \bar{c}) - (A', \bar{b}, \bar{c})\|_\infty \geq \bar{\rho}.$$

This implies that $\|(\bar{A}^T\bar{A})^{-1}\|_2 \leq n/\bar{\rho}^2$, and hence

$$\|h\|_2^2 \leq \|d^i\|_2^2 \left(\frac{n}{\bar{\rho}^2}\right).$$

It follows that

$$\eta_i(\delta) = \|h\|_2 \leq \|d^i\|_2 \left(\frac{\sqrt{n}}{\bar{\rho}}\right).$$

But since $\|d^i\|_2 \leq \sqrt{n}\|d^i\|_\infty = \sqrt{n}\delta$ we have

$$\eta(\delta) \leq \frac{n\delta}{\bar{\rho}} \cdot$$

Now, notice that if $\delta \leq \delta_1 = \hat{\sigma}\bar{\rho}/2n$ then $\eta(\delta) \leq \hat{\sigma}/2$ and $H(\delta) \subset B_2(\hat{y}, \hat{\sigma}/2)$. We conclude that in this case $H(\delta) \subset \{y : y \geq 0\}$.

Now we must prove that for a suitable δ, $H(\delta) \subset \{y : (\bar{b} + \delta e_m)^T y < 0\}$. For this, begin by observing that a given $y \in H(\delta)$ has the form $y = \hat{y} + h$ for some h such that $h = Au, u \in \mathbf{R}^n$. Now,

$$\begin{aligned}(\bar{b} + \delta e_m)^T(\hat{y} + h) &= \bar{b}^T\hat{y} + \delta e_m^T\hat{y} + (\bar{b} + \delta e_m)^T h \\ &= -\hat{\lambda} + \delta + (\bar{b} + \delta e_m)^T h.\end{aligned}$$

Then it suffices to prove that for an appropriate δ, $\delta + (\bar{b} + \delta e_m)^T h < \hat{\lambda}$. First, observe that $\hat{\lambda} \geq \hat{\sigma}\|\bar{b}\|_2$. Now, we will certainly have that $\bar{b} + e_m\|\bar{b}\|_\infty \geq 0$. This implies that the system $\bar{A}x \leq \bar{b} + e_m\|\bar{b}\|_\infty$ is always feasible and hence that $\bar{\rho} \leq \|\bar{b}\|_\infty \leq \|\bar{b}\|_2$. Based on this we conclude that $\hat{\lambda} \geq \hat{\sigma}\bar{\rho}$. Now,

$$\begin{aligned}\delta + (\bar{b} + \delta e_m)^T h &\leq \delta + \|\bar{b} + \delta e_m\|_2\|h\|_2 \\ &\leq \delta + 2\sqrt{m}\|h\|_2 \,.\end{aligned}$$

But we have already deduced that $\|h\|_2 \leq (n/\bar{\rho})\delta$. Using this in the above formula we have

$$\delta + (\bar{b} + \delta e_m)^T h \leq \delta + \left(\frac{2n\sqrt{m}}{\bar{\rho}}\right)\delta \leq \left(\frac{4n\sqrt{m}}{\bar{\rho}}\right)\delta \,.$$

Now suppose we take $\delta \leq \delta_2 = \hat{\sigma}\bar{\rho}^2/8n\sqrt{m}$. Then we have

$$\delta + (\bar{b} + \delta e_m)^T h \leq \frac{\hat{\sigma}\bar{\rho}}{2} \leq \frac{\hat{\lambda}}{2} < \hat{\lambda},$$

as we wanted to prove. Now, clearly $\delta_2 < \delta_1$, so to have everything simultaneously it suffices to take $\delta \leq \delta_2$. This proves the result. □

Now, as we pointed out, $H(\delta)$ intersects every minimal face of $S(\delta)$. Thus, we conclude that if $H(\delta)$ satisfies the condition in Proposition 3.4, then every minimal face of $S(\delta)$ has a point in $\{y : y \geq 0, (\bar{b}+\delta e_m)^T y < 0\}$. This does not imply, however, that every minimal face of $S(\delta) \cap \{y : e^T y = 1\}$ has a point in $\{y \in \mathbf{R}^m : y \geq 0, (\bar{b} + \delta e_m)^T y < 0\}$. So, we cannot immediately conclude that $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$. However, we have the following result.

PROPOSITION 3.5. *Suppose that for some δ we have that $H(\delta) \subset \{y : y \geq 0, (\bar{b} + \delta e_m)^T y < 0\}$. Then $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta') \subset (\mathcal{F}_1^P)^C$ for $\delta' = \delta/(m+1)$.*

Proof. Suppose that $H(\delta) \subset \{y : y \geq 0, (\bar{b} + \delta e_m)^T y < 0\}$. This means that for a given minimal face F_i of the form $\bar{A}^T y = d^i$ of $H(\delta)$ there exists y such that $y \geq 0, (\bar{b} + \delta e_m)^T y < 0$. Now, y does not have to satisfy $e_m^T y = 1$, but $\bar{y} = y/e_m^T y$ certainly does. Clearly we still have that $(\bar{b} + \delta e_m)^T \bar{y} < 0$. But $\bar{A}^T \bar{y} = d^i / e_m^T y$. Now, we know that $y = \hat{y} + h$ for some h such that $h = \bar{A}u, u \in \mathbf{R}^n$. Hence, $e_m^T y = 1 + e_m^T h$ and

$$\bar{A}^T \bar{y} = \bar{d}^i = \frac{d^i}{1 + e_m^T h} \cdot$$

The next step is to bound $|e_m^T h|$. Observe that by construction, if a point $\hat{y} + h \in H(\delta)$, then $\hat{y} - h \in H(\delta)$. By hypothesis $H(\delta) \subset \{y : y \geq 0, (\bar{b} + \delta e_m)^T y < 0\}$ and hence, for any h such that $\hat{y} + h \in H(\delta)$, we have that $h \geq -\hat{y}$ and $h \leq \hat{y}$. This implies that $\|h\|_\infty \leq \|\hat{y}\|_\infty \leq 1$ and hence $\|h\|_2 \leq \sqrt{m}$. With this we have that

$$|e_m^T h| \leq \|e_m\|_2 \|h\|_2 \leq \sqrt{m}\sqrt{m} \leq m.$$

From this we conclude that $\|\bar{d}^i\|_\infty \geq \delta/(m+1)$. It follows that if we take a point of the form $\mu\bar{y} + (1-\mu)\hat{y}$, for $\mu \in [0,1]$, this point is in $\{y : e_m^T y = 1\}$. Now observe that $\bar{A}^T(\mu\bar{y} + (1-\mu)\hat{y}) = \mu\bar{A}^T\bar{y}$. It follows that we can certainly find a suitable μ such that a point y' satisfies $\bar{A}^T y' = d^i/(m+1)$. So, every minimal face of $S(\delta')$ has a point in $\{y : y \geq 0, e_m^T y = 1, (\bar{b} + \delta e_m)^T y < 0\}$, where $\delta' = \delta/(m+1)$. Moreover, since $\delta' < \delta$, we have that $(\bar{b} + \delta' e_m)^T y' < 0$. This implies that $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta') \subset (\mathcal{F}_1^P)^C$, as we wanted to prove. □

Combining both results we reach the following conclusion.

PROPOSITION 3.6. *Let* $(\bar{A}, \bar{b}, \bar{c}) \in \text{int}(\mathcal{F}_1^P)^C$ *and let* $\bar{\rho} = dist_\infty((\bar{A}, \bar{b}, \bar{c}), \mathcal{F}_1^P)$. *Then if* $\bar{A}$ *has full rank* e_m *is not in the column space of* $\bar{A}$ *and*

$$\delta \leq \frac{\hat{\sigma}\bar{\rho}^2}{8n(m+1)\sqrt{m}} \cdot$$

We have that

$$H((m+1)\delta) \subset \{y : y \geq 0, (\bar{b} + \delta(m+1)e_m)^T y < 0\}$$

and hence every minimal face of $S(\delta)$ *has a point in* $\{y : y \geq 0, (\bar{b} + \delta e_m)^T y < 0\}$, *implying that* $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$.

The proposition thus implies that for δ small enough we can use an alternate characterization that will tell us when $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1^P$. To check this condition we are required to compute the analytical center of a polyhedron (actually an approximation to the analytical center, as we will see below), and to use linear programming to check whether $H((m+1)\delta) \subset \{y : y \geq 0, (\bar{b} + \delta(m+1)e_m)^T y < 0\}$. All this can be done efficiently.

It remains to find a relation between $\bar{\rho}$ and $\hat{\sigma}$.

PROPOSITION 3.7. *Let* $(\bar{A}, \bar{b}, \bar{c}) \in \text{int}(\mathcal{F}_1^P)^C$ *and let* $\bar{\rho} = dist_\infty((\bar{A}, \bar{b}, \bar{c}), \mathcal{F}_1^P)$. *Then*

$$\bar{\rho} \leq (2(m+2)\sqrt{m})\hat{\sigma}.$$

Proof. The proof of this uses a well-known property of the analytical center of a polyhedron, as given in the following lemma (see, for example, Renegar [9]).

LEMMA 3.2. *Let* $\mathcal{D} = \{x \in \mathbf{R}^n : Ax \leq b, Bx = d\}$ *be a bounded polyhedron, where* $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{p \times n}$, $b \in \mathbf{R}^m$, $d \in \mathbf{R}^p$. *Let* $f(x) = \sum \log(b_i - \alpha_i^T x)$,

where α_i is the ith row of A. Let $\hat{x}$ be the solution to $\max\{f(x) : x \in \mathcal{D}\}$. *Let* $\hat{\sigma} = \min_i\{dist_2(\hat{x}, \{x : \alpha_i^T \hat{x} = b_i\})\}$ *and let i_0 be the index where the minimum is attained. Then*

$$\mathcal{D} \subset \{x : \alpha_{i_0}^T x > b_{i_0} - (m+1)\hat{\sigma}\|\alpha_{i_0}\|_2\}$$

or, equivalently,

$$\mathcal{D} \cap \{x : \alpha_{i_0}^T x \leq b_{i_0} - (m+1)\hat{\sigma}\|\alpha_{i_0}\|_2\} = \emptyset. \qquad \square$$

Now, back to our problem, recall that $\hat{\sigma} = \min\{\hat{y}_1, \ldots, \hat{y}_m, \bar{b}^T\hat{y}/\|\bar{b}\|_2\}$. Throughout the proof we will assume, to simplify the notation, that e denotes the m-dimensional vector of ones and that e_i is the ith unit vector. We consider two cases.

Case 1. $\hat{\sigma} = \bar{b}^T\hat{y}/\|\bar{b}\|_2$. In this case, it follows from the lemma that

$$\{y \geq 0 : \bar{A}^T y = 0, e^T y = 1, \bar{b}^T y \leq 0\} \cap \{y : \bar{b}^T y \leq -(m+2)|\bar{b}^T\hat{y}|\} = \emptyset. \tag{5}$$

Let $\Delta b = (m+2)|\bar{b}^T\hat{y}|e$. Then it follows that

$$\{y : e^T y = 1, \bar{b}^T y \leq -(m+2)|\bar{b}^T\hat{y}|\} = \{y : e^T y = 1, (\bar{b} + \Delta b)^T y \leq 0\}.$$

So, from (5) we have

$$\{y \geq 0 : e^T y = 1, \bar{A}^T y = 0, (\bar{b} + \Delta b)^T y \leq 0\} = \emptyset$$

and hence $(\bar{A}, \bar{b} + \Delta b, \bar{c}) \in \mathcal{F}_1^P$. Now, since $\|\Delta b\|_\infty = (m+2)\hat{\sigma}\|\bar{b}\|_2$ it follows that $\bar{\rho} \leq \|\Delta b\|_\infty \leq (m+2)\sqrt{m}\hat{\sigma}$.

Case 2. Now suppose that $\hat{\sigma} = \hat{y}_{i_0}$ for some $i_0 \in \{1, \ldots, m\}$. Then

$$\{y \geq 0 : e^T y = 1, \bar{A}^T y = 0, \bar{b}^T y \leq 0\} \cap \{y : y_{i_0} \geq (m+2)\hat{\sigma}\} = \emptyset.$$

Assume first that $\hat{\sigma} < 1/(m+2)$ and let

$$\mathcal{S} = \{y : e^T y = 1, y \geq 0, y_{i_0} \geq (m+2)\hat{\sigma}\}.$$

Notice that $\mathcal{S}$ is a simplex. We will define the following affine transformation of $\mathcal{S}$ onto $\{y \geq 0 : e^T y = 1\}$. Let

$$y' = \frac{y - (m+2)\hat{\sigma}e_{i_0}}{1 - (m+2)\hat{\sigma}}, \tag{6}$$

where e_{i_0} denotes the i_0th unit vector. Then

$$\bar{A}^T y = 0 \Longleftrightarrow \bar{A}^T y' = -\frac{(m+2)\hat{\sigma}\bar{A}^T e_{i_0}}{1 - (m+2)\hat{\sigma}}$$

and

$$\bar{b}^T y \leq 0 \Longleftrightarrow \bar{b}^T y' \leq -\frac{(m+2)\hat{\sigma}\bar{b}^T e_{i_0}}{1 - (m+2)\hat{\sigma}}.$$

Notice also that $e^T y' = 1$. Now let

$$\Delta A = e\left(\frac{(m+2)\hat{\sigma}e_{i_0}^T\bar{A}}{1 - (m+2)\hat{\sigma}}\right), \quad \Delta b = \left(\frac{(m+2)\hat{\sigma}\bar{b}^T e_{i_0}}{1 - (m+2)\hat{\sigma}}\right)e.$$

Then it follows that

$$\left\{y' \ : \ e^T y' = 1, \bar{A}^T y' = -\frac{(m+2)\hat{\sigma} e_{i_0}^T \bar{A}}{1-(m+2)\hat{\sigma}}\right\} = \{y' \ : \ e^T y' = 1, (\bar{A}+\Delta A)^T y' = 0\}$$

and

$$\left\{y' \ : \ e^T y' = 1, \bar{b}^T y' \leq -\frac{(m+2)\hat{\sigma} e_{i_0}^T \bar{b}}{1-(m+2)\hat{\sigma}}\right\} = \{y' \ : \ e^T y' = 1, (\bar{b}+\Delta b)^T y' \leq 0\}.$$

So,

$$\{y' \ : \ e^T y' = 1, (\bar{A}+\Delta A)^T y' = 0\} \cap \{y' \ : \ y' \geq 0, (\bar{b}+\Delta b)^T y' \leq 0\} = \emptyset.$$

Hence, $(\bar{A}+\Delta A, \bar{b}+\Delta b) \in \mathcal{F}_1^P$. Now observe that

$$\|\Delta A\|_\infty \leq \frac{(m+2)\hat{\sigma}}{1-(m+2)\hat{\sigma}}, \ \|\Delta b\|_\infty \leq \frac{(m+2)\hat{\sigma}}{1-(m+2)\hat{\sigma}}.$$

This implies that $\bar{\rho} \leq (m+2)\hat{\sigma}/(1-(m+2)\hat{\sigma})$ and hence $(m+2)\hat{\sigma} \geq \bar{\rho}/(1+\bar{\rho}) \geq \bar{\rho}/2$. It follows then that $\bar{\rho} \leq 2(m+2)\hat{\sigma}$.

If $\hat{\sigma} \geq 1/(m+2)$, then since $\bar{\rho} \leq 1$, the inequality $\bar{\rho} \leq 2(m+2)\hat{\sigma}$ follows immediately. Combining both cases we obtain the desired result. □

PROPOSITION 3.8. *If* $\delta \leq \bar{\rho}^3/(16(m+2)^2 mn)$ *then*

$$H((m+1)\delta) \subset \{y \ : \ y \geq 0, (\bar{b}+(m+1)\delta e_m)^T y < 0\}$$

and thus $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$.

Proof. It follows by simply combining the bound from Propositions 3.6 and 3.7 and doing some algebraic simplifications. □

Now, the construction of $H(\delta)$ requires the computation of an analytical center. We cannot compute that center exactly, but we will argue that an approximation within a fixed relative error from the true center suffices. This approximation can be obtained in polynomial time by any standard centralization technique based on a logarithmic barrier function.

PROPOSITION 3.9. *Let* $(\hat{y}, \hat{\lambda})$ *be the analytical center as defined in* (4) *and let* $(\bar{y}, \bar{\lambda})$ *be such that* $|\hat{y}_i - \bar{y}_i| \leq 0.5|\hat{y}_i|$ *for all* $i = 1, \ldots, m$, $|\hat{\lambda} - \bar{\lambda}| \leq 0.5|\hat{\lambda}|$. *Let*

$$\bar{H}(\delta) = \{y \in S(\delta) \ : \ y = \bar{y} + h, h = \bar{A}u\}.$$

Then if $\delta \leq \bar{\rho}^3/(32(m+2)^2 mn)$ *we have* $\bar{H}((m+1)\delta) \subset \{y \geq 0 \ : \ (\bar{b}+(m+1)\delta e_m)^T y < 0\}$.

Proof. From the hypothesis we have that $0.5\hat{y}_i \leq \bar{y}_i \leq 1.5\hat{y}_i$ for $i = 1, \ldots, m$ and $0.5\hat{\lambda} \leq \bar{\lambda} \leq 1.5\hat{\lambda}$. Let $\bar{\sigma} = \min\{\bar{y}_1, \ldots, \bar{y}_m, \bar{\lambda}/\|\bar{b}\|_2\}$. Then it follows that $0.5\hat{\sigma} \leq \bar{\sigma} \leq 1.5\hat{\sigma}$. Now, from Proposition 3.7 we have that $\bar{\rho} \leq 2(m+2)\sqrt{m}\hat{\sigma} \leq 4(m+2)\sqrt{m}\bar{\sigma}$. It is easy to see that the arguments in Propositions 3.4 and 3.6 do not depend on the fact that $\hat{y}$ is the analytical center of some polyhedra. This means that the same estimate for δ is still valid, but with $\bar{\sigma}$ instead of $\hat{\sigma}$. That is, it suffices to have $\delta \leq \bar{\sigma}\bar{\rho}^2/8n\sqrt{m}(m+1)$. Now, using the relation between $\bar{\sigma}$ and $\bar{\rho}$ we conclude that it suffices to have $\delta \leq \bar{\rho}^3/(32(m+2)^2 mn)$, as proposed. □

It is well known that any interior point algorithm, similar to the ones used in linear programming, can be used to obtain such an approximate analytical center in

polynomial time (see, for example, Renegar [9] or Vaidya [11]). This will allow us to prove the following.

PROPOSITION 3.10. *There exists a polynomial time algorithm which, given* $(\bar{A}, \bar{b}, \bar{c})$ *in* $(\mathcal{F}_1^P)^C$, $\delta > 0$ *rationals, will either tell us that* $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$ *or will request a smaller* δ. *The algorithm will answer positively whenever* $\delta < \bar{\rho}^3/(32(m+2)^2mn)$, *where* $\bar{\rho} = dist_\infty((\bar{A}, \bar{b}, \bar{c}), \mathcal{F}_1^P)$.

Proof. We only need to specify the algorithm. Let us assume that we have a polynomial time algorithm APPROX-CENTER1$(\bar{A}, \bar{b}, \bar{c}, \delta)$ which computes an approximate analytical center $(\bar{y}, \bar{\lambda})$ such that $|\hat{y}_i - \bar{y}_i| \leq 0.5|\hat{y}_i|$ for all $i = 1, \ldots, m$, $|\hat{\lambda} - \bar{\lambda}| \leq 0.5|\hat{\lambda}|$. As we pointed out, this algorithm can be based on any logarithmic barrier method for linear programming. The algorithm is as follows.

FUNCTION CHECK-CONDITION2$(\bar{A}, \bar{b}, \bar{c}, \delta)$
If rank$(\bar{A}) < n$ **or** $\exists u \in \mathbf{R}^n : e_m = \bar{A}u$ **then**
  RETURN(Please provide a smaller δ)
else
  $\bar{y}$ =APPROX-CENTER1$(\bar{A}, \bar{b}, \bar{c}, \delta)$
  construct $\bar{H}((m+1)\delta)$.
  If $\bar{H}((m+1)\delta) \subset \{y \geq 0 : (\bar{b} + (m+1)\delta e_m)^T y \leq 0\}$ **then**
    RETURN$(B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C)$
  else
    RETURN(Please provide a smaller δ)
  end
end

The running time of this algorithm is composed of the time needed to find the approximate analytical center, the time needed to evaluate whether $\bar{H}((m+1)\delta) \subset \{y \geq 0 : (\bar{b} + (m+1)\delta e_m)^T y \leq 0\}$, the time needed to decide whether $\bar{A}$ has full rank, and the time needed to determine feasibility of the system $\bar{A}u = e_m$. We have already pointed out that the computation of the approximate analytical center can be carried out in polynomial time. The evaluation of the inclusion can be done using linear programming and that can also be done in polynomial time. The other two steps can also be done in polynomial time. So, the total running time is polynomial in the bit length of the input data. The bound on δ follows from the analysis of the previous proposition. □

The combination of Propositions 3.3 and 3.10 leads to the following.

PROPOSITION 3.11. *There exists a polynomial time algorithm which, given* $(\bar{A}, \bar{b}, \bar{c})$ *and* $\delta > 0$ *rationals, such that* $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$, *will stop with one of the following alternatives:*

A : $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1^P$ and hence $(A, b, c) \in \mathcal{F}_1^P$,
B : $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^P)^C$ and hence $(A, b, c) \in (\mathcal{F}_1^P)^C$,
C : Cannot decide. Please provide a smaller δ.

If $(A, b, c) \notin \mathcal{B}_1$ *then the algorithm will stop with alternative A or B whenever* $\delta \leq \rho^3/(64(m+2)^2mn)$, *where* $\rho = dist_\infty((A, b, c), \mathcal{B}_1)$.

Proof. All we have to do is put together the algorithms we have defined so far in Propositions 3.3 and 3.10. This gives the following procedure.

FUNCTION PRIMAL-FEAS$(\bar{A}, \bar{b}, \bar{c}, \delta)$
If $\bar{A}x \leq \bar{b}$ is feasible **then**
  CHECK-CONDITION1$((\bar{A}, \bar{b}, \bar{c}, \delta))$;

If $(B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^P)$ **then**
RETURN$(B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^P)$
else
RETURN(Please provide a smaller δ)
end
else
CHECK-CONDITION2$((\bar{A},\bar{b},\bar{c},\delta))$;
If $(B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^P)^C)$ **then**
RETURN$(B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^P)^C)$
else
RETURN(Please provide a smaller δ)
end
end

As we have pointed out, if $(\bar{A},\bar{b},\bar{c}) \in \mathcal{F}_1^P$ and $\bar{\rho} = dist_\infty((\bar{A},\bar{b},\bar{c}),(\mathcal{F}_1^P)^C)$, then whenever $\delta < \bar{\rho}$ function CHECK-CONDITION1 will answer positively and the procedure will stop. Otherwise, if $(\bar{A},\bar{b},\bar{c}) \in (\mathcal{F}_1^P)^C$ and $\bar{\rho} = dist_\infty((\bar{A},\bar{b},\bar{c}),\mathcal{F}_1^P)$, then whenever $\delta < \bar{\rho}^3/(32(m+2)^2mn)$ function CHECK-CONDITION2 will answer positively and the procedure will stop. If neither of these hold, the procedure will ask for a smaller δ. Now to prove the bound on δ, which is in terms of ρ and not $\bar{\rho}$, observe that we can apply the proposed algorithm *in real arithmetic* to the instance (A,b,c) and obtain the same conclusion regarding δ. That is, whenever $\delta \leq \bar{\delta}$, where $\bar{\delta} = \rho^3/(32(m+2)^2mn)$, the algorithm will stop with alternative A or B. Now suppose that $(\bar{A},\bar{b},\bar{c})$ and δ are such that $(A,b,c) \in B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$ and $\delta \leq \bar{\delta}/2$. Then it follows immediately that $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset B_\infty((A,b,c),\bar{\delta})$, and hence the algorithm, when used in rational arithmetic on the instance $(\bar{A},\bar{b},\bar{c})$, will stop with alternative A or B. As expected, if the instance is ill posed, no matter how small δ is, the algorithm will never stop with alternative A or B. Regarding the running time, we have already argued that both subroutines run in time polynomial in the bit length of the input, so this procedure does so also. □

3.2. Deciding feasibility of the dual problem. Now we will show how to decide whether $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^D$ or $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^D)^C$. As we will see, we have already developed most of the concepts needed to solve this problem. Observe that if $A^Ty = c, y \geq 0$ is feasible, and then $A^Ty - \lambda c = 0$, $(y,\lambda) \geq 0$ is also feasible. This suggests considering the problem $A^Ty = 0, y \geq 0$, which can be analyzed similarly to the problem in the previous section, but we must first establish some equivalences between both formulations. Let us define

$$\mathcal{H} = \{A \in \mathbf{R}^{m\times n} \ : \ \exists y \geq 0 \ : \ A^Ty = 0, e_m^Ty = 1\}.$$

PROPOSITION 3.12. *$A \in \text{int}\mathcal{H} \iff$ A has full rank and there exists $y >> 0$ such that $A^Ty = 0$.*

Proof. Suppose first that $A \in \text{int}\mathcal{H}$. If A does not have full rank, then there exists $u \neq 0$ such that $Au = 0$. Let $u_j \neq 0$. Then we can make an arbitrarily small perturbation ΔA to A and obtain $(A+\Delta A)u << 0$. So, $(A+\Delta A)u \leq -e_m$ is feasible and hence $(A+\Delta A)^Ty = 0, e_m^Ty > 0, y \geq 0$ is infeasible for some arbitrarily small ΔA. However, this contradicts the assumption that $A \in \text{int}\mathcal{H}$. We conclude that A must have full rank. Now suppose that for all y such that $A^Ty = 0$, one component of y is less than or equal to zero. Then the system $A^Ty = 0, y \geq e_m$ is infeasible, implying that the system $Ax \leq 0, Ax \neq 0$ is feasible. Since x must be nonzero, we conclude that

we can take an arbitrarily small perturbation of A, ΔA and obtain the feasibility of the system $(A+\Delta A)x << 0$, implying that the system $(A+\Delta A)x \leq -e_m$ is feasible. But this means that the system $(A+\Delta A)^T y = 0, e_m^T y = 1, y \geq 0$ is infeasible for some arbitrarily small ΔA. However, this contradicts the assumption that $A \in \text{int}\mathcal{H}$.

Conversely, suppose that A has full rank and there exists $y >> 0$ such that $A^T y = 0$. Then it follows immediately that there exists y' such that $A^T y' = 0, e_m^T y' = 1, y' >> 0$. Since A has full column rank, we can assume that it can be decomposed in the form $A^T = [B|N]$, where B is a square regular matrix. Let y be partitioned as $y = (y_B, y_N)$. Then $y_B = -B^{-1}Ny_N >> 0, y_N >> 0$. Because of the continuity of the inverse of a matrix and since B is regular, it follows that for any $(\Delta A)^T = [\Delta B|\Delta N]$ arbitrary but small enough, we will still be able to find $y >> 0$ satisfiying $(A+\Delta A)^T y = 0, e_m^T y = 1, y >> 0$. So, $A \in \text{int}\mathcal{H}$. □

For the next proposition observe that $\mathcal{H}$ is now considered to be a set of matrices in $\mathbf{R}^{(m+1)\times n}$.

PROPOSITION 3.13. $(A,b,c) \in \text{int}(\mathcal{F}_1^D) \iff [A^T|-c]^T \in \text{int}\mathcal{H}$.

Proof. Suppose first that $(A,b,c) \in \text{int}(\mathcal{F}_1^D)$. Then there exists $y \geq 0$ such that $A^T y = c$. First we establish that A must have full rank. If not, there exists $u \neq 0$ such that $Au = 0$, and hence $u^T A^T y = u^T c = 0$. Then for some arbitrarily small $(\Delta A, \Delta c)$, $(A+\Delta A)u \leq 0$, $(c+\Delta c)^T u > 0$. This implies that (A,b,c) cannot be in the interior of $\mathcal{F}_1^D$, which is a contradiction. Now we show that there exists $\bar{y} >> 0$ such that $A^T\bar{y} = c$. If not, the system $A^T y = c, y \geq \lambda e_m$ is infeasible for all $\lambda > 0$, implying that for every $\lambda > 0$ there exists $(u,v) \neq 0$ such that $Au - v = 0$, $c^T u - \lambda e_m^T v < 0$, $v \geq 0$. Then the system $Au \leq 0, c^T u > -\lambda e^T v, v \geq 0$ is feasible for all $\lambda > 0$. Taking the limit when $\lambda \longrightarrow 0$, we conclude that the system $Au \leq 0, c^T u = 0$ is feasible, implying that (A,b,c) cannot be in the interior of $\mathcal{F}_1^D$, which is a contradiction. So, taking $\bar{y} >> 0$ and $\bar{\lambda} = 1$ we see that $A^T\bar{y} - \bar{\lambda}c = 0$. Since A has full rank, $[A^T|-c]^T$ also has full rank. By the previous proposition we conclude that $[A^T|-c]^T \in \text{int}\mathcal{H}$.

Consider now the converse statement and assume that $[A^T|-c]^T \in \text{int}\mathcal{H}$. By the previous proposition it follows that $[A^T|-c]^T$ has full rank and there exists $(\bar{y},\bar{\lambda}) >> 0$ such that $A^T\bar{y} - \bar{\lambda}c = 0$. It follows that $(A,b,c) \in \mathcal{F}_1^D$. By a continuity argument similar to the one used in the proof of Proposition 3.12, we can claim that for any $(\tilde{A},\tilde{b},\tilde{c})$ sufficiently close to (A,b,c) we will still have that there exists $(y,\lambda) >> 0$ such that $\tilde{A}^T y - \lambda\tilde{c} = 0$. Then $(A,b,c) \in \text{int}(\mathcal{F}_1^D)$, as we wanted to prove. □

PROPOSITION 3.14. $(A,b,c) \in \text{int}(\mathcal{F}_1^D)^C \iff [A^T|-c]^T \in \text{int}(\mathcal{H})^C$.

Proof. Begin by assuming $(A,b,c) \in \text{int}(\mathcal{F}_1^D)^C$. Then the system $A^T y = c, y \geq 0$ is infeasible. Now, $\mathcal{H}$ is a closed set and hence it suffices to prove that $[A^T|-c]^T \notin \mathcal{H}$. Suppose, on the contrary, that $[A^T|-c]^T \in \mathcal{H}$. Then the system $A^T y - \lambda c = 0, e_m^T y + \lambda = 1, (y,\lambda) \geq 0$ is feasible. (Notice here that $\mathcal{H}$ is now a set of matrices in $\mathbf{R}^{(m+1)\times n}$.) We cannot have a feasible point with $\lambda \neq 0$, for in that case $A^T y = c, y \geq 0$ would be feasible, which is a contradiction. Then $\lambda = 0$ and let y be such that $A^T y = 0, y \geq 0, y \neq 0$. Let j be such that $y_j \neq 0$. Take, for the jth row of A, $\Delta A_j = \epsilon c/y_j$ for $\epsilon > 0$, and $\Delta A_i = 0$ for $i \neq j$. Then it follows that $(A+\Delta A)^T y = \epsilon c, y \geq 0$. This means that for ϵ arbitrarily small, $(A+\Delta A, b, c) \in \mathcal{F}_1^D$, which is a contradiction.

Conversely, if $[A^T|-c]^T \in \mathcal{H}^C$ we have that the system $A^T y - \lambda c = 0, e_m^T y + \lambda = 1, (y,\lambda) \geq 0$ is infeasible. But this implies that the system $Ax \leq -e_m, c^T x \geq 1$ is feasible and hence $(A,b,c) \in (\mathcal{F}_1^D)^C$. Since $\mathcal{H}^C$ is an open set, the argument applies to any instance in a certain neighborhood of (A,b,c). We conclude that $(A,b,c) \in \text{int}(\mathcal{F}_1^D)^C$. □

The previous propositions allow us to analyze $A^T y = c, y \geq 0$ if we know how

to analyze $A^T y = 0, y \geq 0, e_m^T y = 1$. We basically incorporate $-c$ as another row of A. Moreover, the analysis of the system $A^T y = 0, e_m^T y = 1, y \geq 0$ is the same as analyzing the system $A^T y = 0, e_m^T y = 1, y \geq 0, -e_m^T y < 0$, which is equivalent to applying the decision procedures of §3.1 to the problem instance $(A, -e_m, c)$. Since we are including $-c$ as another column, the decision reduces to using the algorithms already developed on an instance of the form $([A^T|-c]^T, [-e_m^T, -1]^T, c)$. Observe that this instance corresponds to a linear program with $m+1$ constraints and n variables. We will denote by $\bar{\mathcal{F}}_1^P$ and $\bar{\mathcal{F}}_1^D$ the sets of primal and dual feasible instances for linear problems of dimension $(m+1) \times n$.

PROPOSITION 3.15. *Let $(A, b, c) \in \mathcal{F}_1^D$ and let $([A^T| - c]^T, [-e_m^T, -1]^T, c)$ be the corresponding instance in $(\bar{\mathcal{F}}_1^P)^C$. Let*

$$\rho = dist_\infty((A, b, c), (\mathcal{F}_1^D)^C),$$
$$\rho' = dist_\infty\left(\left(\begin{bmatrix} A \\ -c^T \end{bmatrix}, \begin{bmatrix} -e_m \\ -1 \end{bmatrix}, c\right), \bar{\mathcal{F}}_1^P\right).$$

Then $\rho' \geq \rho$.

Proof. Begin by taking $(A, b, c) \in \text{int}(\mathcal{F}_1^D)$. Let (A', b', c') be such that $\|(A, b, c) - (A', b', c')\|_\infty = \delta < \rho$. Observe that we can always assume that $\|b - b'\|_\infty \leq \|(A, c) - (A', c')\|_\infty$, since the dual feasibility does not depend on b. Then $(A', b', c') \in \text{int}(\mathcal{F}_1^D)$ and hence, by Proposition 3.13, $[A'^T| - c']^T \in \text{int}(\mathcal{H})$. Then the system $(A')^T y - \lambda c' = 0, -e_m^T y - \lambda < 0$ is feasible. This implies that the system $(A')x \leq -e_m, (c')^T x \geq 1$ is infeasible, implying that $([A'^T| - c']^T, [-e_m^T| - 1], c') \in (\bar{\mathcal{F}}_1^P)^C$. Now,

$$\delta = \|(A, b, c) - (A', b', c')\|_\infty = \left\| \left(\begin{bmatrix} A \\ -c^T \end{bmatrix}, \begin{bmatrix} -e_m \\ -1 \end{bmatrix}, c\right) - \left(\begin{bmatrix} A' \\ -c'^T \end{bmatrix}, \begin{bmatrix} -e_m \\ -1 \end{bmatrix}, c'\right) \right\|_\infty \leq \rho'.$$

So, $\delta \leq \rho'$. Since this is true for any $\delta < \rho$ we conclude that $\rho \leq \rho'$. □

All the previous results allow us to reduce the problem of deciding dual feasibility or infeasibility to the problem of deciding primal feasibility or infeasibility of a particular instance of the form $([A^T| - c]^T, [-e_m| - 1]^T, c)$.

PROPOSITION 3.16. *There exists a polynomial time algorithm which, given $(\bar{A}, \bar{b}, \bar{c})$ and $\delta > 0$ rationals, such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$, will stop with one of the following alternatives:*

A : $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1^D$ and hence $(A, b, c) \in \mathcal{F}_1^D$,
B : $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^D)^C$ and hence $(A, b, c) \in (\mathcal{F}_1^D)^C$,
C : Cannot decide. Please provide a smaller δ.

If $(A, b, c) \notin \mathcal{B}_1$ then the algorithm will stop with alternative A or B whenever $\delta \leq \rho^3/(64(m+3)^2(m+1)n)$ at most, where $\rho = dist_\infty((A, b, c), \mathcal{B}_1)$.

Proof. The algorithm is as follows.

FUNCTION DUAL-FEAS$(\bar{A}, \bar{b}, \bar{c}, \delta)$
PRIMAL-FEAS$([\bar{A}^T| - \bar{c}]^T, [-e_m^T| - 1]^T, \bar{c})$
If $B_\infty(([\bar{A}^T| - \bar{c}]^T, [-e_m^T| - 1]^T, \bar{c}), \delta) \subset \bar{\mathcal{F}}_1^P$ **then**
  RETURN $(B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset (\mathcal{F}_1^D)^C)$
else
  If $B_\infty(([\bar{A}^T| - \bar{c}]^T, [-e_m^T| - 1]^T, \bar{c}), \delta) \subset (\bar{\mathcal{F}}_1^P)^C$ **then**

RETURN $(B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^D)$
else
RETURN(Please provide a smaller δ)
end

The algorithm runs in polynomial time. The precision requirement on δ, from the previous subsection, is $\delta \leq (\rho')^3/(64(m+3)^2(m+1)n)$, where ρ' is as defined in Proposition 3.15. Since $\rho \leq \rho'$, the proposed bound on δ follows. □

4. The decision algorithm. Now we are ready to put all the procedures together. Suppose we want to know whether or not the linear programming problem $\max\{c^T x : Ax \leq b\}$ has a solution. Suppose also that we have an approximation $B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$, where $(\bar{A},\bar{b},\bar{c}),\delta$ are all rational data. Then what we must do is test whether or not all problems in that ball have a solution. Specifically, we must test whether or not all problems in the ball are feasible, and if they are feasible whether or not they are dual feasible. The following procedure will test for those cases.

FUNCTION SOLUTION-EXISTENCE1$(\bar{A},\bar{b},\bar{c},\delta)$
PRIMAL-FEAS$(\bar{A},\bar{b},\bar{c},\delta)$
If $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^P)^C$ **then**
RETURN(Problem is infeasible)
else
If $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^P$ **then**
DUAL-FEAS$(\bar{A},\bar{b},\bar{c},\delta)$
If $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset (\mathcal{F}_1^D)^C$ **then**
RETURN(Problem is unbounded)
else
If $B_\infty((\bar{A},\bar{b},\bar{c}),\delta) \subset \mathcal{F}_1^D$ **then**
RETURN(Problem has a solution)
else
RETURN(Please provide a smaller δ)
end
end
else
RETURN(Please provide a smaller δ)
end
end

Suppose that $(A,b,c) \notin \mathcal{B}_1$ and let $\rho = dist_\infty((A,b,c),\mathcal{B}_1) > 0$. Then it follows from our previous results that as long as δ is small enough, one of the conditions for deciding the different alternatives will hold. In particular, if (A,b,c) is feasible, as long as $\delta \leq \rho/2$, any ball of the form $B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$ such that $(A,b,c) \in B_\infty((\bar{A},\bar{b},\bar{c}),\delta)$ will be contained in $\mathcal{F}_1^P$. Hence, the condition in Proposition 3.1 will hold, and the function PRIMAL-FEAS will give a positive answer. Now, if (A,b,c) is also bounded, then as long as δ is less than or equal to a number in the order of $\rho^3/(mn)^3$, function DUAL-FEAS will give a positive answer. If (A,b,c) is infeasible, PRIMAL-FEAS will detect that as soon as δ has been reduced to the order of $\rho^3/(mn)^3$. If none of the conditions hold, the procedure will ask for more precision. In any case, it follows that if δ is not smaller than a number in the order of $\rho^3/f_1(m,n)$, this procedure will detect whether or not the problem does have a solution. Here, f_1 is a polynomial in m and n. Equivalently, this procedure will efficiently detect existence

or nonexistence of solutions for the linear problem using roughly three times the minimal precision needed by any algorithm. We have also shown that the subroutines used in SOLUTION-EXISTENCE1 are computationally efficient because they run in polynomial time. With this we have finally proved the following theorem.

THEOREM 4.1. *For problem P_1, there exists a polynomial time algorithm which, given $(\bar{A}, \bar{b}, \bar{c}), \delta$, rational data, such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$, will either tell us that the problem does or does not have a solution, or will request more precision. The algorithm will always identify whether or not (A, b, c) has solutions when $\delta = \rho^3/f_1(n, m)$ or less, where f_1 is a polynomial and $\rho = dist_\infty((A, b, c), \mathcal{B}_1)$.*

Theorem 4.2, applicable to problem P_2, follows immediately as a corollary of Theorem 4.1 by noticing that formulation 2 has a structure dual to formulation 1.

THEOREM 4.2. *For problem P_2, there exists a polynomial time algorithm which, given $(\bar{A}, \bar{b}, \bar{c}), \delta$, rational data, such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$, will either tell us that the problem does or does not have a solution, or will request more precision. The algorithm will always identify whether or not (A, b, c) has solutions when $\delta = \rho^3/f_2(n, m)$ or less, where f_2 is a polynomial and $\rho = dist_\infty((A, b, c), \mathcal{B}_2)$.*

We can see here that these two theorems establish the "data complexity" of deciding existence of solutions for linear programming with approximate data. They basically say that we can use roughly three times the minimal precision required by any algorithm, if we want to do it in a computationally efficient way, in this case using polynomial time algorithms.

We can establish a relation between the total computational effort and the amount of data needed, but to do this we must specify the way we obtain the approximate data. Instead of doing this we will simply rely on an oracle. Suppose that there exists a function INTERVAL(δ) that will, given input δ, produce an instance $(\bar{A}, \bar{b}, \bar{c})$ such that $(A, b, c) \in B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta)$. Assume furthermore that if δ is a rational number then the output is also rational. Then, given a precision, this oracle will give us an approximation within that precision. A further property must be required of this oracle in order to link the computational effort to the distance to ill-posedness.

DEFINITION 4.1. *We say the oracle INTERVAL is bit consistent if the bit length of the output $(\bar{A}, \bar{b}, \bar{c})$ is $O(mn \log(1/\delta))$.*

An example of such an oracle is a procedure that generates the first $\log(1/\delta)$ digits of the binary expansion of each number in (A, b, c). So, assume now that the approximation to (A, b, c) has been produced by a bit-consistent data oracle. Observe that the running time of SOLUTION-EXISTENCE1 is dominated by the running time of PRIMAL-FEAS and DUAL-FEAS. We know that both functions run in time bounded by polynomials in the bit length of the input. Now, the bit length of the input is proportional to m, n, and $\log(1/\delta)$. So, we conclude that the total running time (in number of bits operations) of SOLUTION-EXISTENCE1 is bounded by a polynomial in m, n, and $\log(1/\delta)$. Now, to be able to determine whether the problem does or does not have solutions, we need a δ not smaller than $\rho^3/f_1(m, n)$. So, we conclude that the determination of existence or nonexistence of solutions for $\max\{c^T x : Ax \leq b\}$ requires a computational effort bounded by a polynomial in m, n, and $\log(1/\rho)$ that is polynomial in the natural size of the problem and the logarithm of the condition measure of the problem instance. We can see in this context how $\log(1/\rho)$ is playing a role analogous to the role of "L" in bit complexity theory.

The algorithms we have just shown can also be used to obtain an estimate of the distance from a problem instance to ill-posedness. The idea is basically to iterate, reducing δ until the function SOLUTION-EXISTENCE answer is "yes." Observe

that if we have determined that $B_\infty((\bar{A}, \bar{b}, \bar{c}), \delta) \subset \mathcal{F}_1$, then if $B_\infty((\bar{A}, \bar{b}, \bar{c}), 2\delta) \subset \mathcal{F}_1$ we can conclude that $\delta < \rho/2$. This means that all we have to do is continue reducing δ until that is satisfied. Also observe that $1/\delta$ is immediately an upper bound on the condition measure for the instance $(\bar{A}, \bar{b}, \bar{c})$.

5. Conclusions. In this paper we have shown fully efficient algorithms for deciding existence of solutions to linear programming problems known with approximate data. The procedures are both computationally efficient and data efficient and hence we have achieved a balance between these two factors. How close the instance is to ill-posedness plays a central role in the data and computational requirements for deciding the question. In fact, the closer the instance is to ill-posedness, the more precise knowledge of the data will be needed and more computations will have to be performed. The complexity of the problem in this setting appears then related to a measure of its "conditioning." Although we have stated and analyzed here only an existence question, more "computational" questions can also be analyzed in this context. In fact, it is possible to compute an approximate solution to a linear program based only on approximate data, as shown in Vera [12]. As in this paper, the closer the instance is to ill-posedness, the more precise an approximation will be needed and more computation will be required. It is also interesting to point out that although the analysis we have done here uses the classical bit complexity model as an underlying model of computation for analyzing the algorithms, an alternate model can be used here as well, such as the real numbers model of Blum, Shub, and Smale [1]. Renegar [8] has pointed this out as well. The form of our results will remain basically the same as long as efficient algorithms for linear programming exist in this other model. An interesting extension of the analysis in this paper is the consideration of additional knowledge about the problem instance. It is intuitive that a certain problem that is ill posed might become solvable if some additional knowledge is incorporated. The notion of ill-posedness must be revised to consider this situation and some development toward that can be found in Filipowski [4]. It is also worth pointing out that the ideas in this paper can be extensively generalized to other problems in the context of functional analysis, as has been done by Renegar [9].

Acknowledgments. I would like to thank Professor James Renegar for many comments and discussion about this work, as well as the two anonymous referees whose comments have been very useful in improving this paper.

REFERENCES

[1] L. Blum, M. Shub, and S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc., 21 (1989), pp. 1–46.

[2] J. W. Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math., 51 (1987), pp. 251–289.

[3] ———, *The probability that a numerical analysis problem is difficult*, Math. Comp., 50 (1988), pp. 449–480.

[4] S. Filipowski, *Towards a Computational Complexity Theory that Uses Approximate Data and Knowledge*, Ph.D. dissertation, Cornell University, Ithaca, NY, 1993.

[5] G. H. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[6] J. Renegar, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–94.

[7] ———, *Is It Possible to Know a Problem Instance Is Ill-Posed? Some Foundations for a General Theory of Condition Number*, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, Tech. report 939, October 1990.

[8] ———, *Incorporating condition measures into the complexity theory of linear programming*, SIAM J. Optim., 5 (1995), pp. 506–524.

[9] ———, *Linear programming, complexity theory and elementary functional analysis*, Math. Programming, 70 (1995), pp. 279–351.

[10] S. SMALE, *Some remarks on the foundations of numerical analysis*, SIAM Rev., 32 (1990) pp. 211–220.

[11] P. VAIDYA, *A locally well-behaved potential function and a simple Newton-type method for finding the center of a polytope*, in Progress in Mathematical Programming—Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, Berlin, 1989, pp. 79–90.

[12] J. VERA, *Ill-Posedness in Mathematical Programming and Problem Solving with Approximate Data*, Ph.D. dissertation, Cornell University, Ithaca, NY, 1992.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 570–586, August 1996
© 1996 Society for Industrial and Applied Mathematics
002

AN ASYMPTOTICAL $O(\sqrt{n}L)$-ITERATION PATH-FOLLOWING LINEAR PROGRAMMING ALGORITHM THAT USES WIDE NEIGHBORHOODS*

PI-FANG HUNG† AND YINYU YE‡

Abstract. Path-following linear programming (LP) algorithms generate a sequence of points within certain neighborhoods of a central path $\mathcal{C}$ which prevent iterates from prematurely getting too close to the boundary of the feasible region. Depending on the norm used, these neighborhoods include $\mathcal{N}_2(\beta)$, $\mathcal{N}_\infty(\beta)$, and $\mathcal{N}_\infty^-(\beta)$, where $\beta \in (0,1)$ and

$$\mathcal{C} \subset \mathcal{N}_2(\beta) \subset \mathcal{N}_\infty(\beta) \subset \mathcal{N}_\infty^-(\beta) \quad \text{for each} \quad \beta \in (0,1).$$

A paradox is that among all existing (infeasible or feasible) path-following algorithms, the theoretical iteration complexity, $O(\sqrt{n}L)$, of small-neighborhood ($\mathcal{N}_2$) algorithms is significantly better than the complexity, $O(nL)$, of wide-neighborhood ($\mathcal{N}_\infty^-$) algorithms. In practice, wide-neighborhood algorithms outperform small-neighborhood ones by a big margin. Here, n is the number of LP variables and L is the LP data length. In this paper, we present an $O(n^{\frac{n+1}{2n}}L)$-iteration (infeasible) primal-dual high-order algorithm that uses wide neighborhoods. Note that this iteration bound is asymptotically $O(\sqrt{n}L)$, i.e., the best bound for small-neighborhood algorithms, as n increases.

Key words. linear programming, central path, interior-point algorithm, complexity

AMS subject classifications. 90C05, 65K05

1. Introduction.

Consider a linear programming (LP) problem in the standard form:

$$(LP) \quad \begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b,\ x \geq 0, \end{array}$$

where $c \in \mathbf{R}^n$, $A \in \mathbf{R}^{m\times n}$, and $b \in \mathbf{R}^m$ are given; $x \in \mathbf{R}^n$; and ^{T} denotes the transpose. (LP) is said to be feasible if and only if its constraints are consistent; it is called unbounded if there is a sequence $\{x^k\}$ such that x^k is feasible for all k but $c^T x^k \to -\infty$. (LP) has a solution if and only if it is feasible and bounded.

The dual problem of (LP) can be written as

$$(LD) \quad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & A^T y \leq c, \end{array}$$

where $y \in \mathbf{R}^m$. We call $s := c - A^T y \in \mathbf{R}^n$ dual slacks. Denote by $\mathcal{F}$ the set of all (x,s) such that x and (y,z) are feasible for the primal and dual, respectively. Denote by $\mathcal{F}^0$ the set of points with $(x,s) > 0$ in $\mathcal{F}$.

Assuming that the problem has a feasible interior point, Bayer and Lagarias [2], Megiddo [11], and Sonnevend [22] defined the central path for a feasible problem as

$$\mathcal{C} = \left\{ (x,s) \in \mathcal{F}^0 : Xs = \frac{x^T s}{n} e \right\}$$

* Received by the editors April 29, 1994; accepted for publication (in revised form) April 7, 1995. This work was supported in part by NSF grant DDM-9207347. This work was based on the first author's Ph.D. thesis in the Applied Mathematical and Computational Sciences Program, Department of Mathematics, University of Iowa, Iowa City, IA 52242.

† Department of Mathematics, Tunghai University, Taichung, Taiwan 40704, ROC.

‡ Department of Management Sciences, University of Iowa, Iowa City, IA 52242.

in primal-dual form, where e denotes the vector of ones and $X = \mathrm{diag}(x)$ denotes the diagonal matrix with diagonal entries equal to the elements of x. As $x^T s \to 0$, this path leads to a strictly complementary solution of (LP) and (LD).

There are two types of polynomial interior-point algorithms. One is called a potential reduction algorithm and is not based on the central path and its neighborhood. This group includes the $O(nL)$-iteration projective-scaling algorithm (e.g., Karmarkar [7]) and the $O(\sqrt{n}L)$-iteration affine-scaling algorithm (e.g., Anstreicher and Bosch and references therein [1]) among others, where each iteration uses $O(n^3)$ arithmetic operations. Here, L represents the data length of rational LP data or the precision to achieve $(x^T s)/(x^0)^T s^0 \le 2^{-L}$ for a given initial point (x^0, s^0). The practical value of these algorithms relies on an efficient line search technique to select step size, which could be costly for solving some LP problems. (Several other long-step strategies were recently proposed by Nesterov [17].)

The second type of polynomial interior-point algorithm is called a path-following algorithm. Path-following algorithms generate a sequence of points within certain neighborhoods of the central path $\mathcal{C}$ which prevent iterates from prematurely getting too close to the boundary of the feasible region. Depending on the norm used in their definitions, these neighborhoods include

$$\mathcal{N}_2(\beta) = \left\{ (x,s) \in \mathcal{F}^0 : \|Xs - \mu e\| \le (1-\beta)\mu \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\}$$

for some $\beta \in (0,1)$ (where $\|\cdot\|$ without subscript designates the ℓ_2 norm),

$$\mathcal{N}_\infty(\beta) = \left\{ (x,s) \in \mathcal{F}^0 : \|Xs - \mu e\|_\infty \le (1-\beta)\mu \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\}$$

and

$$\mathcal{N}_\infty^-(\beta) = \left\{ (x,s) \in \mathcal{F}^0 : \|Xs - \mu e\|_\infty^- \le (1-\beta)\mu \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\}$$

for any $\beta \in (0,1)$. Here, for any $z \in \mathbf{R}^n$,

$$\|z\|_\infty^- := \|z^-\|_\infty$$

and

$$\|z\|_\infty^+ := \|z^+\|_\infty,$$

where $(z^-)_j := \min\{z_j, 0\}$ and $(z^+)_j := \max\{z_j, 0\}$ and $\|\cdot\|_\infty$ is the usual ℓ_∞ norm. Note that $\|z\|_\infty = \max\{\|z\|_\infty^+, \|z\|_\infty^-\}$ and that neither $\|\cdot\|_\infty^-$ nor $\|\cdot\|_\infty^+$ is a norm, although they obey the triangle inequality. Hence, $\mathcal{N}_\infty^-(\beta)$ can be written as

$$\mathcal{N}_\infty^-(\beta) = \left\{ (x,s) \in \mathcal{F}^0 : Xs \ge \beta\mu e \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\}.$$

We easily see that

$$\mathcal{C} \subset \mathcal{N}_2(\beta) \subset \mathcal{N}_\infty(\beta) \subset \mathcal{N}_\infty^-(\beta) \subset \mathcal{F}^0 \text{ for each } \beta \in (0,1).$$

In particular,

$$\mathcal{N}_\infty^-(0) = \mathcal{F}^0,$$

so when β is close to 0, the neighborhood $\mathcal{N}_\infty^-(\beta)$ spreads almost all over the feasible region $\mathcal{F}$ and the points generated by the algorithm based on $\mathcal{N}_\infty^-(\beta)$ are close to the boundary rather than to the central path.

Among all existing path-following (infeasible or feasible) LP algorithms, the theoretical iteration complexity of small-neighborhood ($\mathcal{N}_2$) algorithms is $O(\sqrt{n}L)$ (e.g., Renegar [21], Gonzaga [5], Kojima, Mizuno, and Yoshise [9], and Monteiro and Adler [13]), and the complexity of wide-neighborhood ($\mathcal{N}_\infty$ or $\mathcal{N}_\infty^-$) algorithms is at least $O(nL)$ (e.g., Kojima, Mizuno, and Yoshise [8], Mizuno, Todd, and Ye [15], Potra [19], and Roos and Vial [20]). In contrast, wide-neighborhood algorithms outperform small-neighborhood ones by a big margin in practice (e.g., Lustig, Marsten, and Shanno [10] and Mehrotra [12]). It seems that smaller neighborhoods generally restrict all iterates moved by a short step, and they might be too conservative for solving real LP problems.

A few results have been developed in attempting to improve the theoretical complexity of wide-neighborhood algorithms that use $\mathcal{N}_\infty$ or $\mathcal{N}_\infty^-$. One result is due to Monteiro, Adler, and Resende [14], who developed an r-order power-series algorithm ($r \in [1, n]$), where iterative points move along an r-order polynomial curve, with iteration complexity $O(n^{\frac{r+1}{2r}} L^{\frac{r+1}{r}})$. Their algorithm requires an initial feasible point in $\mathcal{N}_\infty(1-\beta^0)$ with $\beta^0 \in (0,1)$. Then it generates a direction and moves the iterate in this direction by a fixed step size $O(n^{\frac{-r-1}{2r}} L^{\frac{-1}{r}})$. They have shown that the new iterate will be in $\mathcal{N}_\infty(1-\beta^1)$ where $\beta^1 > \beta^0$. In general, the kth iterate will be in $\mathcal{N}_\infty(1-\beta^k)$ while β^k increases and approaches an upper bound $(1+\beta^0)/2$. We can see that all iterates lie in $\mathcal{N}_\infty((1-\beta^0)/2)$. Unfortunately, their preplanned step size selection still takes a very short step in each iteration.

The second result is due to Zhang and Zhang [27], who proved that a 2-order power-series algorithm, similar to the one introduced by Monteiro, Adler, and Resende [14] and Mehrotra [12], possesses $O(n^{3/4}L)$-iteration complexity under the assumption that the initial point is feasible and in $\mathcal{N}_\infty^-(\beta)$. Subsequently, all iterates lie in $\mathcal{N}_\infty^-(\beta)$, where β is fixed during entire iterations. If the initial positive point is not feasible, then after using a "big-M" technique, their algorithm's iteration complexity is $O(n^{3/2}L)$, again greater than $O(nL)$. Tapia et al. [23] also studied the high-order predictor–corrector algorithm and compared it with the composite Newton method.

On the other hand, the high-order power-series algorithm has gained tremendous popularity in practice due to the computational results of Mehrotra [12], Lustig, Marsten, and Shanno [10], and Carpenter et al. [3], among others. Their high-order algorithms actually use an infeasible initial point and allow iterative points to move in $\mathcal{N}_\infty^\sim(\beta)$, where

$$\mathcal{N}_\infty^\sim(\beta) = \left\{ (x \geq 0, s \geq 0) : Xs \geq \beta\mu e \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\},$$

and β is fixed in the course of iterations and it is very close to 0. Obviously,

$$\mathcal{N}_\infty^-(\beta) \subset \mathcal{N}_\infty^\sim(\beta),$$

and $\mathcal{N}_\infty^\sim(0)$ spreads the whole nonnegative orthant. To our knowledge, no such algorithm, which is based on $\mathcal{N}_\infty^\sim(\beta)$ and whose complexity is below $O(nL)$, is ever developed or proved.

Therefore, it seems that there is a paradox between theory and practice. The question is, can we resolve this conflict?

In this paper, we present an r-order power-series primal-dual path-following algorithm that is based on $\mathcal{N}_\infty^-(\beta)$, where β is any fixed constant in $(0,1)$. We show that its iteration complexity is $O(n^{\frac{r+1}{2r}}L)$, where $r \in [1,n]$. Again, each iteration uses $O(n^3)$ arithmetic operations. Note that if we let $r = n$, then this iteration bound is asymptotical $O(\sqrt{n}L)$ as n increases. Moreover, our algorithm does not need the initial point to be feasible and does not use any big-M parameter; that is, the algorithm is actually based on $\mathcal{N}_\infty^\sim(\beta)$. It solves LP problems without any regularity assumption concerning the existence of optimal, feasible, or interior feasible solutions. If the LP problem has a solution, the algorithm generates a sequence that approaches feasibility and optimality simultaneously; if the problem is infeasible or unbounded, the algorithm correctly detects infeasibility.

Our algorithm is built upon the results mentioned above, and it is an extension of the homogeneous self-dual (HSD) algorithm studied by Ye, Todd, and Mizuno [26] and later by Xu and Ye [24] and Xu, Hung, and Ye [25]. We describe our generic algorithm in the next section and develop several general lemmas. Then we present a special algorithm in §3 and analyze its complexity in §4. Another special algorithm is also shown in §5. Finally, conclusions and implications based on our main result are summarized in §6.

2. The r-order power-series and HSD algorithm. Recently, Xu, Hung, and Ye [25] proposed a simplified version of the HSD algorithm of Ye, Todd, and Mizuno [26]. The algorithm deals with a homogeneous (and self-dual) linear feasibility (HLF) model of Goldman and Tucker [4]:

$$
\begin{array}{lcccl}
(HLF) & & Ax & -b\tau & = 0, \\
 & -A^T y & & +c\tau & \geq 0, \\
 & b^T y & -c^T x & & \geq 0, \\
 & y \text{ free} & x \geq 0 & \tau \geq 0. &
\end{array} \tag{1}
$$

Denote by s the slack vector for the second (inequality) constraint and by κ the slack scalar for the third (inequality) constraint. Then the problem is to find a strictly complementary point such that

$$x^T s = 0, \ \tau\kappa = 0 \quad \text{and} \quad x + s > 0, \ \tau + \kappa > 0.$$

Given that $(y^k, \ x^k > 0, \tau^k > 0, s^k > 0, \kappa^k > 0)$ is an (infeasible) interior point in $\mathcal{N}_\infty^\sim(\beta)$ for (HLF), where $\beta \in (0,1)$, the kth iteration of the HSD algorithm solves the following system of linear equations for the direction $(d_y, d_x, d_\tau, d_s, d_\kappa)$:

$$
\begin{array}{rrrrrcl}
 & A\ d_x & -b\ d_\tau & & & = & \eta\ r_P^k, \\
-A^T\ d_y & & +c\ d_\tau & -d_s & & = & -\eta\ r_D^k, \\
b^T\ d_y & -c^T\ d_x & & & -d_\kappa & = & \eta\ r_G^k
\end{array}
$$

and

$$
\begin{aligned}
X^k d_s + S^k d_x &= \gamma\mu^k e - X^k s^k, \\
\tau^k d_\kappa + \kappa^k d_\tau &= \gamma\mu^k - \tau^k\kappa^k,
\end{aligned}
$$

where nonnegative η and γ are two parameters and

$$\mu^k = ((x^k)^T s^k + \tau^k\kappa^k)/(n+1),$$

$$r_P^k = b\tau^k - Ax^k, \quad r_D^k = c\tau^k - A^T y^k - s^k, \quad r_G^k = c^T x^k - b^T y^k + \kappa^k.$$

Xu, Hung, and Ye [25] showed that if $\eta = 1 - \gamma$ is chosen in each iteration for some $0 < \gamma < 1$, then the algorithm becomes the HSD algorithm of Ye, Todd, and Mizuno [26], which generates iterates such that

$$\frac{\mu^k}{\theta^k} = \frac{\mu^0}{\theta^0} \quad \text{for all} \quad k \geq 0$$

and

$$r_P^k = \theta^k r_P^0, \quad r_D^k = \theta^k r_D^0, \quad r_G^k = \theta^k r_G^0, \quad \text{with} \quad \theta^0 = 1.$$

Here, μ^k represents the complementarity gap and θ^k measures the infeasibility error at the kth iteration. The above relation indicates that both measures tend to 0 at a same rate. Furthermore, (HLF) is self-dual and its solution set is unbounded. Thus, based on Theorem 4 of Mizuno, Todd, and Ye [16] (also see Lemma 2 of Güler and Ye [6]), every limit point of iterates is strictly complementary for (HLF) as long as the iterates lie in $\mathcal{N}_\infty^\sim(\beta)$. Moreover, if $\tau^k \to 0$, then the original (LP) problem is either infeasible or unbounded.

Now, we describe a generic high-order power-series and HSD algorithm with the same feature.

THE r-ORDER HSD ALGORITHM.

Given that $(y^k,\ x^k > 0, \tau^k > 0, s^k > 0, \kappa^k > 0)$ is an (infeasible) interior point in $\mathcal{N}_\infty^\sim(\beta)$ for (HLF),

$$\mathcal{N}_\infty^\sim(\beta) = \{(x \geq 0, s \geq 0, \tau \geq 0, \kappa > 0) : Xs \geq \beta\mu e \quad \text{and} \quad \tau\kappa \geq \beta\mu\},$$

where $\beta \in (0,1)$ and it is fixed during iterations, and

$$\mu = (x^T s + \tau\kappa)/(n+1).$$

Given integer $r \geq 1$ and scalar parameters $\gamma \geq 0$, $\eta^{(j)} \geq 0, j = 1, \ldots, r$, the kth iteration of the algorithm consists of the following steps.

1. Solve for the first-order direction $(d_y^{(1)}, d_x^{(1)}, d_\tau^{(1)}, d_s^{(1)}, d_\kappa^{(1)})$ from

$$\begin{array}{rrrrrcl} & A\, d_x^{(1)} & -b\, d_\tau^{(1)} & & & = & \eta^{(1)} r_P^k, \\ -A^T\, d_y^{(1)} & & +c\, d_\tau^{(1)} & -d_s^{(1)} & & = & -\eta^{(1)} r_D^k, \\ b^T\, d_y^{(1)} & -c^T\, d_x^{(1)} & & & -d_\kappa^{(1)} & = & \eta^{(1)} r_G^k, \end{array} \tag{2}$$

and

$$\begin{array}{rcl} X^k d_s^{(1)} + S^k d_x^{(1)} & = & \gamma\mu^k e - X^k s^k, \\ \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)} & = & \gamma\mu^k - \tau^k\kappa^k. \end{array} \tag{3}$$

2. For $j = 2, 3, \ldots, r$, solve for the jth-order direction $(d_y^{(j)}, d_x^{(j)}, d_\tau^{(j)}, d_s^{(j)}, d_\kappa^{(j)})$ from

$$\begin{array}{rrrrrcl} & A\, d_x^{(j)} & -b\, d_\tau^{(j)} & & & = & \eta^{(j)}\, r_P^k, \\ -A^T\, d_y^{(j)} & & +c\, d_\tau^{(j)} & -d_s^{(j)} & & = & -\eta^{(j)}\, r_D^k, \\ b^T\, d_y^{(j)} & -c^T\, d_x^{(j)} & & & -d_\kappa^{(j)} & = & \eta^{(j)}\, r_G^k, \end{array} \tag{4}$$

and

$$\begin{array}{rcl} X^k d_s^{(j)} + S^k d_x^{(j)} & = & -\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}, \\ \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)} & = & -\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}. \end{array} \tag{5}$$

3. Calculate $f^{(j)}$ according to

$$f^{(1)} = \eta^{(1)} + \gamma - 1, \quad f^{(j)} = \eta^{(j)} + \sum_{t=1}^{j-1} \eta^{(t)} f^{(j-t)}, \quad 2 \le j \le r,$$

and choose a step size $\alpha^k > 0$ such that the new iterate,

$$\begin{aligned}
x^{k+1} &= \left(x^k + \textstyle\sum_{j=1}^r (\alpha^k)^j d_x^{(j)}\right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}\right), \\
y^{k+1} &= \left(y^k + \textstyle\sum_{j=1}^r (\alpha^k)^j d_y^{(j)}\right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}\right), \\
s^{k+1} &= \left(s^k + \textstyle\sum_{j=1}^r (\alpha^k)^j d_s^{(j)}\right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}\right), \\
\tau^{k+1} &= \left(\tau^k + \textstyle\sum_{j=1}^r (\alpha^k)^j d_\tau^{(j)}\right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}\right), \\
\kappa^{k+1} &= \left(\kappa^k + \textstyle\sum_{j=1}^r (\alpha^k)^j d_\kappa^{(j)}\right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}\right),
\end{aligned}$$

lies in $\mathcal{N}_\infty^{\sim}(\beta)$.

Regarding the algorithm, we have the following lemmas, which are similar to those proved in Xu and Ye [24] and Xu, Hung, and Ye [25].

LEMMA 2.1. *The direction defined by* (2), (3), (4), *and* (5) *satisfies*

$$\begin{aligned}
&(d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} + d_\tau^{(i)} d_\kappa^{(j)} + d_\tau^{(j)} d_\kappa^{(i)} \\
&= -(\eta^{(i)} f^{(j)} + \eta^{(j)} f^{(i)})((x^k)^T s^k + \tau^k \kappa^k)
\end{aligned} \tag{6}$$

and

$$(d_x^{(j)})^T d_s^{(j)} + d_\tau^{(j)} d_\kappa^{(j)} = -\eta^{(j)} f^{(j)} ((x^k)^T s^k + \tau^k \kappa^k) \tag{7}$$

for $i, j = 1, \ldots, r$.

Proof. First, we have

$$\begin{aligned}
&(d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} + d_\tau^{(i)} d_\kappa^{(j)} + d_\tau^{(j)} d_\kappa^{(i)} \\
&= (d_x^{(i)})^T (-A^T\, d_y^{(j)} + c\, d_\tau^{(j)} + \eta^{(j)} r_D^k) + (d_x^{(j)})^T (-A^T\, d_y^{(i)} + c\, d_\tau^{(i)} + \eta^{(i)} r_D^k) \\
&\quad + d_\tau^{(i)} (b^T\, d_y^{(j)} - c^T\, d_x^{(j)} - \eta^{(j)} r_G^k) + d_\tau^{(j)} (b^T\, d_y^{(i)} - c^T\, d_x^{(i)} - \eta^{(i)} r_G^k) \\
&= -\eta^{(i)} \left((d_y^{(j)})^T r_P^k - (d_x^{(j)})^T r_D^k + d_\tau^{(j)}\, r_G^k \right) - \eta^{(j)} \left((d_y^{(i)})^T r_P^k - (d_x^{(i)})^T r_D^k + d_\tau^{(i)}\, r_G^k \right) \\
&= -\eta^{(i)} \left\{ \eta^{(j)} ((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(j)} + (s^k)^T d_x^{(j)} + \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)}) \right\} \\
&\quad - \eta^{(j)} \left\{ \eta^{(i)} ((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(i)} + (s^k)^T d_x^{(i)} + \tau^k d_\kappa^{(i)} + \kappa^k d_\tau^{(i)}) \right\}.
\end{aligned}$$

Now, we prove the lemma by induction on i, j. Note that

$$\begin{aligned}
&\eta^{(1)} ((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(1)} + (s^k)^T d_x^{(1)} + \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)}) \\
&= (\eta^{(1)} + \gamma - 1)((x^k)^T s^k + \tau^k \kappa^k)
\end{aligned}$$

implies that (7) is true when $i = j = 1$. Assume (6) and (7) hold for all i, j with $1 \le i, j \le p - 1$. We show that they hold for $i = p$ and $j = 1, \ldots, p$. In fact, we only need to prove that for $j = 1, \ldots, p$,

$$\eta^{(j)} ((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(j)} + (s^k)^T d_x^{(j)} + \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)}) = f^{(j)} ((x^k)^T s^k + \tau^k \kappa^k),$$

which is true because

$$\begin{aligned}
&\eta^{(j)}((x^k)^T s^k + \tau^k\kappa^k) + ((x^k)^T d_s^{(j)} + (s^k)^T d_x^{(j)} + \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)}) \\
&= \eta^{(j)}((x^k)^T s^k + \tau^k\kappa^k) - \textstyle\sum_{t=1}^{j-1}\left((d_x^{(t)})^T d_s^{(j-t)} + d_\tau^{(t)}\ d_\kappa^{(j-t)}\right) \\
&= \eta^{(j)}((x^k)^T s^k + \tau^k\kappa^k) + \left(\textstyle\sum_{t=1}^{j-1}\eta^{(t)} f^{(j-t)}\right)((x^k)^T s^k + \tau^k\kappa^k) \\
&= f^{(j)}((x^k)^T s^k + \tau^k\kappa^k).
\end{aligned}$$

Similarly, we can show that (6) and (7) hold for $i = 1, \ldots, p$ and $j = p$, which completes the proof. □

Let

$$\begin{aligned}
\triangle x &= x^{k+1} - x^k &&= \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_x^{(j)} - f^{(j)} x^k)\right)/\hat{\alpha}^k, \\
\triangle y &= y^{k+1} - y^k &&= \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_y^{(j)} - f^{(j)} y^k)\right)/\hat{\alpha}^k, \\
\triangle s &= s^{k+1} - s^k &&= \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_s^{(j)} - f^{(j)} s^k)\right)/\hat{\alpha}^k, \\
\triangle \tau &= \tau^{k+1} - \tau^k &&= \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_\tau^{(j)} - f^{(j)} \tau^k)\right)/\hat{\alpha}^k, \\
\triangle \kappa &= \kappa^{k+1} - \kappa^k &&= \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_\kappa^{(j)} - f^{(j)} \kappa^k)\right)/\hat{\alpha}^k,
\end{aligned}$$

where

$$\hat{\alpha}^k = 1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}. \tag{8}$$

Then we have

$$\begin{array}{cccccc}
A\triangle x & -b\triangle\tau & & & = & \frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{\hat{\alpha}^k} r_P^k, \\
-A^T\triangle y & +c\triangle\tau & -\triangle s & & = & -\frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{\hat{\alpha}^k} r_D^k, \\
b^T\triangle y & -c^T\triangle x & & -\triangle\kappa & = & \frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{\hat{\alpha}^k} r_G^k
\end{array} \tag{9}$$

and

$$\begin{aligned}
&X^k\triangle s + S^k\triangle x \\
&= \tfrac{1}{\hat{\alpha}^k}\left\{\textstyle\sum_{j=1}^r (\alpha^k)^j (X^k d_s^{(j)} + S^k d_x^{(j)} - f^{(j)} X^k s^k - f^{(j)} S^k x^k)\right\} \\
&= \tfrac{1}{\hat{\alpha}^k}\left\{\alpha^k(\gamma\mu^k e - X^k s^k) + \textstyle\sum_{j=2}^r (\alpha^k)^j (-\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)})\right. \\
&\qquad \left. -2(\textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}) X^k s^k\right\} \\
&= \tfrac{1}{\hat{\alpha}^k}\left\{\alpha^k\gamma\mu^k e - (\alpha^k + 2\textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}) X^k s^k - \sum_{j=2}^r (\alpha^k)^j (\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)})\right\}
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\tau^k\triangle\kappa + \kappa^k\triangle\tau \\
&= \tfrac{1}{\hat{\alpha}^k}\left\{\alpha^k\gamma\mu^k - \left(\alpha^k + 2\textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)}\right)\tau^k\kappa^k - \sum_{j=2}^r (\alpha^k)^j \left(\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}\right)\right\}.
\end{aligned} \tag{11}$$

Using Lemma 2.1 and relations (10) and (11), we further have

$$
\begin{aligned}
&(x^k)^T\triangle s + (s^k)^T\triangle x + \tau^k\triangle\kappa + \kappa^k\triangle\tau \\
&= \tfrac{1}{\hat\alpha^k}\Big\{\alpha^k\gamma\mu^k(n+1) - (\alpha^k + 2\textstyle\sum_{j=1}^r(\alpha^k)^j f^{(j)})((x^k)^T s^k + \tau^k\kappa^k) \\
&\quad - \textstyle\sum_{j=2}^r(\alpha^k)^j \sum_{t=1}^{j-1}((d_x^{(t)})^T d_s^{(j-t)} + d_\tau^{(t)} d_\kappa^{(j-t)})\Big\} \\
&= \tfrac{\mu^k(n+1)}{\hat\alpha^k}\Big\{\alpha^k\gamma - \alpha^k - 2\textstyle\sum_{j=1}^r(\alpha^k)^j f^{(j)} + \sum_{j=2}^r(\alpha^k)^j \sum_{t=1}^{j-1}\eta^{(t)} f^{(j-t)}\Big\} \\
&= \tfrac{\mu^k(n+1)}{\hat\alpha^k}\Big\{\alpha^k(f^{(1)} - \eta^{(1)}) - 2\textstyle\sum_{j=1}^r(\alpha^k)^j f^{(j)} + \sum_{j=2}^r(\alpha^k)^j(f^{(j)} - \eta^{(j)})\Big\} \\
&= \tfrac{\mu^k(n+1)}{\hat\alpha^k}\Big\{-2\textstyle\sum_{j=1}^r(\alpha^k)^j f^{(j)} + \sum_{j=1}^r(\alpha^k)^j(f^{(j)} - \eta^{(j)})\Big\} \\
&= \tfrac{\mu^k(n+1)}{\hat\alpha^k}\Big\{-\textstyle\sum_{j=1}^r(\alpha^k)^j(f^{(j)} + \eta^{(j)})\Big\},
\end{aligned}
\tag{12}
$$

which leads to the next lemma and proposition.

LEMMA 2.2. *We have*

$$\triangle x^T\triangle s + \triangle\tau\triangle\kappa = 0.$$

Proof. From equations (9) and (12), we derive

$$
\begin{aligned}
&\triangle x^T\triangle s + \triangle\tau\triangle\kappa \\
&= -\frac{\sum_{j=1}^r(\alpha^k)^j(\eta^{(j)} + f^{(j)})}{\hat\alpha^k}\Big\{\frac{\sum_{j=1}^r(\alpha^k)^j(\eta^{(j)} + f^{(j)})}{1 + \sum_{j=1}^r(\alpha^k)^j f^{(j)}}((x^k)^T z^k + \tau^k\kappa^k) \\
&\quad + (x^k)^T\triangle s + (s^k)^T\triangle x + \tau^k\triangle\kappa + \kappa^k\triangle\tau\Big\} \\
&= 0. \qquad \square
\end{aligned}
$$

PROPOSITION 2.3. *The generic algorithm generates* $\{y^k, x^k, \tau^k, s^k, \kappa^k\}$ *and* $\{\mu^k, \theta^k\}$ *satisfying*

$$\mu^0 = [(x^0)^T s^0 + \tau^0\kappa^0] \;/\; (n+1),$$

$$\mu^{k+1} = \frac{1 - \sum_{j=1}^r(\alpha^k)^j\eta^{(j)}}{1 + \sum_{j=1}^r(\alpha^k)^j f^{(j)}}\mu^k,$$

and

$$\theta^0 = 1, \quad \theta^{k+1} = \frac{1 - \sum_{j=1}^r(\alpha^k)^j\eta^{(j)}}{1 + \sum_{j=1}^r(\alpha^k)^j f^{(j)}}\theta^k,$$

with

$$r_P^k = \theta^k\, r_P^0, \quad r_D^k = \theta^k\, r_D^0, \quad r_G^k = \theta^k\, r_G^0.$$

Proposition 2.3 states that the high-order algorithm possesses the property

$$\frac{\mu^k}{\theta^k} = \mu^0, \quad k \geq 1,$$

as does the HSD algorithm of Ye, Todd, and Mizuno [26]. As we discussed earlier, every limit point of iterates is a strictly complementary solution for (HLF).

3. Special case: Algorithm 1. We now analyze a special case where

$$\eta^{(1)} = 1 - \gamma \quad \text{and} \quad \eta^{(j)} = 0 \quad \forall \quad j \geq 2.$$

Note that $f^{(j)} = 0 \; \forall \; j \geq 1$, in this case, and the algorithm becomes Algorithm 1.

ALGORITHM 1.

1. Solve for the first-order direction from

$$\begin{array}{rrrrrcl} & A\, d_x^{(1)} & -b\, d_\tau^{(1)} & & & = & (1-\gamma) r_P^k, \\ -A^T\, d_y^{(1)} & & +c\, d_\tau^{(1)} & -d_s^{(1)} & & = & -(1-\gamma) r_D^k, \\ b^T\, d_y^{(1)} & -c^T\, d_x^{(1)} & & & -d_\kappa^{(1)} & = & (1-\gamma) r_G^k, \end{array} \tag{13}$$

and

$$\begin{array}{rcl} X^k d_s^{(1)} + S^k d_x^{(1)} & = & \gamma \mu^k e - X^k s^k, \\ \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)} & = & \gamma \mu^k - \tau^k \kappa^k. \end{array} \tag{14}$$

2. For $j = 2, 3, \ldots, r$, solve for the jth-order direction from

$$\begin{array}{rrrrrcl} & A\, d_x^{(j)} & -b\, d_\tau^{(j)} & & & = & 0, \\ -A^T\, d_y^{(j)} & & +c\, d_\tau^{(j)} & -d_s^{(j)} & & = & 0, \\ b^T\, d_y^{(j)} & -c^T\, d_x^{(j)} & & & -d_\kappa^{(j)} & = & 0, \end{array} \tag{15}$$

and

$$\begin{array}{rcl} X^k d_s^{(j)} + S^k d_x^{(j)} & = & -\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}, \\ \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)} & = & -\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}. \end{array} \tag{16}$$

3. Choose a step size α^k such that the new iterate

$$\begin{array}{rcl} x^{k+1} & = & x^k + \sum_{j=1}^r (\alpha^k)^j d_x^{(j)}, \\ y^{k+1} & = & y^k + \sum_{j=1}^r (\alpha^k)^j d_y^{(j)}, \\ s^{k+1} & = & s^k + \sum_{j=1}^r (\alpha^k)^j d_s^{(j)}, \\ \tau^{k+1} & = & \tau^k + \sum_{j=1}^r (\alpha^k)^j d_\tau^{(j)}, \\ \kappa^{k+1} & = & \kappa^k + \sum_{j=1}^r (\alpha^k)^j d_\kappa^{(j)} \end{array}$$

lies in $\mathcal{N}_\infty^\sim(\beta)$. □

Note that Algorithm 1, with $r = 1$, is exactly the HSD algorithm described in Ye, Todd, and Mizuno [26] and Xu, Hung, and Ye [25]. In the rest of this section and in the next section, the notation

$$x = \begin{pmatrix} x \\ \tau \end{pmatrix} \in \mathbf{R}^{n+1}, \quad s = \begin{pmatrix} s \\ \kappa \end{pmatrix} \in \mathbf{R}^{n+1},$$

$$d_x^{(j)} = \begin{pmatrix} d_x^{(j)} \\ d_\tau^{(j)} \end{pmatrix} \in \mathbf{R}^{n+1}, \quad \text{and} \quad d_s^{(j)} = \begin{pmatrix} d_s^{(j)} \\ d_\kappa^{(j)} \end{pmatrix} \in \mathbf{R}^{n+1}$$

will be introduced. Note that for Algorithm 1, Lemma 2.1 becomes

$$(17) \quad (d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} = 0 \quad \text{and} \quad (d_x^{(j)})^T d_s^{(j)} = 0 \quad \forall \quad i, j = 1, \ldots, r.$$

Again, let

$$\begin{array}{rcccl} \triangle x & = & x^{k+1} - x^k & = & \sum_{j=1}^r (\alpha^k)^j d_x^{(j)}, \\ \triangle y & = & y^{k+1} - y^k & = & \sum_{j=1}^r (\alpha^k)^j d_y^{(j)}, \\ \triangle s & = & s^{k+1} - s^k & = & \sum_{j=1}^r (\alpha^k)^j d_s^{(j)}, \end{array}$$

where $(\triangle x, \triangle s, \triangle y)$ satisfy

$$X^k \triangle s + S^k \triangle x \quad = \quad \alpha^k \gamma \mu^k e - \alpha^k X^k s^k - \sum_{j=2}^r (\alpha^k)^j (\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}).$$

Thus, relation (12), Lemma 2.2, and Proposition 2.3 become

$$(x^k)^T \triangle s + (s^k)^T \triangle x = \alpha^k (\gamma - 1)(x^k)^T s^k,$$

$$\triangle x^T \triangle s = 0,$$

and

$$(18) \qquad \mu^{k+1} = [1 - \alpha^k (1 - \gamma)]\mu^k, \quad \theta^{k+1} = [1 - \alpha^k (1 - \gamma)]\theta^k.$$

By relation (18), γ needs to be less than 1 to guarantee that μ^{k+1} is always less than μ^k. The selection of α^k plays a very important role in our complexity analysis. Let us define

$$x(\alpha) = x^k + \sum_{j=1}^r \alpha^j d_x^{(j)}, \quad s(\alpha) = s^k + \sum_{j=1}^r \alpha^j d_s^{(j)},$$

and

$$(19) \qquad \begin{array}{rcl} g(\alpha) & = & \min_{i=1,\ldots,(n+1)} \{x_i(\alpha) s_i(\alpha)\} - \beta \frac{x(\alpha)^T s(\alpha)}{n+1} \\ & = & \min_{i=1,\ldots,(n+1)} \{x_i(\alpha) s_i(\alpha)\} - \beta [1 - \alpha(1 - \gamma)]\mu^k. \end{array}$$

We can choose α^k as

$$(20) \qquad \alpha^k = \max_{\alpha \in (0, \frac{1}{1-\gamma})} \{\alpha : \ g(\zeta) \geq 0 \ \forall \ \zeta \leq \alpha\}.$$

This choice of α^k minimizes $x(\alpha)^T s(\alpha)$ while it guarantees that the new iterate still lies in $\mathcal{N}_\infty^\sim(\beta)$.

In general, problem (20) involves locating roots for each of $n+1$ polynomials with degree $2r$, which with specified error is in the complexity class NC and can be solved efficiently in theory (see Pan [18]). In practice, we need to locate only an approximate step size. In the next section, we will establish a lower bound for α^k:

$$\alpha^k \geq \frac{\text{const}}{(n+1)^{\frac{r+1}{2r}}},$$

where

$$\text{const} = \frac{\beta}{4(1-\gamma)}\sqrt[r]{(1-\beta)\gamma}.$$

Thus, we need only to compute an approximate step size, $\bar{\alpha}$, such that

$$(x(\bar{\alpha}), s(\bar{\alpha})) \in \mathcal{N}_\infty^{\sim}(\beta)$$

and

$$\alpha^k - \bar{\alpha} \leq \frac{.001\text{const}}{n+1} \leq 0.001\alpha^k;$$

that is, $\bar{\alpha}$ will be at least a fraction, .999, of the exact step size α^k, and it approaches above α^k as $n \to \infty$.

We may compute such an approximate step size using the bisection technique. We know that the step size must be in

$$\left[\frac{\text{const}}{(n+1)^{\frac{r+1}{2r}}}, \frac{1}{1-\gamma}\right].$$

Thus, we take α as the middle point of this candidate interval. Compute α^j, $j = 1, \ldots, r$; this will use r arithmetic operations. Then compute $x(\alpha)$, $s(\alpha)$, and $g(\alpha)$, which is done basically by multiplication of two $n \times r$ matrices with the vector α^j. This arithmetic operation cost is of order $O(nr)$. If $x(\alpha) > 0$, $s(\alpha) > 0$, and $g(\alpha) \geq 0$, then set the new candidate interval as

$$\left[\alpha, \frac{1}{1-\gamma}\right];$$

otherwise, set it as

$$\left[\frac{\text{const}}{n^{\frac{r+1}{2r}}}, \alpha\right].$$

Continue this bisection process until the length of the candidate interval is less than $.001\text{const}/(n+1)$. Obviously, the total number of operations of this process is of order $nr(\log n)$. Even when $r = n$, the cost $n^2(\log n)$ is well below n^3.

4. Complexity analysis of Algorithm 1. In this section, we first develop several technical results which are needed to establish our main complexity result. Let $D = (S^k)^{1/2}(X^k)^{-1/2}$. Then relation (17) implies that, for $j \geq 1$,

$$\|D^{-1}d_s^{(j)}\|^2 + \|Dd_x^{(j)}\|^2 = \|D^{-1}d_s^{(j)} + Dd_x^{(j)}\|^2.$$

Define $q^{(j)} = D^{-1}d_s^{(j)} + Dd_x^{(j)}$, $j \geq 1$; then

$$\max(\|Dd_x^{(j)}\|, \|D^{-1}d_s^{(j)}\|) \leq \|q^{(j)}\|. \tag{21}$$

Moreover, relations (14) and (16) show that

$$q^{(j)} = \begin{cases} (X^kS^k)^{-\frac{1}{2}}(\gamma\mu^k e - X^k s^k) & \text{if } j = 1, \\ (X^kS^k)^{-\frac{1}{2}}(-\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}) & \text{if } j \geq 2. \end{cases} \tag{22}$$

And it is easy to verify the following lemma (e.g., see Mizuno, Todd, and Ye [15]).

LEMMA 4.1. *We have*

$$\|q^{(1)}\|^2 \leq \delta (x^k)^T s^k, \quad \text{where} \quad \delta = \gamma^2/\beta - 2\gamma + 1.$$

Relations (21) and (22) are now used to prove the following lemma.

LEMMA 4.2. *For* $j = 1, 2, \ldots, r$,

$$\|q^{(j)}\| \leq \frac{p(j)}{(\beta\mu^k)^{\frac{j-1}{2}}} \|q^{(1)}\|^j, \tag{23}$$

where the sequence $p(j)$ *is defined recursively as follows:*

$$\begin{aligned} p(1) &= 1, \\ p(j) &= \textstyle\sum_{t=1}^{j-1} p(t)p(j-t), \quad j = 2, \ldots, r. \end{aligned}$$

Proof. The proof is by induction on j. Note that (23) holds for $j = 1$. Assume (23) holds for all t with $1 \leq t < j$. We will show that it holds for j.

$$\begin{aligned} \|q^{(j)}\| &= \|(X^k S^k)^{-\frac{1}{2}} (-\textstyle\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)})\| \\ &\leq \|(X^k S^k)^{-\frac{1}{2}}\| \|(-\textstyle\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)})\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}} \textstyle\sum_{t=1}^{j-1} \|D d_x^{(t)} D^{-1} d_s^{(j-t)}\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}} \textstyle\sum_{t=1}^{j-1} \|D d_x^{(t)}\| \|D^{-1} d_s^{(j-t)}\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}} \textstyle\sum_{t=1}^{j-1} \|q^{(t)}\| \|q^{(j-t)}\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}} \textstyle\sum_{t=1}^{j-1} \left(\frac{p(t)}{(\beta\mu^k)^{\frac{t-1}{2}}} \|q^{(1)}\|^t\right)\left(\frac{p(j-t)}{(\beta\mu^k)^{\frac{j-t-1}{2}}} \|q^{(1)}\|^{j-t}\right) \\ &= \frac{1}{(\beta\mu^k)^{\frac{j-1}{2}}} \{\textstyle\sum_{t=1}^{j-1} p(t)p(j-t)\} \|q^{(1)}\|^j \\ &= \frac{p(j)}{(\beta\mu^k)^{\frac{j-1}{2}}} \|q^{(1)}\|^j, \end{aligned}$$

which proves the lemma. □

Next, we establish a lower bound for α^k.

LEMMA 4.3. *Let* α^k *be defined in* (20); *then*

$$\alpha^k \geq \frac{\xi^k}{(n+1)^{\frac{r+1}{2r}}},$$

where

$$\xi^k = \min_{j=r+1,\ldots,2r} \left(\sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma}{rp(j)\delta^{\frac{j}{2}}}} \right).$$

Proof. From the definition of α^k and $g(\alpha)$, α^k is the largest number in $(0, \frac{1}{1-\gamma})$ such that for $\alpha \in [0, \alpha^k]$,

$$x_i(\alpha)s_i(\alpha) - \beta[x(\alpha)^T s(\alpha)]/(n+1) \geq 0 \quad \forall \quad i = 1, 2, \ldots, n+1.$$

Using relation (21), Lemmas 4.1 and 4.2, and the fact that $x_i^k s_i^k - \beta\mu^k \geq 0$, we derive that, $\forall\ i = 1, \ldots, n+1$,

$$
\begin{aligned}
& x_i(\alpha)s_i(\alpha) - \beta[x(\alpha)^T s(\alpha)]/(n+1) \\
&= (1-\alpha)(x_i^k s_i^k - \beta\mu^k) + (1-\beta)\alpha\gamma\mu^k + \sum_{j=r+1}^{2r} \left\{ \alpha^j \sum_{t=j-r}^{r} (d_x^{(t)})_i (d_s^{(j-t)})_i \right\} \\
&\geq (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ \alpha^j \sum_{t=j-r}^{r} \|q^{(t)}\| \|q^{(j-t)}\| \right\} \\
&\geq (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ \alpha^j \frac{\|q^{(1)}\|^j}{(\beta\mu^k)^{\frac{j-2}{2}}} \sum_{t=j-r}^{r} p(t)p(j-t) \right\} \\
&\geq (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ \alpha^j \frac{\|q^{(1)}\|^j}{(\beta\mu^k)^{\frac{j-2}{2}}} \sum_{t=1}^{j-1} p(t)p(j-t) \right\} \\
&\geq (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ (n+1)^{\frac{j}{2}} \mu^k \alpha^j \frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j) \right\} \\
&= \alpha\mu^k \left\{ (1-\beta)\gamma - \sum_{j=r+1}^{2r} \frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j)(n+1)^{\frac{j}{2}} \alpha^{j-1} \right\} \\
&= \alpha\mu^k h(\alpha),
\end{aligned}
$$

where

$$
h(\alpha) = (1-\beta)\gamma - \sum_{j=r+1}^{2r} \frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j)(n+1)^{\frac{j}{2}} \alpha^{j-1}.
$$

As a result, for $\alpha > 0$, it is clear that $g(\alpha) \geq 0$ if $h(\alpha) \geq 0$.

By substitution, we see that $h(\alpha) \geq 0$ for $\alpha \in [0, \xi^k/(n+1)^{\frac{r+1}{2r}}]$, since for $j = r+1, \ldots, 2r$ each

$$
\frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j)(n+1)^{\frac{j}{2}} \alpha^{j-1} \leq (n+1)^{\frac{r+1-j}{2r}} \frac{(1-\beta)\gamma}{r} \leq \frac{(1-\beta)\gamma}{r}.
$$

Hence

$$
\alpha^k \geq \xi^k/(n+1)^{\frac{r+1}{2r}}. \qquad \square
$$

Finally, we need to check whether or not ξ^k can be bounded from below. Recall that $\delta = \gamma^2/\beta - 2\gamma + 1$. Assume that $\gamma \leq 1/2$. Then we have

$$
\delta \leq \gamma^2/\beta + (1-2\gamma)/\beta = (1-\gamma)^2/\beta.
$$

Also recall that the solution for $p(j)$ in Lemma 4.2 is given by

$$
p(j) = \frac{1}{j} \begin{pmatrix} 2(j-1) \\ j-1 \end{pmatrix} \leq \frac{1}{j} 2^{2j-2}.
$$

Thus, for $j = r+1, \ldots, 2r$, we have

$$\sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma}{rp(j)\delta^{\frac{j}{2}}}} \geq \sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma\beta^{\frac{j}{2}}}{rp(j)(1-\gamma)^j}} \geq \frac{\beta}{1-\gamma}\sqrt[j-1]{\frac{(1-\beta)\gamma}{rp(j)(1-\gamma)}}$$
$$\geq \frac{\beta}{1-\gamma}\sqrt[j-1]{\frac{(1-\beta)\gamma}{\frac{r}{j}2^{2j-2}}} \geq \frac{\beta}{4(1-\gamma)}\sqrt[j-1]{\frac{j(1-\beta)\gamma}{r}}$$
$$\geq \frac{\beta}{4(1-\gamma)}\sqrt[j-1]{(1-\beta)\gamma} \geq \frac{\beta}{4(1-\gamma)}\sqrt[r]{(1-\beta)\gamma}.$$

Therefore,

$$\xi^k \geq \frac{\beta}{4(1-\gamma)}\sqrt[r]{(1-\beta)\gamma}. \tag{24}$$

We now present our main complexity result.

THEOREM 4.4. *Given any (infeasible) initial point in $\mathcal{N}_\infty^\sim(\beta)$ for any constant $\beta \in (0,1)$, Algorithm 1, with any constant $0 < \gamma < 1$, generates a strictly complementary feasible solution for (HLF) in $O(n^{\frac{r+1}{2r}}L)$ iterations where $r \in [1,n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

Proof. We easily see that when $1 \leq r \leq n$, the total arithmetic operation in each iteration is $O(n^3)$, since the linear equations in all corrections have the same left-hand matrix. Furthermore, inequality (24) implies that

$$\mu^{k+1} = \left(1-\alpha^k(1-\gamma)\right)\mu^k \leq \left(1-\frac{\xi^k(1-\gamma)}{n^{\frac{r+1}{2r}}}\right)\mu^k \leq \left(1-\frac{\beta\sqrt[r]{(1-\beta)\gamma}}{4(n+1)^{\frac{r+1}{2r}}}\right)\mu^k,$$

which, together with Proposition 2.3, proves the theorem. □

As we can see, if $r = n$ and n increases, the iteration complexity of Algorithm 1 tends to $O(\sqrt{n}L)$ asymptotically. Furthermore, a popular choice for γ in practice is not a constant but $\gamma = O(1/n)$ (e.g., [10]). Interestingly, the asymptotical iteration complexity of Algorithm 1 for such a choice of γ is still $O(\sqrt{n}L)$. More precisely, we have Corollary 4.5.

COROLLARY 4.5. *Given any (infeasible) initial point in $\mathcal{N}_\infty^\sim(\beta)$ for any constant $\beta \in (0,1)$, Algorithm 1, with $\gamma = 1/(n+1)$, generates a strictly complementary feasible solution for (HLF) in $O(n^{\frac{r+3}{2r}}L)$ iterations where $r \in [1,n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

5. Algorithm 2 and its complexity. Another special case is to set

$$\eta^{(j)} = 0 \quad \forall \quad j \geq 1.$$

Therefore $f^{(1)} = \gamma - 1$, $f^{(j)} = 0 \ \forall\ j \geq 2$, and the algorithm becomes the following.

ALGORITHM 2.

1. Solve for the first-order direction from

$$\begin{array}{rrrrrcl} & A\,d_x^{(1)} & -b\,d_\tau^{(1)} & & & = & 0, \\ -A^T\,d_y^{(1)} & & +c\,d_\tau^{(1)} & -d_s^{(1)} & & = & 0, \\ b^T\,d_y^{(1)} & -c^T\,d_x^{(1)} & & & -d_\kappa^{(1)} & = & 0, \end{array}$$

and

$$\begin{aligned} X^k d_s^{(1)} + S^k d_x^{(1)} &= \gamma\mu^k e - X^k s^k, \\ \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)} &= \gamma\mu^k - \tau^k \kappa^k. \end{aligned}$$

2. For $j = 2, 3, \ldots, l$, solve for the jth-order direction from

$$\begin{array}{rrrrrcl} & A\, d_x^{(j)} & -b\, d_\tau^{(j)} & & & = & 0, \\ -A^T\, d_y^{(j)} & & +c\, d_\tau^{(j)} & -d_s^{(j)} & & = & 0, \\ b^T\, d_y^{(j)} & -c^T\, d_x^{(j)} & & & -d_\kappa^{(j)} & = & 0, \end{array}$$

and

$$\begin{aligned} X^k d_s^{(j)} + S^k d_x^{(j)} &= -\textstyle\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}, \\ \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)} &= -\textstyle\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}. \end{aligned}$$

3. Choose a step size α^k such that the new iterate,

$$\begin{aligned} x^{k+1} &= \left(x^k + \textstyle\sum_{j=1}^{r} (\alpha^k)^j d_x^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ y^{k+1} &= \left(y^k + \textstyle\sum_{j=1}^{r} (\alpha^k)^j d_y^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ s^{k+1} &= \left(s^k + \textstyle\sum_{j=1}^{r} (\alpha^k)^j d_s^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ \tau^{k+1} &= \left(\tau^k + \textstyle\sum_{j=1}^{r} (\alpha^k)^j d_\tau^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ \kappa^{k+1} &= \left(\kappa^k + \textstyle\sum_{j=1}^{r} (\alpha^k)^j d_\kappa^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right) \end{aligned}$$

lies in $\mathcal{N}_\infty^{\sim}(\beta)$.

As discussed in Xu and Ye [24], a practical, useful feature of Algorithm 2 is that it does not need to compute the infeasibility residuals at all. In this case, we again have

$$(d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} + d_\tau^{(i)} d_\kappa^{(j)} + d_\tau^{(j)} d_\kappa^{(i)} = 0 \quad \text{and} \quad (d_x^{(j)})^T d_s^{(j)} + d_\tau^{(j)} d_\kappa^{(j)} = 0$$

for all $i, j = 1, \ldots, r$. Relation (12), Lemma 2.2, and Proposition 2.3 become

$$(x^k)^T \triangle s + (s^k)^T \triangle x + \tau^k \triangle\kappa + \kappa^k \triangle\tau = \frac{-\alpha^k(\gamma - 1)}{1 + \alpha^k(\gamma - 1)}((x^k)^T s^k + \kappa^k \tau^k),$$

$$\triangle x^T \triangle s + \triangle\tau \triangle\kappa = 0,$$

$$\mu^{k+1} = \frac{1}{1 + \alpha^k(\gamma - 1)}\mu^k, \quad \theta^{k+1} = \frac{1}{1 + \alpha^k(\gamma - 1)}\theta^k. \tag{25}$$

From relation (25), that γ is greater than 1 implies that μ^{k+1} is less than μ^k. Similarly, we choose the step size α^k as

$$\alpha^k = \max_{\alpha > 0}\{\alpha : g(\zeta) \geq 0 \forall\ \zeta \leq \alpha\},$$

where g is defined as in (19).

One can verify that all lemmas in the last two sections hold for Algorithm 2. The only difference is that $\delta \leq \frac{\gamma^2}{\beta}$, since $\gamma > 1$. Therefore, for $j = r+1, \ldots, 2r$,

$$\sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma}{rp(j)\delta^{\frac{j}{2}}}} \geq \sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\beta^{\frac{j}{2}}\gamma}{rp(j)\gamma^j}} \geq \frac{\beta}{4\gamma}\sqrt[j-1]{\frac{(1-\beta)}{rp(j)}} \geq \frac{\beta}{4\gamma}\sqrt[r]{1-\beta},$$

which implies that $\xi^k \geq \frac{\beta}{4\gamma}\sqrt[r]{1-\beta}$. Thus, we have

$$\mu^{k+1} = \frac{1}{1+\alpha^k(\gamma-1)}\mu^k \leq \left(1 - \frac{\beta(\gamma-1)\sqrt[r]{1-\beta}}{4\gamma(n+1)^{\frac{r+1}{2r}} + \beta(\gamma-1)\sqrt[r]{1-\beta}}\right)\mu^k.$$

Therefore, the complexity result for Algorithm 2 can be presented as follows.

THEOREM 5.1. *Given any (infeasible) initial point in $\mathcal{N}_\infty^\sim(\beta)$ for any constant $\beta \in (0,1)$, Algorithm 2, with any $\gamma > 1$, generates a strictly complementary feasible solution for (HLF) in $O(n^{\frac{r+1}{2r}}L)$ iterations where $r \in [1,n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

6. Conclusion. A number of implications and points can be drawn from our main result.

(i) The high-order power-series method, where iterative points move along a high-order polynomial curve, has been used in practice and partially analyzed in theory. Our result indicates that the use of this method also significantly improves the worst-case iteration complexity. Our result provides a further theoretical base for using this approach.

(ii) From our theoretical result, the order r of the power-series method has a diminishing role in improving the worst-case complexity result. Thus, we probably expect only the first few order steps really make a difference in algorithm performance. This seems to be what is observed in practice.

(iii) Our result also provides a justification for using the wider neighborhood $\mathcal{N}_\infty^-$ or $\mathcal{N}_\infty^\sim$, coupled with a high-order method. The theoretical complexity based on wider neighborhoods is not much worse than that based on smaller neighborhoods. We hope this is a significant step to bridge the gap between theory and practice.

(iv) Our result also indicates how insensitive of the value of γ the centering weight is in high-order power-series methods. Virtually, γ can be set to any number in $[1/n^{\text{const}}, 1)$ for any positive constant, const, if iterative points move along a high-order polynomial curve. This implies that the method has a sufficient self-centering function even if γ is close to 0. Note that when $\gamma = 0$, the algorithm becomes the primal-dual affine scaling method.

Acknowledgments. The first author wishes to thank Xiaojie Xu for his comments and suggestions on this topic.

REFERENCES

[1] K. M. ANSTREICHER AND R. A. BOSCH, *Long steps in an $O(n^3L)$ algorithm for linear programming*, Math. Programming, 54 (1992), pp. 251–267.

[2] D. A. BAYER AND J. C. LAGARIAS, *The nonlinear geometry of linear programming:* I. *Affine and projective scaling trajectories,* II. *Legendre transform coordinates and central trajectories*, Trans. Amer. Math. Soc., 314 (1989), pp. 499–581.

[3] T. J. CARPENTER, I. J. LUSTIG, J. M. MULVEY, AND D. F. SHANNO, *Higher-order predictor-corrector interior point methods with application to quadratic objectives*, SIAM J. Optim., 3 (1993), pp. 696–725.

[4] A. J. GOLDMAN AND A. W. TUCKER, *Polyhedral convex cones*, in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 19–40.

[5] C. C. GONZAGA, *An algorithm for solving linear programming programs in $O(n^3L)$ operations*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1988, pp. 1–28.

[6] O. GÜLER AND Y. YE, *Convergence behavior of interior-point algorithms*, Math. Programming, 60 (1993), pp. 215–228.

[7] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[8] M. KOJIMA, S. MIZUNO, AND A. YOSHISE, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1988, pp. 29–47.

[9] ———, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1–26.

[10] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[11] N. MEGIDDO, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1988, pp. 131–158.

[12] S. MEHROTRA, *On the implementation of a (primal-dual) interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[13] R. C. MONTEIRO AND I. ADLER, *Interior path following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–42.

[14] R. C. MONTEIRO, I. ADLER, AND M. G. C. RESENDE, *A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension*, Math. Oper. Res., 15 (1990), pp. 191–214.

[15] S. MIZUNO, M. J. TODD, AND Y. YE, *On adaptive-step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., 18 (1993), pp. 964–981.

[16] ———, *A surface of analytic centers and infeasible-interior-point algorithms for linear programming*, Math. Oper. Res., 20 (1995), pp. 135–162.

[17] YU. NESTEROV, *Long steps strategies in interior-point potential reduction methods*, Working paper, Central Economical and Mathematical Institute, Russian Academy of Science, Moscow, Russia, 1993.

[18] V. PAN, *Complexity of computations with matrices and polynomials*, SIAM Rev., 34 (1992), pp. 225–262.

[19] F. A. POTRA, *A quadratically convergent predictor-corrector method for solving linear programs from infeasible starting points*, Math. Programming, 67 (1994), pp. 383–406.

[20] C. ROOS AND J.-PH. VIAL, *Long steps with the logarithmic penalty barrier function in linear programming*, in Economic Decision-Making: Games, Econometrics & Optimization, J. Gabszeywicz, J. F. Richard, and L. Wolsey, eds., Elsevier Science Publisher B. V., Amsterdam, 1989, pp. 433–441.

[21] J. RENEGAR, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–94.

[22] G. SONNEVEND, *An analytical center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in Lecture Notes in Control and Information Sciences, Vol. 84, Springer-Verlag, New York, 1985, pp. 866–876.

[23] R. A. TAPIA, Y. ZHANG, M. SALTZMAN, AND A. WEISER, *The Predictor-Corrector Interior-Point Method as a Composite Newton Method*, Tech. report TR 90-17, Department of Mathematical Sciences, Rice University, Houston, TX, 1990.

[24] X. XU AND Y. YE, *A Generalized Homogeneous and Self-Dual Linear Programming Algorithm*, Working paper, College of Business Administration, The University of Iowa, Iowa City, IA, 1993.

[25] X. XU, P.-F. HUNG, AND Y. YE, *A Simplified Homogeneous and Self-Dual Linear Programming Algorithm and its Implementation*, Working paper, College of Business Administration, The University of Iowa, Iowa City, IA, 1993.

[26] Y. YE, M. J. TODD, AND S. MIZUNO, *An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm*, Math. Oper. Res., 19 (1994), pp. 53–67.

[27] Y. ZHANG AND D. ZHANG, *On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms*, Math. Programming, 68 (1995), pp. 303–318.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 587–599, August 1996
© 1996 Society for Industrial and Applied Mathematics
003

TWO INFEASIBLE INTERIOR-POINT PREDICTOR-CORRECTOR ALGORITHMS FOR LINEAR PROGRAMMING *

JIANMING MIAO†

Abstract. Two new infeasible interior-point predictor-corrector algorithms for linear programming are proposed; one uses a Newton direction and the other uses a Euler direction for the predictor step. They require the solutions of two linear systems at each iteration. The algorithms are globally convergent, and ε-feasibility and ε-complementarity can be obtained in $O(n \ln (\frac{1}{\varepsilon}))$ iterations with proper initialization. The first algorithm is also quadratically convergent under the assumption that an optimal solution exists.

Key words. interior-point algorithm, linear programming

AMS subject classification. 90

1. Introduction.

Mizuno, Todd, and Ye [7] considered a predictor-corrector algorithm for linear programming. In their algorithm the predictor step is a Newton step and the corrector step is a centered Newton step. They established linear convergence of the duality gap sequence to zero and a superior iteration complexity of $O(\sqrt{n}L)$ for their algorithm. In [13] and [14], Ye et al. proposed a large step modification of the Mizuno–Todd–Ye algorithm; they demonstrated that the algorithm is quadratically convergent while maintaining the $O(\sqrt{n}L)$-iteration complexity. The algorithm, like the vast majority of interior-point methods, assumes that a starting point satisfying exactly the linear constraints and lying in the interior of the region defined by the inequality constraints is given. However, in practice it is sometimes very difficult to obtain such a starting point, and many efficient implementations of interior-point methods, like OB1 [4], use starting points that are in the interior of the region defined by the inequality constraints but do not necessarily satisfy the equality constraints.

Recently Kojima, Megiddo, and Mizuno [3] proposed a globally convergent infeasible interior-point algorithm. The first polynomial-time convergence result for an infeasible interior-point algorithm was obtained by Zhang [15]. Since then, many studies (for example, [2, 6], [8]–[12]) have been reported for linear programming and linear complementarity problems.

Potra [8] proposed an infeasible interior-point extension of the Mizuno–Todd–Ye predictor-corrector algorithm. The predictor of Potra consists of two directions, that is, one for improving "complementarity" and the other for improving "feasibility." The step lengths for the two directions were different, but they were chosen in such a way that both "complementarity" and "feasibility" are improved at the same rate. The corrector step coincides with the Mizuno–Todd–Ye corrector. It was shown [8] that the algorithm is globally convergent under the assumption that the problem has an optimal solution. Under some additional assumptions on the starting point, ε-feasibility and ε-complementarity can be obtained in $O(n \ln (\frac{1}{\varepsilon}))$ iterations. The algorithm requires two matrix factorizations and three backsolves per iteration. The disadvantages of the

* Received by the editors October 25, 1993; accepted for publication (in revised form) March 17, 1995.

† RUTCOR–Rutgers Center for Operations Research, Rutgers University, P. O. Box 5062, New Brunswick, NJ 08903-5062 (miao@rutcor.rutgers.edu).

algorithm, however, are that it requires one more backsolve than the original Mizuno–Todd–Ye algorithm and the determination of the step length requires the solution of an eighth-order polynomial equation. Besides, the superlinear convergence of the algorithm has not been proved. In [9], Potra presented another infeasible interior-point predictor-corrector algorithm that has $O(n \ln (\frac{1}{\varepsilon}))$-iteration complexity and is quadratically convergent. The algorithm requires two matrix factorizations and three backsolves per iteration; however, this time one backsolve for the predictor and two for the corrector. The step length is determined by solving five quadratic equations.

In this paper we propose two new infeasible interior-point predictor-corrector algorithms for linear programming, where both "complementarity" and "feasibility" are improved at the same rate. Algorithm 1 is a modification and extension of the large step version of the Mizuno–Todd–Ye algorithm given in [14]. Under the same assumptions as those considered in Potra [8, 9], the algorithm exhibits the same global convergence properties and polynomial complexity as those of [8, 9]. However, the algorithm requires one less backsolve, that is, two matrix factorizations and two backsolves per iteration. Moreover, it is quadratically convergent. The step length is determined by solving a quadratic equation and therefore is easy to compute. Algorithm 2 is an infeasible starting point extension of the predictor-corrector method of [1], where the predictor step is a Euler step. It demonstrates the same polynomial complexity $O(n \ln (\frac{1}{\varepsilon}))$ as Algorithm 1. We mention that Stoer [10, 11] obtained similar results by using a slightly different approach based on the properties of a perturbed central path.

The paper is organized as follows. In §2 we describe predictor-corrector Algorithm 1. The quadratic convergence and polynomial complexity of Algorithm 1 are proved in §§3 and 4. An analysis of Algorithm 2 is given in §5. Concluding remarks are presented in §6.

The notation used throughout this paper is standard; capital letters denote matrices, lower case letters denote vectors, and greek letters denote scalars. Given vectors $x, \Delta x$, the corresponding upper case symbol denotes, as usual, the diagonal matrices $X = \text{diag}\ (x), \Delta X = \text{diag}\ (\Delta x)$.

Consider a linear program in standard form

$$\begin{aligned} &\min\ c^T x \\ &\text{s.t.}\ \ Ax = b, x \geq 0, \end{aligned} \tag{1.1}$$

and its dual

$$\begin{aligned} &\max\ b^T w \\ &\text{s.t.}\ \ A^T w + y = c, y \geq 0, \end{aligned} \tag{1.2}$$

where $c, x, y \in \mathbb{R}^n, b, w \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$ is of full row rank. The KKT conditions lead to the following nonlinear system with nonnegative constraints:

$$F(x, w, y) = \begin{pmatrix} Ax - b \\ A^T w + y - c \\ XYe \end{pmatrix} = 0, \qquad (x, y) \geq 0, \tag{1.3}$$

where e is a vector of ones. It is easy to see that

$$F'(x, w, y) = \begin{pmatrix} A & O & O \\ O & A^T & I \\ Y & O & X \end{pmatrix} \tag{1.4}$$

is nonsingular for all $(x, y) > 0$.

For convenience of reference, we define the following sets:

$$\mathcal{F}_+ = \{(x, w, y) : Ax = b, A^T w + y = c, (x, y) > 0\}, \tag{1.5}$$

$$\mathcal{S} = \{(x, w, y) : F(x, w, y) = 0, (x, y) \geq 0\}, \tag{1.6}$$

$$\begin{aligned} \mathcal{S}_\varepsilon = \{(x, w, y) : x^T y \leq \varepsilon, \|Ax - b\| \leq \varepsilon, \\ \|A^T w + y - c\| \leq \varepsilon, (x, y) > 0\}. \end{aligned} \tag{1.7}$$

Clearly, $\mathcal{F}_+$ is the strictly feasible set of (1.3) and $\mathcal{S}$ the solution set for the primal-dual problem. For any given $\varepsilon > 0$, our algorithm will produce a point in the ε-approximate solution set $\mathcal{S}_\varepsilon$. Throughout the paper we assume the following.

Assumption 1. $\mathcal{S} \neq \varnothing$.

2. Predictor-corrector Algorithm 1. In this section, we describe the predictor-corrector algorithm. Consider the set of the neighborhood defined by

$$N(\alpha) = \{(x, y) | (x, y) > 0, \|Xy - \mu e\| \leq \alpha\mu\}, \tag{2.1}$$

where $\|\cdot\|$ represents the 2-norm, $\mu = x^T y/n$, and α is a constant between 0 and 1. To begin, choose a constant $0 < \alpha \leq 1/4$ (we will take $\alpha = 1/4$). Our algorithm will produce pairs (x^k, y^k) belonging to $N(\alpha)$.

ALGORITHM 1. Given (x^0, w^0, y^0) with $(x^0, y^0) \in N(\frac{1}{4})$. For $k = 0, 1, 2, \ldots$, do

Step 1. If $(x^k, w^k, y^k) \in \mathcal{S}_\varepsilon$, then stop the iteration.

Step 2. Solve the following system for the predictor direction:

$$F'(x^k, w^k, y^k) \begin{pmatrix} \Delta x^p \\ \Delta w^p \\ \Delta y^p \end{pmatrix} = -F(x^k, w^k, y^k). \tag{2.2}$$

Step 3. Choose suitable step length θ^k and let

$$(\hat{x}^k, \hat{w}^k, \hat{y}^k) = (x^k, w^k, y^k) + \theta^k(\Delta x^p, \Delta w^p, \Delta y^p). \tag{2.3}$$

Step 4. Let $\bar{\mu}^k = (1 - \theta^k)\mu^k$, where $\mu^k = \frac{1}{n}(x^k)^T y^k$. Solve the following system for the corrector direction:

$$F'(\hat{x}^k, \hat{w}^k, \hat{y}^k) \begin{pmatrix} \Delta x^c \\ \Delta w^c \\ \Delta y^c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \bar{\mu}^k e - \hat{X}^k \hat{y}^k \end{pmatrix}. \tag{2.4}$$

Step 5. Compute the new iterate

$$(x^{k+1}, w^{k+1}, y^{k+1}) = (\hat{x}^k, \hat{w}^k, \hat{y}^k) + (\Delta x^c, \Delta w^c, \Delta y^c). \tag{2.5}$$

If $(x^0, w^0, y^0) \in \mathcal{F}_+$, then $(x^k, w^k, y^k) \in \mathcal{F}_+$ for all $k \geq 0$. In this case, Algorithm 1 is a (feasible) interior-point algorithm.

Algorithm 1 is a modification and extension of the feasible predictor-corrector method of [14]. In Step 4, we use $\bar{\mu}^k$ as opposed to $\hat{\mu}^k$, which is used originally. It turns out that the use of $\bar{\mu}^k$ is crucial to extend the method of [14] to an infeasible starting point.

Let us define

$$p^k = Ax^k - b, \qquad q^k = A^T w^k + y^k - c, \tag{2.6}$$

$$\nu^0 = 1, \qquad \nu^k = \prod_{j=0}^{k-1} (1 - \theta^j). \tag{2.7}$$

Notice that

$$(\Delta x^c)^T \Delta y^c = 0; \tag{2.8}$$

from Algorithm 1 we have the following result.

LEMMA 2.1. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1. *Then*

$$p^{k+1} = (1 - \theta^k)p^k = \nu^{k+1}p^0, \qquad q^{k+1} = (1 - \theta^k)q^k = \nu^{k+1}q^0, \tag{2.9}$$

and

$$(x^{k+1})^T y^{k+1} = (1 - \theta^k)(x^k)^T y^k = \nu^{k+1}(x^0)^T y^0. \tag{2.10}$$

It follows from Lemma 2.1 that if $\theta^k = 1$ at some iteration k, then $(x^{k+1}, w^{k+1}, y^{k+1}) \in \mathcal{S}$ and Algorithm 1 can be terminated. However, this finite termination seems extremely unlikely to happen in practice. In what follows, we will assume that $\{(x^k, w^k, y^k)\}$ is an infinite sequence.

Our choice for the step length θ^k is based on the following consideration.

For a given constant $\beta > \alpha = 1/4$ (we will take $\beta = 2\alpha$), we want to choose the parameter θ^k such that the intermediate point $(\hat{x}^k, \hat{y}^k)$ satisfies

$$\|\hat{X}^k \hat{y}^k - \bar{\mu}^k e\| \leq \beta \bar{\mu}^k \tag{2.11}$$

and the next iterate satisfies

$$(x^{k+1}, y^{k+1}) \in N(\alpha) \tag{2.12}$$

so that Algorithm 1 can be repeated.

Let us define

$$x(\theta) = x^k + \theta \Delta x^p, \qquad y(\theta) = y^k + \theta \Delta y^p. \tag{2.13}$$

By our notation, clearly $x^k = x(0), y^k = y(0)$ and $\hat{x}^k = x(\theta^k), \hat{y}^k = y(\theta^k)$.

Our specific choice for θ^k is based on the following lemma.

LEMMA 2.2. *For a given constant* $0 < \beta < 1$, *if there exists a positive number* $\bar{\theta} < 1$ *such that*

$$\|X(\theta)y(\theta) - (1 - \theta)\mu^k e\| \leq \beta(1 - \theta)\mu^k \quad \textit{for all } 0 \leq \theta \leq \bar{\theta}, \tag{2.14}$$

then $(x(\bar{\theta}), y(\bar{\theta})) > 0$.

Proof. By (2.14), $X(\theta)y(\theta) \geq (1-\beta)(1-\theta)\mu^k e > 0$ for all $0 \leq \theta \leq \bar{\theta}$. This implies that $(x(\theta), y(\theta)) > 0$ for all such θ by continuity. □

Thus we choose the step length $\theta^k \leq 1$ as the largest $\bar{\theta}$ such that (2.14) is satisfied for $\beta = 2\alpha$. From now on, Algorithm 1 is always considered with such a choice of θ^k with $\alpha = 1/4$ and $\beta = 1/2$.

For future reference, we give the following lemma.

LEMMA 2.3. (see [5]). *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1. *Then*

$$\|\Delta X^c \Delta y^c\| \leq \frac{\sqrt{2}}{4} \|(\hat{X}^k \hat{Y}^k)^{-\frac{1}{2}} (\hat{X}^k \hat{y}^k - \bar{\mu}^k e)\|^2. \tag{2.15}$$

Some basic properties of the algorithm are given as follows.

LEMMA 2.4. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1. *Then for all* $k \geq 0$

(i) $\|\hat{X}^k \hat{y}^k - \bar{\mu}^k e\| \leq \frac{1}{2}\bar{\mu}^k$;

(ii) $(x^k, y^k) \in N(\frac{1}{4})$.

Proof.

(i) By the definition of $\bar{\mu}^k$ and θ^k, Lemma 2.4(i) is obvious.

(ii) For $k \geq 0$, let $(\hat{x}^k, \hat{y}^k)$ satisfy Lemma 2.4(i). From Algorithm 1, it is easily seen that

$$X^{k+1} y^{k+1} - \mu^{k+1} e = (\hat{X}^k + \Delta X^c)(\hat{y}^k + \Delta y^c) - \bar{\mu}^k e = \Delta X^c \Delta y^c.$$

Therefore, from Lemma 2.3 we have

$$\begin{aligned} \|X^{k+1} y^{k+1} - \mu^{k+1} e\| &\leq \|\Delta X^c \Delta y^c\| \\ &\leq \frac{\sqrt{2}}{4} \|(\hat{X}^k \hat{Y}^k)^{-\frac{1}{2}} (\bar{\mu}^k e - \hat{X}^k \hat{y}^k)\|^2. \end{aligned} \tag{2.16}$$

From Lemma 2.4(i), it follows that

$$\hat{x}_i^k \hat{y}_i^k \geq \frac{1}{2} \bar{\mu}^k, \qquad i = 1, 2, \ldots, n; \tag{2.17}$$

therefore

$$\|(\hat{X}^k \hat{Y}^k)^{-\frac{1}{2}} (\bar{\mu}^k e - \hat{X}^k \hat{y}^k)\|^2 \leq \frac{\frac{1}{4}(\bar{\mu}^k)^2}{\frac{1}{2}\bar{\mu}^k} = \frac{1}{2} \mu^{k+1}, \tag{2.18}$$

where the last equality follows from Lemma 2.1. Substituting (2.18) into (2.16) gives

$$\|X^{k+1} y^{k+1} - \mu^{k+1} e\| \leq \frac{\sqrt{2}}{4} \frac{1}{2} \mu^{k+1} < \frac{1}{4} \mu^{k+1}. \qquad \square$$

3. Global and quadratic convergence. In this section, we will establish the global and local behavior of Algorithm 1. We first give some technical lemmas that will be used in our convergence analysis.

LEMMA 3.1. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1. *Then*

$$\theta^k \geq \theta_1 \equiv \min \left\{ \frac{1}{2}, \sqrt{\frac{\mu^k}{8\|\Delta X^p \Delta y^p\|}} \right\}. \tag{3.1}$$

Proof. For $\theta \in [0,1]$, we have

$$\begin{aligned}\|X(\theta)y(\theta) - (1-\theta)\mu^k e\| &= \|X^k y^k + \theta(-X^k y^k) + \theta^2 \Delta X^p \Delta y^p - (1-\theta)\mu^k e\| \\ &\le (1-\theta)\|X^k y^k - \mu^k e\| + \theta^2 \|\Delta X^p \Delta y^p\| \\ &\le (1-\theta)\frac{1}{4}\mu^k + \theta^2 \|\Delta X^p \Delta y^p\|.\end{aligned}$$

For all $0 \le \theta \le \theta_1$, we have $\theta \le 1/2$ and $8\theta^2\|\Delta X^p \Delta y^p\| \le \mu^k$. Therefore, the previous inequality gives

$$\|X(\theta)y(\theta) - (1-\theta)\mu^k e\| \le \frac{1}{4}(1-\theta)\mu^k\left[1 + \frac{1}{2(1-\theta)}\right] \le \frac{1}{2}(1-\theta)\mu^k.$$

Therefore, by the definition of the step length θ^k, we have $\theta^k \ge \theta_1$, as desired. □

We will prove that the boundedness of $\|\Delta X^p \Delta y^p/\mu^k\|$ implies the linear convergence of Algorithm 1. First we give the following lemma, which was essentially proved by Potra [9].

LEMMA 3.2. *Assume $\mathcal{S} \neq \varnothing$. Then the sequence $\{z^k = (x^k, w^k, y^k)\}$ generated by Algorithm* 1 *is bounded; that is, there is a constant γ_1 (which may depend on the problem data and n) such that*

$$\|z^k\| \le \gamma_1. \tag{3.2}$$

The lemma was proved before (see, for example, [15]) under the stronger assumption that $\mathcal{F}_+ \neq \varnothing$. It should be pointed out that it is because of Lemma 2.1 that only a weaker assumption $\mathcal{S} \neq \varnothing$ is needed for the lemma.

With the help of Lemma 3.2 we can give an upper bound for $\|\Delta X^p \Delta y^p/\mu^k\|$, which was given in [16, Thm. 4.1].

LEMMA 3.3. *Let $\{(x^k, w^k, y^k)\}$ be generated by Algorithm* 1 *and $\mathcal{S} \neq \varnothing$. Then there is a constant γ_2 (which may depend on the problem data and n) such that*

$$\|\Delta X^p \Delta y^p\| \le \gamma_2[(x^k)^T y^k]^2. \tag{3.3}$$

Now we can prove the linear convergence of the algorithm.

THEOREM 3.1. *Let $\{(x^k, w^k, y^k)\}$ be generated by Algorithm* 1 *and $\mathcal{S} \neq \varnothing$. Then $\{(x^k)^T y^k\}$, $\{p^k\}$, and $\{q^k\}$ converge to zero Q-linearly.*

Proof. To prove the Q-linear convergence, by Lemma 2.1 it suffices to show that

$$\theta^k \ge \theta_0 > 0$$

for some positive number θ_0. From Lemma 3.1 and Lemma 3.3, it follows that

$$\begin{aligned}\theta^k &\ge \min\left\{\frac{1}{2}, \sqrt{\frac{1}{8n\gamma_2(x^k)^T y^k}}\right\} \\ &\ge \min\left\{\frac{1}{2}, \sqrt{\frac{1}{8n\gamma_2(x^0)^T y^0}}\right\} \equiv \theta_0 > 0.\end{aligned} \tag{3.4}$$

This completes the proof. □

From Lemma 2.1, we see that if $(1-\theta^k) \to 0$, then $(x^k)^T y^k$ converges to zero Q-superlinearly. Moreover, if

$$1-\theta^k \le \gamma (x^k)^T y^k \tag{3.5}$$

for some constant γ, then $(x^k)^T y^k$ converges to zero Q-quadratically. The following lemma gives another lower bound for θ^k which leads to (3.5).

LEMMA 3.4. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1. *Then*

$$\theta^k \ge \frac{2}{1+\sqrt{1+16\|\Delta X^p \Delta y^p/\mu^k\|}}. \tag{3.6}$$

Proof. From the proof of Lemma 3.1 we have

$$\|X(\theta)y(\theta) - (1-\theta)\mu^k e\| \le \frac{1}{4}(1-\theta)\mu^k + \theta^2 \|\Delta X^p \Delta y^p\|.$$

Thus

$$\frac{1}{4}(1-\theta)\mu^k + \theta^2 \|\Delta X^p \Delta y^p\| \le \frac{1}{2}(1-\theta)\mu^k \tag{3.7}$$

guarantees that (2.14) is satisfied for $\beta = 1/2$. Clearly, (3.7) is satisfied for all θ less than or equal to the positive root

$$\theta^+ = \frac{2}{1+\sqrt{1+16\|(\Delta X^p \Delta y^p)/\mu^k\|}}$$

of the quadratic equation (3.7). The result follows immediately. □

LEMMA 3.5. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1. *Then there is a constant* $\gamma_3 > 0$ *such that*

$$1-\theta^k \le \gamma_3 (x^k)^T y^k. \tag{3.8}$$

Proof. It follows from Lemma 3.4 that

$$\begin{aligned}
1-\theta^k &\le 1 - \frac{2}{1+\sqrt{1+16\|\Delta X^p \Delta y^p/\mu^k\|}} \\
&= \frac{\sqrt{1+16\|\Delta X^p \Delta y^p/\mu^k\|}-1}{\sqrt{1+16\|\Delta X^p \Delta y^p/\mu^k\|}+1} \\
&= \frac{16\|\Delta X^p \Delta y^p/\mu^k\|}{(\sqrt{1+16\|\Delta X^p \Delta y^p/\mu^k\|}+1)^2} \\
&\le 4\|\Delta X^p \Delta y^p/\mu^k\| \\
&\le \gamma_3 (x^k)^T y^k,
\end{aligned}$$

where the last inequality follows from Lemma 3.3 with $\gamma_3 = 4n\gamma_2$. □

Thus we have the following result.

THEOREM 3.2. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1 *and* $\mathcal{S} \neq \varnothing$. *Then* $\{(x^k)^T y^k\}, \{p^k\}$, *and* $\{q^k\}$ *converge to zero Q-quadratically.*

Proof. From Lemma 2.1 and Lemma 3.5, it follows that

$$(x^{k+1})^T y^{k+1} \leq \gamma_3 [(x^k)^T y^k]^2$$

and

$$\|p^{k+1}\| \leq \gamma_3 \frac{\mu^0}{\|p^0\|} \|p^k\|^2, \qquad \|q^{k+1}\| \leq \gamma_3 \frac{\mu^0}{\|q^0\|} \|q^k\|^2. \qquad \square$$

4. Polynomiality. In §3, we established that $(x^k)^T y^k$ converges to zero Q-linearly with a global rate $1 - \theta_0$. However, the lower bound θ_0 of θ^k (3.4) involves a constant γ_2 that may depend on the problem data and n and, thus, does not necessarily imply the polynomiality of the algorithm.

For the polynomiality of the algorithm, we specify a particular starting point. Let

$$(x^0, y^0) \in N\left(\frac{1}{4}\right), \qquad x^* \leq \rho x^0, y^* \leq \rho y^0 \tag{4.1}$$

for some $(x^*, w^*, y^*) \in \mathcal{S}$ and $\rho \geq 1$. Define the scalars ω^{f} and ω^{o} that measure feasibility and optimality of a starting point as (see [2])

(4.2)

$$\omega^{\mathrm{f}} = \min \{\max \{\|(X^0)^{-1}(x - x^0)\|_\infty, \|(Y^0)^{-1}(y - y^0)\|_\infty\} : Ax = b, A^T w + y = c\},$$

$$\omega^{\mathrm{o}} = \min \left\{ \max \left\{ \frac{(x^*)^T y^0}{(x^0)^T y^0}, \frac{(x^0)^T y^*}{(x^0)^T y^0}, 1 \right\} : (x^*, w^*, y^*) \in \mathcal{S} \right\}. \tag{4.3}$$

The next lemma provides upper bounds for ω^{f} and ω^{o}.

LEMMA 4.1 (see [2, Lem. 3.4]). *Let* (x^0, w^0, y^0) *be given by* (4.1). *Then*

$$\omega^{\mathrm{f}} \leq \rho, \qquad \omega^{\mathrm{o}} \leq \rho. \tag{4.4}$$

Let

$$D^k = (X^k)^{\frac{1}{2}} (Y^k)^{-\frac{1}{2}}. \tag{4.5}$$

Then the following lemma gives a bound for $\|(D^k)^{-1} \Delta x^p\|$ and $\|D^k \Delta y^p\|$ that is the key to polynomial-time computational complexity.

LEMMA 4.2. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1 *with* (x^0, w^0, y^0) *given by* (4.1). *Then*

$$\max \{\|(D^k)^{-1} \Delta x^p\|, \|D^k \Delta y^p\|\} \leq \zeta \sqrt{n (x^k)^T y^k}, \tag{4.6}$$

where

$$\zeta = \frac{1}{\sqrt{n}} + \omega^{\mathrm{f}} (1 + 2\omega^{\mathrm{o}}) \frac{2}{\sqrt{3}}. \tag{4.7}$$

Proof. Applying Lemma 3.3 of [2] with $\lambda_p = \lambda_d = 1, \xi = 1$ to Algorithm 1, we have

$$\|(D^k)^{-1}\Delta x^p\|, \qquad \|D^k\Delta y^p\| \le \|(X^k y^k)^{\frac{1}{2}}\| + \omega^{\mathrm{f}}(1+2\omega^{\mathrm{o}})\,\frac{(x^k)^T y^k}{\min_i\,(x_i^k y_i^k)^{1/2}}\,. \tag{4.8}$$

Since $(x^k, y^k) \in N(\frac{1}{4})$,

$$\frac{(x^k)^T y^k}{\min_i\,(x_i^k y_i^k)^{1/2}} \le \sqrt{\frac{4n(x^k)^T y^k}{3}}\,. \tag{4.9}$$

Thus we obtain

$$\begin{aligned}\|(D^k)^{-1}\Delta x^p\|, \|D^k\Delta y^p\| &\le \sqrt{(x^k)^T y^k} + \omega^{\mathrm{f}}(1+2\omega^{\mathrm{o}})\sqrt{\frac{4n(x^k)^T y^k}{3}} \\ &= \zeta\sqrt{n(x^k)^T y^k}. \qquad \square\end{aligned} \tag{4.10}$$

LEMMA 4.3. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1 *with* (x^0, w^0, y^0) *given by* (4.1). *Then there is a constant* $0 < \gamma_4 \le 1$ *independent of* n *such that*

$$\theta^k \ge \frac{\gamma_4}{n}\,. \tag{4.11}$$

Proof. From Lemmas 4.1 and 4.2 it follows that

$$\begin{aligned}\|\Delta X^p \Delta y^p\| &\le \|(D^k)^{-1}\Delta x^p\|\|D^k\Delta y^p\| \\ &\le n\zeta^2 (x^k)^T y^k \\ &\le n(\zeta^*)^2 (x^k)^T y^k,\end{aligned} \tag{4.12}$$

where

$$\zeta^* = 1 + \rho(1+2\rho)\,\frac{2}{\sqrt{3}}\,. \tag{4.13}$$

Therefore, from the previous inequality, Lemma 3.1 gives

$$\theta^k \ge \min\left\{\frac{1}{2}\,, \sqrt{\frac{1}{8(\zeta^*)^2 n^2}}\right\} = \frac{\gamma_4}{n}\,,$$

where $\gamma_4 = \frac{\sqrt{2}}{4\zeta^*}$. $\square$

By a standard argument, the polynomial complexity of the algorithm follows immediately from Lemmas 2.1 and 4.3.

THEOREM 4.1. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 1 *with* (x^0, w^0, y^0) *given by* (4.1). *For any* $\varepsilon > 0$, *the algorithm will terminate with* $(x^k, w^k, y^k) \in \mathcal{S}_\varepsilon$ *in at most* $O(nL)$ *iterations, where*

$$L = \max\{\ln((x^0)^T y^0/\varepsilon), \ln(\|Ax^0 - b\|/\varepsilon), \ln(\|A^T w^0 + y^0 - c\|/\varepsilon)\}. \tag{4.14}$$

If in addition the initial point (x^0, w^0, y^0) is feasible, then $\omega^f = 0$ and $\zeta = \frac{1}{\sqrt{n}}$. In this case

$$\theta^k \geq \min\left\{\frac{1}{2}, \sqrt{\frac{1}{8n}}\right\} = \sqrt{\frac{1}{8n}}. \tag{4.15}$$

Therefore Algorithm 1 will terminate with $(x^k)^T y^k \leq \varepsilon$ in at most $O(\sqrt{n}L)$ iterations, where

$$L = \ln\left((x^0)^T y^0/\varepsilon\right). \tag{4.16}$$

Remark 4.1. The choice of θ^k in the algorithm requires that one find the roots of a quartic polynomial; however, from Lemma 3.4 and the proofs of our main results we see that we can simply choose

$$\theta^k = \frac{2}{1 + \sqrt{1 + 16\|(\Delta X^p \Delta y^p)/\mu^k\|}},$$

and all the convergence results still hold for this modified algorithm.

5. Predictor-corrector Algorithm 2. We now consider a predictor-corrector algorithm that uses a Euler direction as a predictor.

ALGORITHM 2. Given (x^0, w^0, y^0) with $(x^0, y^0) \in N(\frac{1}{4})$. For $k = 0, 1, 2, \ldots,$ do

Step 1. If $(x^k, w^k, y^k) \in \mathcal{S}_\varepsilon$, then stop the iteration.

Step 2. Solve the following system for the predictor direction:

$$F'(x^k, w^k, y^k)\begin{pmatrix}\Delta x^p\\ \Delta w^p\\ \Delta y^p\end{pmatrix} = -\begin{pmatrix}Ax^k - b\\ A^T w^k + y^k - c\\ \mu^k e\end{pmatrix}, \tag{5.1}$$

where $\mu^k = \frac{1}{n}(x^k)^T y^k$.

Step 3. Choose suitable step length θ^k and let

$$(\hat{x}^k, \hat{w}^k, \hat{y}^k) = (x^k, w^k, y^k) + \theta^k(\Delta x^p, \Delta w^p, \Delta y^p). \tag{5.2}$$

Step 4. Let $\bar{\mu}^k = (1 - \theta^k)\mu^k$. Solve the following system for the corrector direction:

$$F'(\hat{x}^k, \hat{w}^k, \hat{y}^k)\begin{pmatrix}\Delta x^c\\ \Delta w^c\\ \Delta y^c\end{pmatrix} = \begin{pmatrix}0\\ 0\\ \bar{\mu}^k e - \hat{X}^k \hat{y}^k\end{pmatrix}. \tag{5.3}$$

Step 5. Compute the new iterate

$$(x^{k+1}, w^{k+1}, y^{k+1}) = (\hat{x}^k, \hat{w}^k, \hat{y}^k) + (\Delta x^c, \Delta w^c, \Delta y^c). \tag{5.4}$$

If the current iterate is feasible and the step length is fixed, then Algorithm 2 yields the predictor-corrector method of [1]. The algorithm is the same as Algorithm 1, except for the choice of the predictor direction. The step length $\theta^k \leq 1$ is chosen as the largest $\bar{\theta}$ such that (2.14) is satisfied for $\beta = 1/2$. It turns out that iteration sequences generated by these two algorithms have similar properties. In particular,

Lemmas 2.1–2.4 also hold for $\{(x^k, w^k, y^k)\}$ generated by Algorithm 2. Therefore, the analysis for the method in this section is quite similar to that for Algorithm 1. Analogous to Lemma 3.1, we can give a lower bound for θ^k.

LEMMA 5.1. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 2. *Then*

$$\theta^k \geq \theta_2 \equiv \min\left\{\frac{1}{4}, \sqrt{\frac{\mu^k}{8\|\Delta X^p \Delta y^p\|}}\right\}. \tag{5.5}$$

Proof. For $\theta \in [0,1]$, we have

$$\begin{aligned}\|X(\theta)y(\theta) - (1-\theta)\mu^k e\| &= \|X^k y^k - \theta\mu^k e + \theta^2 \Delta X^p \Delta y^p - (1-\theta)\mu^k e\| \\ &\leq \|X^k y^k - \mu^k e\| + \theta^2 \|\Delta X^p \Delta y^p\| \\ &\leq \frac{1}{4}\mu^k + \theta^2 \|\Delta X^p \Delta y^p\|.\end{aligned}$$

For all $0 \leq \theta \leq \theta_2$, we have $\theta \leq 1/4$ and $8\theta^2\|\Delta X^p \Delta y^p\| \leq \mu^k$. Therefore, the previous inequality gives

$$\|X(\theta)y(\theta) - (1-\theta)\mu^k e\| \leq \frac{1}{4}\mu^k + \frac{1}{8}\mu^k = \frac{3}{8}\mu^k \leq \frac{1}{2}(1-\theta)\mu^k. \qquad \square$$

LEMMA 5.2. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 2 *with* (x^0, w^0, y^0) *given by* (4.1). *Then*

$$\max\{\|(D^k)^{-1}\Delta x^p\|, \|D^k \Delta y^p\|\} \leq \eta\sqrt{n(x^k)^T y^k}, \tag{5.6}$$

where

$$\eta = \frac{2}{\sqrt{3n}} + \rho(1+2\rho)\frac{2}{\sqrt{3}}. \tag{5.7}$$

Proof. By the definition of ω^{f}, there exists an (x, w, y) such that

$$Ax = b, \qquad A^T w + y = c,$$

$$|x_i - x_i^0| \leq \omega^{\mathrm{f}} x_i^0, \qquad |y_i - y_i^0| \leq \omega^{\mathrm{f}} y_i^0.$$

Therefore,

$$\begin{pmatrix} A & O & O \\ O & A^T & I \\ Y^k & O & X^k \end{pmatrix} \begin{pmatrix} \Delta x^p + \nu^k(x^0 - x) \\ \Delta w^p + \nu^k(w^0 - w) \\ \Delta y^p + \nu^k(y^0 - y) \end{pmatrix}$$

$$= -\begin{pmatrix} 0 \\ 0 \\ \mu^k e - \nu^k Y^k(x^0 - x) - \nu^k X^k(y^0 - y) \end{pmatrix}.$$

Similar to Lemma 3.3 of [2], we can get

$$\begin{aligned}\|(D^k)^{-1}\Delta x^p\|, \qquad \|D^k \Delta y^p\| &\leq \|(V^k)^{-1}\mu^k e\| + \|(V^k)^{-1}\nu^k Y^k(x^0 - x)\| \\ &\quad + \|(V^k)^{-1}\nu^k X^k(y^0 - y)\|,\end{aligned}$$

where $V^k = (X^k Y^k)^{\frac{1}{2}}$. From Lemmas 3.1–3.4 of [2] we have

$$\|(V^k)^{-1}\nu^k Y^k(x^0 - x)\| + \|(V^k)^{-1}\nu^k X^k(y^0 - y)\| \leq \rho(1 + 2\rho)\frac{(x^k)^T y^k}{\min_i (x_i^k y_i^k)^{1/2}}. \tag{5.8}$$

Therefore, by using (4.9) we have

$$\begin{aligned} \|(D^k)^{-1}\Delta x^p\|, \|D^k \Delta y^p\| &\leq \mu^k \frac{\sqrt{n}}{\min_i (x_i^k y_i^k)^{1/2}} + \rho(1 + 2\rho)\frac{(x^k)^T y^k}{\min_i (x_i^k y_i^k)^{1/2}} \\ &\leq \sqrt{\frac{4(x^k)^T y^k}{3}} + \rho(1 + 2\rho)\sqrt{\frac{4n(x^k)^T y^k}{3}} \\ &\leq \eta\sqrt{n(x^k)^T y^k}. \qquad \square \end{aligned} \tag{5.9}$$

As a consequence of Lemmas 5.1 and 5.2 we can prove the following results. We omit the proofs here.

LEMMA 5.3. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 2 *with* (x^0, w^0, y^0) *given by* (4.1). *Then*

$$\theta^k \geq \frac{\sqrt{2}}{4\eta n}. \tag{5.10}$$

THEOREM 5.1. *Let* $\{(x^k, w^k, y^k)\}$ *be generated by Algorithm* 2 *with* (x^0, w^0, y^0) *given by* (4.1). *For any* $\varepsilon > 0$, *the algorithm will terminate with* $(x^k, w^k, y^k) \in \mathcal{S}_\varepsilon$ *in at most* $O(nL)$ *iterations, where*

$$L = \max\{\ln((x^0)^T y^0/\varepsilon), \ln(\|Ax^0 - b\|/\varepsilon), \ln(\|A^T w^0 + y^0 - c\|/\varepsilon)\}. \tag{5.11}$$

6. Concluding remarks. In this paper we proposed two infeasible predictor-corrector algorithms that have the same complexity bound of $O(nL)$. However, Algorithm 1 is in a competitive category of Newton-like methods; moreover, it is also quadratically convergent. We feel strongly that Algorithm 1 is more efficient than Algorithm 2. Past computational results do indicate the correlation between asymptotic convergence rate and algorithmic efficiency.

Acknowledgments. The author is grateful to Professor Adi Ben-Israel and Professor David Shanno for their valuable comments.

REFERENCES

[1] J. DING AND T. Y. LI, *A polynomial-time predictor-corrector algorithm for a class of linear complementarity problems*, SIAM J. Optim., 1 (1991), pp. 83–92.

[2] M. KOJIMA, *Basic Lemmas in Polynomial-Time Infeasible-Interior-Point Method for Linear Programming*, Res. Report on Information Sciences B-268, Department of Information Sciences, Tokyo Institute of Technology, Japan, 1993.

[3] M. KOJIMA, N. MEGIDDO, AND S. MIZUNO, *A Primal-Dual Exterior Point Algorithm for Linear Programming*, Res. Report RJ 8500, IBM Almaden Research Center, San Jose, CA, 1991.

[4] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[5] S. MIZUNO, *A new polynomial time method for a linear complementarity problem*, Math. Programming, 56 (1992), pp. 31–43.

[6] ———, *Polynomiality of Kojima–Megiddo–Mizuno Infeasible Interior Point Algorithm for Linear Programming*, Tech. Report, School of OR and IE, Cornell University, Ithaca, NY, May 1992; revised July 1992.

[7] S. MIZUNO, M. TODD, AND Y. YE, *On adaptive-step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., 18 (1992), pp. 964–981.

[8] F. A. POTRA, *An infeasible-interior-point predictor-corrector algorithm for linear programming*, SIAM J. Optim., 6 (1996), pp. 19–32.

[9] ———, *A quadratically convergent predictor-corrector method for solving linear programs from infeasible starting points*, Math. Programming, 67 (1994), pp. 383–406.

[10] J. STOER, *The complexity of an infeasible interior-point path-following method for the solution of linear programs*, Optim. Methods Software, 3 (1994), pp. 1–12.

[11] ———, *Infeasible interior-point methods for solving linear programs*, in Algorithms for Continuous Optimization, the State of the Art, E. Spedicato, ed., Kluwer Academic Publishers, Norwell, MA, 1994, pp. 415–434.

[12] S. WRIGHT, *A Path-Following Infeasible-Interior-Point Algorithm for Linear Complementarity Problems*, preprint MCS-P331-0991, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, October 1992.

[13] Y. YE, R. TAPIA, AND Y. ZHANG, A Superlinearly Convergent $O(\sqrt{n}L)$-iteration Algorithm for Linear Programming, Tech. Report TR91-22, Department of Mathematical Sciences, Rice University, Houston, TX, 1991.

[14] Y. YE, O. GULER, R. TAPIA, AND Y. ZHANG, *A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming*, Math. Programming, 59 (1993), pp. 151–162.

[15] Y. ZHANG, *On the convergence of an infeasible interior-point algorithm for linear programming and other problems*, SIAM J. Optim., 4 (1994), pp. 208–227.

[16] Y. ZHANG AND D. ZHANG, Superlinear Convergence of Infeasible Interior-Point Methods for Linear Programming, Tech. Report 92-15, Department of Mathematics and Statistics, University of Maryland, College Park, MD, September 1992.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 600–616, August 1996
© 1996 Society for Industrial and Applied Mathematics
004

A NEW FINITE CONTINUATION ALGORITHM FOR LINEAR PROGRAMMING*

KAJ MADSEN†, HANS BRUUN NIELSEN‡, AND MUSTAFA Ç. PINAR‡

Abstract. We describe a new finite continuation algorithm for linear programming. The dual of the linear programming problem with unit lower and upper bounds is formulated as an ℓ_1 minimization problem augmented with the addition of a linear term. This nondifferentiable problem is approximated by a smooth problem. It is shown that the minimizers of the smooth problem define a family of piecewise-linear paths as a function of a smoothing parameter. Based on this property, a finite algorithm that traces these paths to arrive at an optimal solution of the linear program is developed. The smooth problems are solved by a Newton-type algorithm. Preliminary numerical results indicate that the new algorithm is promising.

Key words. finite algorithms, continuation methods, linear programming, ℓ_1 optimization, Huber function, Newton's method

AMS subject classifications. 90C05, 65K05

1. Introduction.

This paper introduces a new finite algorithm for linear programming (LP) by taking an unconstrained view of the problem. The algorithm operates on a special form of the linear program. We consider an LP problem where the variables are restricted to be between -1 and $+1$. We call this problem a "normalized" linear program. It is well known [3, 14] that the dual of this linear program can be formulated as an unconstrained "augmented" ℓ_1 minimization problem. The term "augmented" is used in this context to indicate that the function to be minimized is composed of an ℓ_1 term and a linear term. In this paper, the ℓ_1 norm is approximated by a smooth "Huber" function [10]. The approximation involves a positive smoothing parameter γ. We also refer to this smooth problem as the "Huber" problem. The following properties of the approximation are emphasized as the main contributions of our analysis in this paper.

(P0) The minimizers of the smooth problem define a family of piecewise-linear paths as a function of the smoothing parameter γ (Lemma 3).

(P1) The solution to the LP problem is detected for sufficiently small $\gamma > 0$ using these paths. That is, γ does not have to be decreased to zero in order to obtain an exact solution to the LP problem (Theorem 1).

(P2) The structure of the set of minimizers of the smooth problem closely "mimics" the structure of the solution set of the ℓ_1 and LP problems (Theorems 1 and 4).

(P3) The nature of the approximation allows a constructive duality link to the original primal problem, thereby giving the resulting algorithm a primal-dual flavor (Theorem 4).

These results generalize the results of an earlier study on the ℓ_1 problem by the first two authors [13].

These properties suggest an algorithm to trace these piecewise-linear paths to arrive at a solution of the linear program. We refer to these paths as "solution paths" throughout the paper. We note that the solution path could be unique as is the case

* Received by the editors November 16, 1993; accepted for publication (in revised form) December 28, 1994. This research was partially supported by Danish National Science Council grant 11-0505.

† Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, 2800 Lyngby, Denmark (km@imm.dtu.dk).

‡ Department of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey.

with the concept of the central path in interior point methods. Our algorithm is best interpreted as a continuation algorithm since it possesses the following main features of continuation algorithms.

1. The solution of a parametrized family of subproblems as a parameter varies over an interval—in our case the smooth "Huber" problem as a function of the smoothing parameter γ.
2. The use of information from the solutions of the previous problems to aid in the solution of the next member of the family. Here, we utilize the piecewise-linear dependence of the solution paths on γ and the duality gap to compute an initial point for subsequent Huber problems. Using information from the previous solution makes the convergence of subsequent iterations faster.
3. A method for adjusting the continuation parameter for the efficient and accurate solution of subsequent problems. Piecewise linearity of the solution paths is again exploited in devising an efficient scheme for adjusting γ.
4. The use of a local iterative method to solve the subproblems. We use a finite Newton method [12] to solve the smooth Huber problem.

As a result of (P1), the continuation algorithm is a finite procedure provided that γ is decreased by at least a certain factor after each unconstrained minimization. We discuss these ideas more precisely in the forthcoming sections.

Our algorithm is related to many ideas in the optimization literature. In particular, there exists a close relationship between our method and penalty function algorithms since these methods can also be interpreted as continuation methods. Penalty methods usually treat all or a subset of the constraints of an optimization problem appending a penalty term to the objective function. The resulting penalty problem is solved by an iterative method, and solutions of the previous problems are used as starting points for the solution of subsequent problems. The process is controlled using a penalty parameter that is adjusted after the solution of each penalty problem if certain optimality criteria are not met. Barrier function and interior point algorithms can also be subsumed under this category, as well as more recent noninterior continuation algorithms [2]. In this regard our algorithm offers a novelty in the utilization of piecewise linearity and the absence of a penalty function.

The use of the ℓ_1 function in LP was proposed by Conn in [5] and later studied by Bartels in [1]. In these studies the ℓ_1 function is used as a penalty function. The resulting unconstrained piecewise-linear problem is solved using a combined projected descent and active set strategy.

Another related idea is due to Pinar and Zenios [19], where the Huber function is utilized as a linear-quadratic penalty function to penalize the side constraints in network models with side constraints and multicommodity network models. Linear-quadratic functions have also been the subject of study by Rockafellar [20] and Sun [21].

A more recent idea was proposed by Coleman and Li in [4, 3]. Their departure point is identical to ours in that they also work on the augmented ℓ_1 dual of a normalized linear program. They develop an affine scaling algorithm to solve the augmented ℓ_1 problem after a slight reformulation. Unfortunately, an implementation of their algorithm is not available for comparison.

To the best of our knowledge, our algorithm stands as a novel contribution to the linear optimization literature while it displays similarities with its chief competitors. In particular, the main computational burden is the solution of symmetric positive (semi)definite systems of linear equations reminiscent of interior point methods. On

the other hand, two successive systems are closely related enough to warrant an update of the previous factorization, much like the simplex method. However, the iterates are neither required to stay in the interior nor on the surface of the polytope. Furthermore, the new algorithm stops when the duality gap is closed.

We develop a preliminary, but stable and efficient, implementation of the new algorithm for problems with full matrices. The algorithm is tested both on randomly generated problems and some problems from the Netlib collection. The results indicate a modest growth in the number of iterations as the problem size increases. We compare our results with a dense simplex subroutine from the NAG subroutine library based on the package LSSOL of Stanford Systems Optimization Library. The algorithm outperforms the dense simplex subroutine by a wide margin on the randomly generated problems. It is also competitive with the simplex subroutine on most of the Netlib problems tried. We also offer some comparisons with interior point methods and in particular with the CPLEX^{TM} Barrier package which implements the primal-dual interior point algorithm [11]. Several issues that remain unresolved about the algorithm will be addressed in future studies, including the development of a sparse implementation.

The rest of the paper is organized as follows. After some preliminaries, we study the structure of the smooth Huber problem in §2. The connection between the ℓ_1 and LP problems and the smooth Huber problem is studied in §3. The results of these two sections form a constructive theory in that they suggest an algorithm. The algorithm is described in §4. Numerical results along with some implementation details are given in §5.

1.1. Preliminaries. We wish to solve the following problem:
[NormLP]

$$\begin{array}{ll} \underset{y}{\text{maximize}} & c^T y \\ \text{subject to} & \begin{array}{rcl} Ay & = & b, \\ -e \leq\ y & \leq & e, \end{array} \end{array}$$

where $c \in \Re^m$, $A \in \Re^{n \times m}$, $b \in \Re^n$, and $e = (1, \ldots, 1)$. More general problems can be transformed into [NormLP] if the bounds on y can be replaced by general upper and lower bounds. Furthermore, as we see in §5, we will also be able to solve the general LP problem via [NormLP] using the idea of slack of variables and some artificial bounds. Therefore, we limit our study to [NormLP].

The new method is based on solving the dual problem to [NormLP]:
[AL1]

$$\text{minimize } G(x) \equiv \|A^T x - c\|_1 + b^T x. \tag{1}$$

A proof of the duality correspondence can be found, for instance, in [14]. If x^* and y^* are dual solutions then

$$a_i^T x^* - c_i < 0 \Rightarrow y_i^* = 1, \tag{2}$$

$$a_i^T x^* - c_i > 0 \Rightarrow y_i^* = -1, \tag{3}$$

where a_i is the ith column of A. Therefore,

$$G(x^*) = -(A^T x^* - c)^T y^* + b^T x^* = c^T y^*, \tag{4}$$

where the last equality is the duality equality and expresses the necessary condition for a solution of [AL1]:

$$-Ay^* + b = 0, \tag{5}$$

$$-e \leq y^* \leq e. \tag{6}$$

The method for solving [AL1] is similar to that in [13] for minimizing the linear ℓ_1 function. We estimate a minimizer of G by solving a sequence of approximating smooth problems, each of which depends on a parameter $\gamma > 0$. These problems are defined as follows. Let

$$r(x) = A^T x - c, \tag{7}$$

and for a given threshold $\gamma > 0$ define

$$\mathbf{s}_\gamma(x) = [s_{\gamma 1}(x), \ldots, s_{\gamma m}(x)] \tag{8}$$

with

$$s_{\gamma i}(x) = \begin{cases} -1 & \text{if } r_i(x) < -\gamma, \\ 0 & \text{if } |r_i(x)| \leq \gamma, \\ 1 & \text{if } r_i(x) > \gamma. \end{cases} \tag{9}$$

$\mathbf{s}_\gamma(x)$ is a *sign vector*, and in general a sign vector is any vector $\mathbf{s} \in \Re^m$ with components $s_i \in \{-1, 1, 0\}$. For any sign vector $\mathbf{s}$ we define

$$W_{\mathbf{s}} = \operatorname{diag}(w_1, \ldots, w_m), \tag{10}$$

where

$$w_i = 1 - s_i^2. \tag{11}$$

If $\mathbf{s} = \mathbf{s}_\gamma(x)$ then we also denote $W_{\mathbf{s}}$ by $W_\gamma(x)$ or W_γ if no confusion is possible.

Now, the nondifferentiable problem [AL1] is approximated by the smooth "Huber problem" [10]:
[SAL1]

$$\text{minimize } G_\gamma(x) \equiv \frac{1}{2\gamma} r^T W_\gamma r + \mathbf{s}_\gamma^T \left[r - \frac{1}{2}\gamma \mathbf{s}_\gamma \right] + b^T x, \tag{12}$$

where the argument x is dropped for notational convenience. Clearly, G_γ measures the "small" residuals ($|r_i(x)| \leq \gamma$) by their squares while the "large" residuals are measured by the ℓ_1 function. Thus, G_γ is a piecewise quadratic function, and it is continuously differentiable in $\Re^n$. We prove that when $\gamma \to 0_+$ any solution of [SAL1] is close to a solution of [AL1]. Furthermore, we show that dual solutions to [AL1] and [NormLP] can be detected directly when γ is below a certain (problem dependent) threshold $\gamma_0 > 0$. Therefore, the new algorithm consists of solving [SAL1] for a decreasing set of threshold values γ until dual solutions are detected. We summarize the algorithm as follows.

```
Compute initial γ
repeat
    compute a solution of [SAL1]
    decrease γ
until duality gap is closed.
```

Since solving [SAL1] $\gamma > 0$ is a finite procedure [12], the whole algorithm is finite provided γ is decreased at least by a certain factor in each iteration.

2. Properties of G_γ. In this section we describe some essential properties of G_γ.

We can assume without loss of generality that A has rank n and that every column a_i of A is nonzero. Otherwise, the problem could easily be reformulated to have these properties.

Clearly G_γ is composed of a finite number of quadratic functions. In each domain $D \subseteq \Re^n$, where $\mathbf{s}_\gamma(x)$ is constant and G_γ is equal to a specific quadratic function as seen from the above definition. These domains are separated by the following union of hyperplanes:

$$B_\gamma = \{x \in \Re^n | \exists\, i : |r_i(x)| = \gamma\}. \tag{13}$$

A sign vector $\mathbf{s}$ is γ-*feasible* at x if

$$\forall\, \varepsilon > 0\ \exists\, z \in \Re^n \setminus B_\gamma : \|x - z\| < \varepsilon\ \wedge\ \mathbf{s} = \mathbf{s}_\gamma(z). \tag{14}$$

If $\mathbf{s}$ is a γ-feasible sign vector at some point x, then let $Q_{\mathbf{s}}^\gamma$ be the quadratic function which equals G_γ on the subset

$$\mathcal{C}_{\mathbf{s}}^\gamma = \mathrm{cl}\{z \in \Re^n | \mathbf{s}_\gamma(z) = \mathbf{s}\}. \tag{15}$$

$\mathcal{C}_{\mathbf{s}}^\gamma$ is called a Q-*subset* of $\Re^n$. Notice that any $x \in \Re^n \setminus B_\gamma$ has exactly one corresponding Q-subset ($\mathbf{s} = \mathbf{s}_\gamma(x)$), whereas a point $x \in B_\gamma$ belongs to two or more Q-subsets. Therefore, in general, we must give a sign vector $\mathbf{s}$ in addition to x in order to specify which quadratic function we are currently considering as representative of G_γ. However, the gradient of G_γ is independent of the choice of $\mathbf{s}$.

$Q_{\mathbf{s}}$ can be defined as follows:

$$Q_{\mathbf{s}}(z) = \frac{1}{2\gamma}(z - x)^T (A W_{\mathbf{s}} A^T)(z - x) + G_\gamma'^T(x)(z - x) + G_\gamma(x). \tag{16}$$

The gradient of the function G_γ is given by

$$G_\gamma'(x) = A\left[\frac{1}{\gamma} W_{\mathbf{s}} r + \mathbf{s}\right] + b, \tag{17}$$

where $\mathbf{s}$ is a γ-feasible sign vector at x. For $x \in \Re^n \setminus B_\gamma$, the Hessian of G_γ exists and is given by

$$G_\gamma''(x) = \frac{1}{\gamma} A W_\gamma A^T. \tag{18}$$

The set of indices corresponding to "small" residuals

$$\mathcal{A}_\gamma(z) = \{i | 1 \le i \le m\ \wedge\ s_{\gamma i}(z) = 0\} \tag{19}$$

is called the γ-*active* set at z and the subspace

$$\mathcal{V}_\gamma(z) = \mathrm{span}\{a_i | i \in \mathcal{A}_\gamma(z)\} \tag{20}$$

is called the γ-*active* subspace at z. The set of minimizers of G_γ is denoted by M_γ. In [12] it is shown that, for the case $b = \mathbf{0}$, there exists a minimizer $x_\gamma \in M_\gamma$ for which $\mathcal{V}_\gamma(x_\gamma) = \Re^n$. The idea of the proof is that if $\mathcal{V}_\gamma(z_\gamma) \neq \Re^n$ for a minimizer z_γ then there exists a connected set of minimizers of the form $z_\gamma + \alpha h$, where $h \perp \mathcal{V}_\gamma(z_\gamma)$.

By letting α increase (or decrease) from zero one can find another minimizer where the rank of the γ-active subspace is increased. Repeating this argument we finally find a minimizer where the rank of the corresponding γ-active subspace is maximal (i.e., the rank is equal to n). Clearly, this proof extends to the case $b \neq \mathbf{0}$.

Let $\mathbf{s}^\gamma(x)$ be the γ-feasible sign vector at x with components

$$s_i^\gamma(x) = \begin{cases} -1 & \text{if } r_i(x) \leq -\gamma, \\ 0 & \text{if } |r_i(x)| < \gamma, \\ 1 & \text{if } r_i(x) \geq \gamma. \end{cases} \tag{21}$$

Then we have the following.

LEMMA 1. *$\mathbf{s}^\gamma(x)$ is constant for $x \in M_\gamma$. Furthermore, $r_i(x)$ is constant for $x \in M_\gamma$ if $s_i^\gamma = 0$.*

Proof. Let $z \in M_\gamma$ and let $\mathbf{s} = \mathbf{s}^\gamma(z)$; i.e., $G_\gamma(x) = Q_\mathbf{s}(x)$ for $x \in \mathcal{C}_\mathbf{s}^\gamma$. If $x \in \mathcal{C}_\mathbf{s}^\gamma \cap M_\gamma$ then $Q_\mathbf{s}''(x)(x - z) = 0$. Therefore, if $|r_i(z)| < \gamma$ then $a_i^T(x - z) = 0$ (see (16)), and hence $r_i(x) = r_i(z)$. Thus r_i is constant in $\mathcal{C}_\mathbf{s}^\gamma \cap M_\gamma$. Using the fact that M_γ is connected and r_i is continuous, it is easily seen by repeating the argument above that r_i is constant in M_γ. Next suppose $r_i(z) \geq \gamma$. Then $r_i(x) \geq \gamma$ for all $x \in M_\gamma$ because existence of $x \in M_\gamma$ with $r_i(x) < \gamma$ is excluded by the convexity of M_γ, the continuity of r_i, and the first part of the lemma. Similarly, $r_i(z) \leq -\gamma \implies r_i(x) \leq -\gamma$ for $x \in M_\gamma$. This proves Lemma 1. □

Following the lemma it is meaningful to use the notation $\mathbf{s}^\gamma(M_\gamma) = \mathbf{s}^\gamma(x), x \in M_\gamma$ as the sign vector corresponding to the solution set. Lemma 1 has the following consequences which characterize the solution set M_γ.

COROLLARY 1. *M_γ is a convex set that is contained in one Q-subset: $\mathcal{C}_\mathbf{s}^\gamma$ where $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$.*

Proof. The proof follows immediately from the linearity of the problem and Lemma 1. □

COROLLARY 2. *Let $z \in M_\gamma$, and $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$. Let $\mathcal{N}_\mathbf{s}$ be the orthogonal complement of $\mathcal{V}_\mathbf{s} = \text{span}\{a_i | s_i = 0\}$. Then*

$$M_\gamma = (z + \mathcal{N}_\mathbf{s}) \cap \mathcal{C}_\mathbf{s}^\gamma.$$

Proof. It follows from (17) that $G_\gamma'(z + u) = 0$ if $u \in \mathcal{N}_\mathbf{s}$ and $z + u \in \mathcal{C}_\mathbf{s}^\gamma$. Thus

$$M_\gamma \supseteq (z + \mathcal{N}_\mathbf{s}) \cap \mathcal{C}_\mathbf{s}^\gamma.$$

If $x \in M_\gamma$ then $r_i(x) = r_i(z)$ for $s_i = 0$, and hence $x - z \in \mathcal{N}_\mathbf{s}$. Therefore, Corollary 1 implies

$$M_\gamma \subseteq (z + \mathcal{N}_\mathbf{s}) \cap \mathcal{C}_\mathbf{s}^\gamma,$$

which proves Corollary 2. □

Notice that Corollary 2 has the following implication. Suppose the minimizer z of G_γ belongs to B_γ. Then M_γ is the intersection of $\mathcal{C}_\mathbf{s}^\gamma$ with a space of dimension at least p where $p = n - \text{rank}\{a_i^T | |r_i(z)| < \gamma\}$. Therefore, one can normally expect multiple solutions if there exists a solution in B_γ.

3. The connection between G_γ, G, and [NormLP]. In this section we let the threshold γ be variable and we analyze how the solution set M_γ changes as γ decreases and approaches zero. Furthermore, we show how to identify a solution to [NormLP].

Assume $x_\gamma \in M_\gamma$, and let $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$. Let $\mathcal{V}_\mathbf{s}$ and $\mathcal{N}_\mathbf{s}$ be defined as in Corollary 2 (i.e., if $x_\gamma \notin B_\gamma$ then $\mathcal{V}_\mathbf{s} = \mathcal{V}_\gamma(x_\gamma)$).

Since x_γ satisfies the necessary condition for a minimizer of G_γ, i.e.,

$$\mathbf{0} = AW_\mathbf{s}(A^T x_\gamma - c) + \gamma(A\mathbf{s} + b), \tag{22}$$

the following linear system is consistent:

$$(AW_\mathbf{s}A^T)d = A\mathbf{s} + b. \tag{23}$$

Inserting (23) into (22) we see that $x_0 \equiv x_\gamma + \gamma d$ is the least squares solution to the linear system

$$W_\mathbf{s}(A^T z - c) = \mathbf{0}. \tag{24}$$

LEMMA 2. *Assume that the linear system* (24) *is consistent. Then*

$$\frac{1}{\gamma} W_\mathbf{s} r(x_\gamma) = -W_\mathbf{s} A^T d, \tag{25}$$

where d is a solution to (23).

Proof. The proof follows by inserting the solution $(x_\gamma + \gamma d)$ into (24). □

Now let d solve (23) and assume $\mathbf{s}^{\gamma-\varepsilon}(x_\gamma + \varepsilon d) = \mathbf{s}$; i.e., $x_\gamma + \varepsilon d \in \mathcal{C}_\mathbf{s}^{\gamma-\varepsilon}$ for some $\varepsilon > 0$. The linearity of the problem implies $x_\gamma + \delta d \in \mathcal{C}_\mathbf{s}^{\gamma-\delta}$ for $0 \le \delta \le \varepsilon$. Therefore (22) and (23) show that $(x_\gamma + \delta d)$ is a minimizer of $G_{\gamma-\delta}$. Thus we have the following consequence of Corollary 2.

LEMMA 3. *Let $x_\gamma \in M_\gamma$ and let $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$. Let d solve* (23). *If $\mathbf{s}^{\gamma-\varepsilon}(x_\gamma + \varepsilon d) = \mathbf{s}$ for $\varepsilon > 0$ then $\mathbf{s}^{\gamma-\delta}(x_\gamma + \delta d) = \mathbf{s}$, and*

$$M_{\gamma-\delta} = (x_\gamma + \delta d + \mathcal{N}_\mathbf{s}) \cap \mathcal{C}_\mathbf{s}^{\gamma-\delta} \tag{26}$$

for $0 \le \delta \le \varepsilon$.

Lemma 3 states that the minimizers of G_γ form a family of piecewise-linear paths as a function of the parameter γ.

THEOREM 1. *There exists $\gamma_0 > 0$ such that $\mathbf{s}^\gamma(M_\gamma)$ is constant for $0 < \gamma \le \gamma_0$. Furthermore,*

$$M_{\gamma-\delta} = (x_\gamma + \delta d + \mathcal{N}_\mathbf{s}) \cap \mathcal{C}_\mathbf{s}^{\gamma-\delta} \text{ for } 0 \le \delta < \gamma \le \gamma_0,$$

where $\mathbf{s} = s^\gamma(M_\gamma)$ and d solves (23).

Proof. Since there are only a finite number of different sign vectors the theorem is a consequence of Lemma 3. □

COROLLARY 3. *Let $0 < \gamma \le \gamma_0$ and $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$. Let $x_\gamma \in M_\gamma$. If $s_i = 0$ then*

$$r_i(x_\gamma) = -(a_i^T d)\gamma, \tag{27}$$

where d is any solution of (23); *i.e., $r_i(x_\gamma)/\gamma$ is constant.*

Proof. Let $s_i = 0$. Since Theorem 1 implies $|r_i(x_\gamma + \delta d)| < \gamma - \delta$ for $0 \le \delta < \gamma$ we obtain $r_i(x_\gamma + \gamma d) = 0$. Therefore (24) is consistent and the result is a consequence of Lemma 2. □

If $x_\gamma \in M_\gamma$ then $y_\gamma = -(W_\mathbf{s} r(x_\gamma)/\gamma + \mathbf{s})$, where $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$, is feasible in [NormLP] as it is seen from (22). y_γ and x_γ are called dual solutions, and we show that y_γ solves [NormLP] when γ is small enough.

In the following two theorems we show that one can often expect the gap between the dual objective function G and the primal objective $c^T y_\gamma$ to decrease from x_γ in the direction d. In fact, we prove that the primal objective always increases and that G decreases under a certain assumption. However, all our experience indicates that G always decreases in the direction d from x_γ, but we have not been able to prove it in general.

THEOREM 2. *Let $x_\gamma \in M_\gamma$ and $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$. Let d solve (23) and let $x_0 = x_\gamma + \gamma d$. Let*

$$y_\delta = -(W_{\mathbf{s}} r(x_0 - \delta d)/\delta + \mathbf{s}), \quad \delta > 0,$$

and $H(\delta) = c^T y_\delta$. Then $H'(\gamma) \leq 0$.

Proof. Since x_0 is the least squares solution of (24), the vector

$$\rho = W_{\mathbf{s}} A^T x_0 - W_{\mathbf{s}} c$$

is orthogonal to $W_{\mathbf{s}} A^T x_0$ and

$$\|W_{\mathbf{s}} A^T x_0\|_2^2 + \|\rho\|_2^2 = \|W_{\mathbf{s}} c\|_2^2.$$

Therefore,

$$\begin{aligned} c^T W_{\mathbf{s}} r(x_0) &= c^T W_{\mathbf{s}} W_{\mathbf{s}} A^T x_0 - c^T W_{\mathbf{s}} c \\ &= (W_{\mathbf{s}} A^T x_0 - \rho)^T (W_{\mathbf{s}} A^T x_0) - \|W_{\mathbf{s}} c\|_2^2 \\ &= -\|\rho\|_2^2 \leq 0. \end{aligned} \tag{28}$$

Furthermore,

$$\begin{aligned} H(\gamma) &= -c^T (W_{\mathbf{s}} (A^T (x_0 - \gamma d) - c)/\gamma + \mathbf{s}) \\ &= -c^T W_{\mathbf{s}} r(x_0)/\gamma + c^T W_{\mathbf{s}} A^T d - c^T s. \end{aligned}$$

Therefore (28) implies

$$H'(\gamma) = c^T W_{\mathbf{s}} r(x_0)/\gamma^2 \leq 0. \qquad \square$$

It follows from the proof that if (24) is consistent then $H'(\gamma) = 0$.

We now extend the definition of $\mathbf{s}_\gamma$ to the case $\gamma = 0$:

$$s_{0i}(x) = \begin{cases} -1 & \text{if } r_i(x) < 0, \\ 0 & \text{if } r_i(x) = 0, \\ 1 & \text{if } r_i(x) > 0. \end{cases} \tag{29}$$

Furthermore, let

$$\mathcal{C}_{\mathbf{s}}^0 = \mathrm{cl}\{z \in \Re^n | \mathbf{s}_0(z) = \mathbf{s}\}. \tag{30}$$

THEOREM 3. *Let $x_\gamma \in M_\gamma$. If (24) is consistent then the directional derivative*

$$\lim_{\delta \downarrow 0} \{(G(x_\gamma + \delta d) - G(x_\gamma))/\delta\}$$

is nonpositive.

Proof. Let $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$, $\mathbf{s}_0 = \mathbf{s}_0(x_\gamma)$, and $\mathcal{A} = \{i | s_i = 0\}$. First we notice that for $i \in \mathcal{A}$ the consistency of (24) implies that if $r_i(x_\gamma) = 0$ then $r_i(x_\gamma + \delta d) = 0$ for any δ. Therefore the following holds for δ being a sufficiently small positive number:

$$\begin{aligned} G(x_\gamma + \delta d) &= \mathbf{s}^T r(x_\gamma + \delta d) + \mathbf{s}_0^T W_\mathbf{s} r(x_\gamma + \delta d) + b^T(x_\gamma + \delta d) \\ &= G(x_\gamma) + \delta[\mathbf{s}^T A^T d + b^T d + \mathbf{s}_0^T W_\mathbf{s} A^T d]. \end{aligned}$$

Using (23) and (25) we obtain

$$\begin{aligned} [G(x_\gamma + \delta d) - G(x_\gamma)]/\delta &= d^T A W_\mathbf{s} A^T d + \mathbf{s}_0^T W_\mathbf{s} A^T d \\ &= d^T A W_\mathbf{s}^T W_\mathbf{s} A^T d + \mathbf{s}_0^T W_\mathbf{s} A^T d \\ &= (W_\mathbf{s} r(x_\gamma)/\gamma)^T (W_\mathbf{s} r(x_\gamma)/\gamma) - \mathbf{s}_0^T W_\mathbf{s} r(x_\gamma)/\gamma \\ &= \sum_{i \in \mathcal{A}} \left[\left(\frac{r_i(x_\gamma)}{\gamma} \right)^2 - s_{0i} \frac{r_i(x_\gamma)}{\gamma} \right]. \end{aligned}$$

The last sum is nonpositive because $|r_i(x_\gamma)| < \gamma$ for $i \in \mathcal{A}$. This proves the theorem. □

THEOREM 4. *Let $0 < \gamma \leq \gamma_0$, where γ_0 is given in Theorem 1, and let $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$. Let $x_\gamma \in M_\gamma$ and d solve (23). Then any point in the set*

$$M_0 = (x_\gamma + \gamma d + \mathcal{N}_\mathbf{s}) \cap \mathcal{C}_\mathbf{s}^0 \tag{31}$$

solves [AL1], *and*

$$y_\gamma = -\left(\frac{1}{\gamma} W_\mathbf{s} r(x_\gamma) + \mathbf{s} \right) \tag{32}$$

solves [NormLP].

Proof. Assume $x_0 \in M_0$. Then there exists a solution d_0 to (23) such that $x_0 = x_\gamma + \gamma d_0$. Now the linearity and Theorem 1 imply that $x_{\gamma-\delta} = x_\gamma + \delta d_0 \in M_{\gamma-\delta}$ for $0 \leq \delta \leq \gamma$.

Since $\mathbf{s}^\gamma(x_\gamma) = \mathbf{s}^{\gamma-\delta}(x_{\gamma-\delta})$ for $0 \leq \delta < \gamma$, the continuity of r gives

$$r_i(x_0) \neq 0 \implies \operatorname{sign}(r_i(x_0)) = \operatorname{sign}(r_i(x_{\gamma-\delta})) = s_i = -y_{\gamma i} \tag{33}$$

for δ close enough to γ. Therefore,

$$\begin{aligned} G(x_0) &= -r(x_0)^T y_\gamma + b^T x_0 \\ &= -x_0^T A y_\gamma + c^T y_\gamma + x_0^T b \\ &= c^T y_\gamma. \end{aligned}$$

Furthermore, y_γ is feasible for [NormLP]. Hence, x_0 and y_γ are dual solutions to [AL1] and [NormLP]. Since this holds for any $x_0 \in M_0$ the theorem is proved. □

It is shown in [15] that M_0 coincides with the solution set of [AL1]. Notice that every $x_0 \in M_0$ corresponds to the same dual solution y_γ. This means that if M_0 contains more than one point then the Lagrange multipliers corresponding to y_γ are nonunique; i.e., y_γ is a degenerate solution to [NormLP]. On the other hand, if [AL1] has a degenerate solution (i.e., nonunique Lagrange multiplier) then [NormLP] has multiple solutions. In that case, Theorem 4 picks one specific solution y_γ to [NormLP].

4. The algorithm. The new algorithm is based on minimizing the function G_γ for a set of decreasing values of γ. It can be described as follows. Starting from a point x, we find a minimizer of G_γ for some $\gamma > 0$. For example, we locate a path among the possibly infinitely many paths defined by the minimizers of G_γ. Utilizing the piecewise-linear dependence of the paths on γ (Lemma 3) and the duality gap, we compute a guess at the minimizer of G_γ for a smaller value of γ. Starting from this guess, we compute the exact minimizer of G_γ using a Newton-type algorithm. Note that this gives the algorithm a predictor–corrector flavor in that we allow ourselves to deviate from the path only to return to it later from the predicted point using the Newton algorithm. Hence, we follow the solution paths closely without having to stay on any of them. Based on Theorem 4, this process terminates when the duality gap is closed.

To initiate the algorithm, a starting point x^0 and threshold value γ^0 are found as follows. Compute a least squares solution, i.e., a minimum x_{ls} of the quadratic function $f(x) \equiv r^T(x)r(x) + b^T x$, by solving

$$(AA^T)x_{ls} = Ac - \frac{1}{2}b. \tag{34}$$

Then set $x^0 = x_{ls}$ and γ^0 such that $|\mathcal{A}_{\gamma^0}(x^0)| \geq n$.

The algorithm has two main components: (1) the solution of the smooth problem, i.e., minimization of G_γ for a given value of γ, and (2) the reduction of γ and the computation of an initial point for the solution of the subsequent Huber problem. We now consider these two components in detail.

4.1. Computing a minimizer of G_γ. The algorithm for computing a minimizer x_γ of G_γ is based on a modified Newton algorithm given in [12]. A search direction h is computed by minimizing the quadratic $Q_{\mathbf{s}}$ where $\mathbf{s} = \mathbf{s}_\gamma(x)$ and x is the current iterate. More precisely, we consider the equation

$$Q''_{\mathbf{s}} h = -Q'_{\mathbf{s}}(x), \tag{35}$$

where $Q''_{\mathbf{s}}$ and $Q'_{\mathbf{s}}$ denote the gradient and Hessian of $Q_{\mathbf{s}}$, respectively. From (16), (17), and (18) we obtain

$$(AW_{\mathbf{s}}A^T)h = -AW_{\mathbf{s}}r - \gamma(A\mathbf{s} + b). \tag{36}$$

For ease of notation let $C \equiv AW_{\mathbf{s}}A^T$ and $g \equiv -AW_{\mathbf{s}}r - \gamma(A\mathbf{s} + b)$. Furthermore, let $\mathcal{N}(C)$ denote the null space of C. If C has full rank, then h is the solution of (36). Otherwise, if the system of equations (36) is consistent, h is taken to be the minimum norm solution. If the system is inconsistent, h is taken to be the projection of g on $\mathcal{N}(C)$. The motivation for these choices is discussed in [12]. The next iterate is found by a line search aiming for a zero of the directional derivative. This procedure is computationally cheap as a result of the piecewise linear nature of G'_γ. It is shown in [12] that the iteration is finite; i.e., after a finite number of iterations we have $x + h \in \mathcal{C}^\gamma_{\mathbf{s}(x)}$. Therefore, $x + h$ is a minimizer of G_γ as a result of (15), (16), and the convexity of G_γ. We summarize below the modified Newton algorithm.

```
repeat
    s = s_γ(x)
    if (36) consistent then
        find h from (36)
```

```
        if x + h ∈ C_s^γ then
            x ← x + h
            stop = true
        else
            x ← x + αh (line search)
        endif
    else
        compute h = null space projection of g
        x ← x + αh (line search)
    endif
until stop.
```

4.2. Checking optimality and reducing γ. Let x_γ be a minimizer of G_γ computed using the modified Newton algorithm of the previous section. Then either the continuation algorithm terminates or the modified Newton algorithm is restarted using a reduced value of γ. Let $\mathbf{s} = \mathbf{s}^\gamma(M_\gamma)$ and d be a solution to (23). From the necessary condition (22) it follows that (23) is equivalent to

$$(AW_{\mathbf{s}}A^T)d = -\frac{1}{\gamma}AW_{\mathbf{s}}r(x_\gamma). \tag{37}$$

Let d solve (37), and let $r_\gamma \equiv r(x_\gamma)$. For $0 \le \delta \le \gamma$ we define

$$x(\gamma - \delta) = x_\gamma + \delta d. \tag{38}$$

Then

$$r(x(\gamma - \delta)) = r_\gamma + \delta A^T d. \tag{39}$$

Notice that, following Lemma 3, $x(\gamma-\delta)$ is a minimizer of $G_{\gamma-\delta}$ as long as $\mathbf{s}_{\gamma-\delta}(x(\gamma-\delta)) = \mathbf{s}$.

The stopping test is based on Theorem 4. It consists of checking the duality gap $(G(x(0)) - c^T y_\gamma)$, where y_γ is given in (32). If this is zero (within the roundoff tolerance) then the algorithm is stopped provided the residuals satisfy the following sign accordance:

$$\begin{aligned} s_{\gamma i} = 0 &\implies r_i(x(0)) = 0, \\ s_{\gamma i} \neq 0 &\implies s_{\gamma i} r_i(x(0)) \ge 0. \end{aligned}$$

Otherwise, we let $y(\gamma - \delta) = -(\frac{1}{\gamma-\delta}W_{\mathbf{s}}r(x(\gamma - \delta)) + \mathbf{s})$, and then we attempt to find the value δ^* of δ for which $|G(x(\gamma - \delta)) - c^T y(\gamma - \delta)|$ attains its minimum. This is basically done by examining the points $\delta = \frac{k}{10}\gamma$, $k = 1, \ldots, 10$, and choosing the best of these in the above sense provided that $\mathbf{s}_{\gamma-\delta^*}(x(\gamma - \delta^*)) \neq \mathbf{s}$. Then $x(\gamma - \delta^*)$ given by (38), with $r(x(\gamma - \delta^*))$ given by (39), starts the Newton iteration using the new threshold value $\gamma - \delta^*$. The precise description of this procedure is given below.

```
Δ_0 = G(x_γ) − c^T y(γ)
k ← 0
while not STOP do
    k ← k + 1
    δ ← k * γ/10
```

$x \leftarrow x_\gamma + \delta d$
$r \leftarrow r_\gamma + \delta A^T d$
$y \leftarrow y(\gamma - \delta)$
$\Delta_k = G(x) - c^T y$
end while
if $k \leq 2$ then $\delta \leftarrow 0.1 * \delta$
else $\delta \leftarrow (k-1) * \gamma/10$
endif
$x \leftarrow x_\gamma + \delta d$
$r \leftarrow r_\gamma + \delta A^T d$
$\gamma \leftarrow \gamma - \delta$

STOP is a function that returns true when one of the following conditions hold:

$$\Delta_k > \Delta_{k-1}, \quad \Delta_k < -0.1\Delta_0,$$

provided that $\mathbf{s}_{\gamma-\delta}(x(\gamma - \delta)) = \mathbf{s}$ or when

$$k = 9.$$

The first condition ensures that the procedure stops close to a minimum. The second condition guards against the case where the sign of the difference in the function values is reversed.

We have also experimented with alternative procedures for reducing γ. We used the total iteration count as the performance measure. The experiments showed that there is no significant difference among the alternative procedures in terms of performance. As a result, we adopted the above procedure since it has a sound justification as a result of Theorem 4. We note that this procedure guarantees that unless the duality gap is closed, γ is decreased by at least a factor of 0.9 after each unconstrained minimization. Hence, we have the following theorem.

THEOREM 5. *The continuation algorithm described in* §§4.1 *and* 4.2 *stops at an optimal solution* y^* *of* [NormLP] *after a finite number of iterations.*

Proof. As a result of the observation, γ is reduced by a certain factor after each unconstrained minimization phase unless optimality is reached. Hence, using Theorem 1, γ can only be decreased a finite number of times. Since the modified Newton algorithm is finite [12], the result follows. □

5. Numerical results. In this section we report our numerical experience with a Fortran 77 implementation of the new algorithm, which does not exploit sparsity. We use two sets of test problems:

1. randomly generated full problems,
2. sparse problems from the Netlib collection [7].

We have three goals when we perform numerical experiments. The first goal is to examine the growth in solution time and iteration count of the new algorithm as the problem size is increased. The second goal is to test the numerical accuracy of the algorithm. We use randomly generated test problems to achieve these ends. The third goal is to test the viability of the algorithm in solving problems of practical interest. To accomplish this we choose a set of small, sparse test problems from the Netlib collection. Since these problems are very different in nature from the randomly generated problems this should give us insight into the potential of the new algorithm as a competitor to the well-known algorithms for LP. To this end, we compare our

results with an LP subroutine E04MBF from the NAG subroutine library. E04MBF is based on LSSOL from Stanford Systems Optimization Library. It is a Fortran 77 package for constrained linear least squares problems, LP, and convex quadratic programming [9]. It does not exploit sparsity. Hence, it provides a fair comparison with our numerical results.

The major effort in the Newton algorithm of §4.1 is spent in solving the systems (36). We use the AAFAC package of [17] to perform this. The solution is obtained via an LDL^T factorization of the matrix $C_k = AW_\gamma(x_k)A^T$, so D and L are computed directly from the active columns of A, i.e., without squaring the condition number as would be the case if C_k was first computed. It is essential for the efficiency of the continuation algorithm that normally the γ-active set $\mathcal{A}_\gamma(x_k)$ differs relatively little from $\mathcal{A}_\gamma(x_{k-1})$. This implies that the factorization of C_k can be obtained by relatively few up- and downdates of the factorization of C_{k-1}. Therefore, the computational cost of a typical iteration step is $O(n^2)$. Occasionally, a refactorization is performed. This consists of the successive updating $LDL^T \leftarrow LDL^T + a_j a_j^T$ for all j in the active set (starting with $L = I, D = 0$). It is considered only when some columns of A leave the active set, i.e., when downdating is involved. If many columns leave we may refactorize because it is cheaper. Otherwise, a refactorization is used when a downdating results in a rank decrease and there is an indication that rounding errors have marked influence. This part of the algorithm combines ideas from [6, 8]. For details see §2 in [17]. The refactorization is an $O(n^3)$ process.

The solution of (23) is done via (37), and the matrix $AW_{\mathbf{s}}A^T$ is identical to the last C_k used in the Newton iteration. Thus, the LDL^T factorization is available. In order to enhance accuracy we use one step of iterative refinement when solving (37). For further details see [18]. We refer to the subroutine that implements the continuation algorithm as LPASL1. With the exception of some internal tolerance parameters (e.g., tolerances used for numerical checks for zero), LPASL1 does not allow any control over the execution of the algorithm. Hence, all the results reported in this study were obtained under identical algorithmic choices. We note that LPASL1 is available for distribution as a Fortran 77 subroutine. Details can be found in [16].

First we report computational results using randomly generated problems. Our random problems are generated using ideas similar to the generator detailed in §4.1 of [18]. All the problems are of the form [NormLP]. Half of the matrix A in each problem was taken as the identity matrix to simulate the presence of slack columns. We solve ten problems of each size. The tests were performed on a DEC 3000 Alpha APX Model 500 running DEC OSF1 V1.2. The Fortran code was compiled using the `-O` option of the DEC Fortran compiler. The ratio $\mu = m/n$ is kept constant at 2. In all cases, the duality gap in LPASL1 was below 10^{-8}. In Table 1, we report the average over ten problems of the following LPASL1 statistics: number of iterations, run time in CPU seconds, number of refactorizations, and number of γ-reductions. The column "iter" refers to the total number of iterations, i.e., the total number of solutions to (36) and (37), and "reduc" to the number of γ-reductions. The column "refac" refers to the total number of refactorizations in connection with the computations of the factors L and D. In all cases, the cardinality of the set $\mathcal{A}_0(x_0)$ and the rank of the associated matrix $AW_{\mathbf{s}}A^T$ on termination are equal to n. The two columns under the heading LSSOL contain the run time statistics of LSSOL. The rightmost column displays the ratio run time of LSSOL/run time of LPASL1. In all cases, the optimal objective function values reported by LPASL1 and LSSOL agree to at least eight digits. All runs with LPASL1 and LSSOL were performed using default parameters; i.e., no fine

TABLE 1
Solution statistics of LPASL1 and LSSOL using randomly generated problems (NA: not solved due to memory shortage).

m, n	LPASL1				LSSOL		ratio
	iter	refac	reduc	CPU	iter	CPU	
(200,100)	56.2	4	16.4	0.6	352.2	4.7	7.83
(320,160)	74.1	4.4	22.2	2.6	713.5	21.8	8.38
(480,240)	104.7	6	30.2	9.9	1300.7	91.3	9.22
(720,360)	117.2	4.8	32.4	30.5	2312	360.4	11.81
(1080,540)	165.2	5.4	44.3	116.5	4305.6	1470.9	12.62
(1620,810)	217.4	8.5	51.7	469.4	8628.6	6501.7	13.85
(2430,1215)	290.7	13.1	62.9	2648.7	16205.1	27777	10.48
(3644,1822)	335	13.2	62.6	8327.3	NA	NA	NA

tuning of the codes was done for any test problem.

It was observed that over a set of ten problems of a given size, the performance statistics of LPASL1 display a larger variation around the mean than those of LSSOL. For instance, the minimum run time for problems of size (2430,1215) was 1400 seconds while the maximum run time was 3569 seconds. For LSSOL, the two extreme values were 26441 seconds and 29113 seconds. However, we observe that LPASL1 performs increasingly better than LSSOL as the problem size is increased.

A least squares fit with the model $\beta \cdot n^{\alpha}$ shows that

$$\text{iter}_{LPASL1} \approx 3.0 \cdot n^{0.63}, \quad \text{iter}_{LSSOL} \approx 0.3 \cdot n^{1.53},$$

$$\text{CPU}_{LPASL1} \approx 1.2 \cdot 10^{-7} \cdot n^{3.31}, \quad \text{CPU}_{LSSOL} \approx 4.7 \cdot 10^{-7} \cdot n^{3.48}.$$

We observe that on both measures LSSOL exhibits a higher growth rate than LPASL1.

Finally we consider ten problems from the Netlib collection. We also used a test problem from a civil engineering application, referred to as PLATE. The characteristics of the problems are given in Table 2.

TABLE 2
Characteristics of the sparse test problems.

Problem name	Constraints	Variables	Slack variables
afiro	27	32	19
sc50b	50	48	30
sc50a	50	48	30
sc105	105	103	60
adlittle	56	97	41
scagr7	129	140	45
stocfor1	117	111	54
blend	74	83	31
sc205	205	203	114
share2b	96	79	83
plate	61	73	60

All the problems we consider are of the form

$$\begin{aligned} \underset{y}{\text{maximize}} \quad & c^T y \\ \text{subject to} \quad & Ay \leq b, \\ & y \geq 0. \end{aligned}$$

In order to apply the continuation algorithm to these problems, we prescribe a uniform bound u to all the variables. Hence, we artificially transform the problem to the following equivalent problem:

$$\begin{array}{lll} \underset{y}{\text{maximize}} & c^T y & \\ \text{subject to} & Ay & = b, \\ & 0 \leq y & \leq U, \end{array}$$

where $U = (u, \ldots, u)$. Then the problem can be transformed to the form [NormLP] by a simple linear transformation. No preprocessing was used prior to solving the problems. In Table 3, we show the solution statistics of LPASL1 and LSSOL. The tests were performed on an IBM PS/2 486. The Salford F77 compiler was used. The columns "iter," "reduc," and "refac" are as defined in Table 1. The columns "ν_0"and "rank" refer to the cardinality of the set $\mathcal{A}_0(x_0)$ and the rank of the associated matrix $AW_{\mathbf{s}}A^T$ on termination, respectively. Finally, we report the CPU time in seconds exclusive of input/output under "CPU." In all cases, the optimal objective function values reported by LPASL1 and LSSOL agree to at least eight digits. Both LPASL1 and LSSOL were run using default parameters.

TABLE 3
Solution statistics of LPASL1 and LSSOL on sparse problems.

Problem name	LPASL1						LSSOL	
	iter	refac	reduc	ν_0	rank	CPU	iter	CPU
afiro	21	4	4	28	27	0.49	19	0.60
sc50b	22	2	1	48	48	0.93	47	3.07
sc50a	32	3	3	47	47	1.64	40	2.47
sc105	56	8	1	101	101	9.78	78	14.28
adlittle	89	18	15	62	56	8.13	149	14.56
scagr7	76	7	10	129	129	19.06	126	58.07
stocfor1	101	13	3	111	111	19.01	73	15.93
blend	60	8	3	73	72	7.25	85	12.96
sc205	152	16	3	196	196	107.63	192	122.74
share2b	87	8	5	91	86	11.86	122	13.07
plate	20	3	4	55	43	2.47	150	9.23

We observe that LPASL1 is faster than LSSOL on ten of the eleven problems. Furthermore, in the remaining problem, stocfor1, the performance figures of the two codes are very close. We expect to report results with all the Netlib problems in the future when we have an implementation that exploits sparsity.

Comparison with interior point and related methods. Our algorithm displays some similarities to interior point methods for LP and to the affine scaling methods of Coleman and Li in the main computational step. A common feature of all these algorithms is that their main computational step is the solution of a weighted least squares system of the form

$$(ADA^T)p = f, \tag{40}$$

where D is a diagonal positive definite matrix, and p and f are vectors of appropriate dimensions. Since the numerical values of the entries of D change from one iteration to the next, a refactorization of the matrix ADA^T is performed at each iteration of these methods. Since we do only occasional refactorizations of the matrix AWA^T

in our method, the cost of our average iteration, which consists of a series of up- and downdates of the factors, is lower compared with that of the above-mentioned methods. Hence, the relative standing of our method with respect to these methods can only be established by a thorough computational comparison. To give the reader a feel for how the algorithms compare, we ran some of our randomly generated test problems using the CPLEX Barrier optimizer, which is a state-of-the-art implementation of a primal-dual interior point algorithm. The results are summarized in Table 4 below. These runs were made on a SUN 490 workstation.

TABLE 4
Solution statistics of LPASL1 and CPLEX on randomly generated problems.

(m,n)	LPASL1				CPLEX barrier	
	iter	refac	reduc	CPU	iter	CPU
200,100	43	8	5	19.95	13	16.95
320,160	51	6	8	65.83	14	67.84
400,200	58	4	12	104.90	15	140.39
600,300	68	5	13	356.73	15	451.79

We notice that LPASL1 statistics in Table 4 differ from the results displayed in Table 1 for similar size problems. The explanation of this is the following. To carry out the tests reported in Table 4 we generated random problems where all columns were fully dense in contrast to our problems in Table 1, where half of the matrix A was taken as the identity matrix to simulate slack columns. The reason for this change is to force CPLEX Barrier to use dense linear algebra. Interestingly, we observed that LPASL1 consistently took fewer iterations and reductions on problems of the type used in Table 4 compared with its performance reported in Table 1. Computing times in Tables 1 and 4 also differ due to the difference in the computing platforms used for the two experiments.

We also observe in Table 4 that the iteration number in the interior point method grows very slowly. The time performances of LPASL1 and CPLEX Barrier are very close on these test problems with a slight advantage of LPASL1 as the problems get larger. Given the preliminary nature of our numerical work on the new algorithm, these results are encouraging.

We expect similar behavior from Coleman and Li's algorithm since it is reported in their paper that the iteration count of their algorithm is almost constant as a function of the problem size. We are not able to do any comparisons since they only have a MATLAB^{TM} code.

6. Summary. In this paper, we described a new finite continuation algorithm for LP based on linear ℓ_1 optimization. The algorithm solves a sequence of smooth nonlinear problems that yield a solution to the LP in a finite number of iterations. We studied the structure of the solution set of the approximating problems and showed how primal and dual solutions can be obtained. We also developed an efficient and stable implementation of the algorithm for full matrices and tested it on both randomly generated problems and some problems from the Netlib collection. Although these results are of a preliminary nature, they indicate that the algorithm exhibits a promising performance on both kinds of problems. We compared our results with a simplex LP routine from the Stanford Systems Optimization Library that is available as a subroutine in the NAG library, and with the CPLEX Barrier Optimizer. The comparison showed that the new algorithm is substantially faster than the NAG routine on randomly generated problems with full matrices. It is also competitive on the

Netlib problems tried. The comparison with CPLEX also gave encouraging results. The following issues remain for further studies:

- "hot" starts,
- preprocessing and automatic choice of an upper bound for problems with only nonnegativity restrictions on the variables,
- exploiting sparsity.

REFERENCES

[1] R. H. BARTELS (1980), *A penalty linear programming method using reduced-gradient basis-exchange techniques*, Linear Algebra Appl., 29, pp. 17–32.

[2] B. CHEN AND P.T. HARKER (1993), *A non-interior continuation algorithm for linear and quadratic programming*, SIAM J. Optim., 3, pp. 503–515.

[3] T. COLEMAN AND Y. LI (1990), *A globally and quadratically convergent affine scaling algorithm for the linear programming problem with bounded variables*, in Large Scale Numerical Optimization, T. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 49–57.

[4] ——— (1992), *A globally and quadratically convergent affine scaling algorithm for ℓ_1 problems*, Math. Programming, 56, pp. 189–222.

[5] A. R. CONN (1976), *Linear programming via a nondifferentiable penalty function*, SIAM J. Numer. Anal., 13, pp. 145–154.

[6] R. FLETCHER AND M. J. D. POWELL (1974), *On the modification of LDL^T factorizations*, Math. Comp., 28, pp. 1067–1087.

[7] D. M. GAY (1985), *Electronic mail distribution of linear programming test problems*, Math. Prog. Soc. COAL Newsletter, 13, pp. 10–12.

[8] W. M. GENTLEMAN (1973), *Least squares computation by Givens transformations without square roots*, J. Inst. Math. Appl., 12, pp. 329–336.

[9] P. E. GILL, S. HAMMARLING, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT (1986), *User's Guide for LSSOL (Version 1.0): A Fortran Package for Constrained Linear Least Squares and Convex Quadratic Programming*, Report SOL 86-1, Department of Operations Research, Stanford University, Stanford, CA.

[10] P. HUBER (1981), *Robust Statistics*, John Wiley, New York.

[11] I. LUSTIG, R. MARSTEN, AND D. SHANNO (1991), *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152, pp. 191–222.

[12] K. MADSEN AND H.B. NIELSEN (1990), *Finite algorithms for robust linear regression*, BIT, 30, pp. 682–699.

[13] ——— (1993), *A finite smoothing algorithm for linear ℓ_1 estimation*, SIAM J. Optim., 3, pp. 223–235.

[14] K. MADSEN, H. B. NIELSEN, AND M. C. PINAR (1992), *Linear, Quadratic and Minimax Programming Using ℓ_1 Optimization*, Report NI-92-11, Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, Lyngby, Denmark.

[15] ——— (1994), *New characterizations of ℓ_1 solutions to overdetermined linear systems*, Oper. Res. Lett., 16, 3, pp. 159–166.

[16] ——— (1994), *User's Guide to* LPASL1: *A Fortran 77 Subroutine, Based on a Continuation Algorithm, for Dense Linear Programming*, Report IMM-94-01, Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, Lyngby, Denmark.

[17] H. B. NIELSEN (1990), *AAFAC: A Package of Fortran 77 Subprograms for Solving $A^T Ax = c$*, Report NI-90-11, Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, Lyngby, Denmark.

[18] ——— (1991), *Implementation of a Finite Algorithm for Linear ℓ_1 Estimation*, Report NI-91-01, Institute of Mathematical Modelling, Numerical Analysis Group, Technical University of Denmark, Lyngby, Denmark.

[19] M. Ç. PINAR AND S. A. ZENIOS (1994), *On smoothing exact penalty functions for convex constrained optimization*, SIAM J. Optim., 4, pp. 486–511.

[20] T. ROCKAFELLAR (1990), *Computational schemes for large-scale problems in extended linear-quadratic programming*, Math. Programming, 48, pp. 447–474.

[21] J. SUN (1992), *On the structure of convex piecewise quadratic functions*, J. Optim. Theory Appl., 72, pp. 499–510.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 617–625, August 1996
© 1996 Society for Industrial and Applied Mathematics
005

A CONJUGATE DUALITY SCHEME GENERATING A NEW CLASS OF DIFFERENTIABLE DUALS*

AHARON BEN-TAL† AND MARC TEBOULLE‡

Abstract. We construct a mechanism to generate a large class of duality schemes for (not necessarily differentiable) convex optimization problems, for which the dual problem is continuously differentiable. We use the conjugate duality framework of Rockafellar; the original primal problem is embedded in a family of perturbed problems. The perturbation function is constructed in terms of two perturbation vectors and a single-variable function q. The differentiability is a consequence of the dual objective function's being a kind of "proximal regularization," but one which is expressed in terms of a *nonquadratic* regularizing term associated with the function q.

Key words. conjugate duality, convex optimization, augmented Lagrangians, entropic proximal mappings

AMS subject classifications. 90C25, 90C30

1. Introduction.

For a convex programming problem

$$\text{(P)} \qquad \min_{x\in\mathbb{R}^n}\{f(x):\ g_i(x)\le 0,\quad i=1,\dots,m\},$$

the *classical Lagrangian dual* problem consists of maximizing the dual objective function

$$h_0(y) := \inf_x l_0(x,y)$$

given in terms of the *classical Lagrangian*

$$\ell_0(x,y) := f(x) + \sum_{i=1}^m y_i g_i(x).$$

A disadvantage associated with this duality scheme is the fact that $h_0(y)$ may well be nondifferentiable, and thus not amenable to solution by algorithms requiring continuous differentiability of the objective function.

If, instead of ℓ_0, one employs the *quadratic augmented Lagrangian* (see Rockafellar [7])

$$\mathcal{L}_r(x,y) = f(x) + \frac{1}{2r}\sum_{i=1}^m \left[\max^2(0, y_i + rg_i(x)) - y_i^2\right],\quad r>0$$

then the corresponding dual objective function

$$G_r(y) = \inf_x\ \mathcal{L}_r(x,y)$$

* Received by the editors August 23, 1993; accepted for publication (in revised form) November 14, 1994.

† Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa 32000, Israel (morbt@techunix.technion.ac.il). The research of this author was partially supported by German-Israeli Foundation for Scientific Research and Development grant I-143.138.06/89.

‡ School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel (teboulle@math.tau.ac.il). The research of this author was partially supported by National Science Foundation grants DMS-9201297 and DMS-9401871. Currently on leave from the Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD.

is continuously differentiable [7, Thm. 3.2]. This fact follows from a dual expression for G_r:

$$G_r(y) = \max_z \left\{ h_0(z) - \left(\frac{1}{2r}\right) \|z - y\|^2 \right\},$$

which shows that G_r is a *Moreau–Yosida proximal regularization* of the function h_0, and thus its differentiability follows from the celebrated result of Moreau [4].

Many other nonquadratic augmented Lagrangians have been suggested in the past (see, e.g., the monographs [1], [3] and the more recent paper [5]), each giving rise to a particular dual problem. However, a result similar to the differentiability of the dual objective function $G_r(y)$ (originating from the *quadratic* augmented Lagrangian $\mathcal{L}_r(x, y)$) is not generally available for *nonquadratic* augmented Lagrangians.

In this paper, we construct a mechanism to produce a large class of duality schemes, for which the *dual objective function is continuously differentiable, even if the primal problem* (P) *is nondifferentiable.* We use the conjugate duality framework of Rockafellar [6], [8]. The essential ingredient in this scheme is the embedding of the original primal problem (P) in a family of perturbed problems, for which (P) corresponds to the "zero-perturbation" case. In our scheme, the perturbation function is constructed in terms of two perturbation vectors u and v and a function $q : \mathbb{R} \to \mathbb{R}$. It is the choice of this function q which determines the particular dual problem that emerges.

The above perturbation function and the induced augmented Lagrangian function are discussed in §2. In §3, the continuously differentiable dual problem is obtained. The differentiability is established by showing that the dual objective function is a kind of "proximal regularization," but one which uses "entropic distance" instead of the usual quadratic distance. This type of entropic proximal regularization was recently introduced in [9].

The possibility of having a *differentiable* dual problem may have important consequences in devising new algorithms that solve a convex optimization problem by way of its dual. This possibility will be explored in future work.

2. A family of perturbed problems and their conjugate duals. In the theory of conjugate duality (see, e.g., Rockafellar [8]), the problem (P) is embedded in a family of perturbed optimization problems parametrized by a "perturbation vector." Here, we embed (P) in the following family:

$$(P_\gamma) \quad \min_x F_\gamma(x; u, v), \tag{2.1}$$

where the perturbation function $F_\gamma : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is defined, for a given parameter $\gamma > 0$, in terms of *two* perturbation vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^m$ and in terms of a function $q : \mathbb{R} \to \mathbb{R}$ as follows:

$$F_\gamma(x; u, v) = \begin{cases} f(x) & \text{if } g_i(x) \le v_i + \gamma q(u_i), \quad i = 1, \dots, m, \\ \infty & \text{otherwise.} \end{cases} \tag{2.2}$$

The function q is chosen so that

(Q1) q is a differentiable strictly increasing and strictly concave on its domain $(a, +\infty)$ with $q(a) = -\infty$,

(Q2) $q(0) = 0$,

(Q3) $q'(0) = 1$,

(Q4) $q'(a) = +\infty$.

Two examples of a function q satisfying (Q1)–(Q4) are

$$\text{(i)}\quad q(t) = 1 - e^{-t}\ (a = -\infty),$$
$$\text{(ii)}\quad q(t) = \log(1+t)\ (a = -1).$$

Note that the function F_γ is convex in (x, u, v) since $f, \{g_i\}_{i=1}^m$ and $-q$ are given convex functions.

To derive a dual problem via conjugate duality, we construct the Lagrangian function

$$L_\gamma(x, y, z) = \inf_{u,v}\{F_\gamma(x; u, v) + \langle u, y\rangle + \langle v, z\rangle\} \tag{2.3}$$

and define the dual problem as

$$(D_\gamma) \qquad \sup_{y,z} H_\gamma(y, z), \tag{2.4}$$

where the dual objective function H_γ is computed by

$$H_\gamma(y, z) = \inf_x L_\gamma(x, y, z). \tag{2.5}$$

For $\gamma = 0$ or when $u = 0$ (since $q(0) = 0$ by (Q2)), we have $L_0(x, y, z) = f(x) + \langle z, g(x)\rangle = l_0(x, z)$ if $z \geq 0$ and $-\infty$ otherwise, recovering the usual Lagrangian associated with problem (P) and from (2.4) the corresponding dual problem (D_0).

Our next result gives an explicit representation of the Lagrangian L_γ defined in (2.3). Let $C := (\mathbb{R}^2_{++} \cup \{(0,0)\})^m$.

PROPOSITION 2.1. *For any $\gamma \geq 0$ one has*

$$L_\gamma(x, y, z) = \begin{cases} \ell_0(x, z) + \gamma \sum_{i=1}^m z_i q_*\left(\dfrac{y_i}{\gamma z_i}\right) & \text{if } (y, z) \in C, \\ -\infty & \text{otherwise,} \end{cases} \tag{2.6}$$

where $q_*(s) := \inf_t\{st - q(t)\}$ is the concave conjugate of q.

Proof. By the definition of the Lagrangian function (2.3), and using (2.2), we have

$$L_\gamma(x, y, z) = \inf_u\{f(x) + \langle u, y\rangle + \inf_v\{\langle v, z\rangle :\ v_i \geq g_i(x) - \gamma q(u_i),\ \ i = 1, \ldots, m\}\}$$

$$= \begin{cases} f(x) + \langle z, g(x)\rangle + \inf_u \left\{\langle u, y\rangle - \gamma \sum_{i=1}^m z_i q(u_i)\right\} & \text{if } z \geq 0, \\ -\infty & \text{if } z \ngeq 0. \end{cases} \tag{2.7}$$

To compute the infimum in (2.7), we first note that

$$\operatorname{dom} q_* := \{s : q_*(s) > -\infty\} \subset \mathbb{R}_+.$$

Indeed, since q is monotone, using the fact that ri dom $q_* \subset$ range $q' \subset$ dom q_* (see, e.g., [5]), where ri denotes relative interior, we conclude that dom $q_* \subset \mathbb{R}_+$. Therefore

with the convention $0q_*(0/0) = 0$ we have

$$\inf_u \left\{ \langle u, y \rangle - \gamma \sum_{i=1}^m z_i q(u_i) \right\} = \gamma \sum_{i=1}^m z_i \inf_{u_i} \left\{ \frac{u_i y_i}{\gamma z_i} - q(u_i) \right\}$$

$$= \gamma \sum_{i=1}^m z_i q_* \left(\frac{y_i}{\gamma z_i} \right) \tag{2.8}$$

if $(y, z) \in C$ and $-\infty$ otherwise. Substituting (2.8) in (2.7) we have proved the desired result (2.6). □

The dual of (P) corresponding to L_γ is thus (cf. eq. (2.4))

$$(D_\gamma) \qquad \sup\{H_\gamma(y, z) : \ (y, z) \in C\}. \tag{2.9}$$

Using (2.5) and Proposition 2.1, we obtain

$$H_\gamma(y, z) = \begin{cases} h_0(z) + \gamma \sum_{i=1}^m z_i q_* \left(\frac{y_i}{\gamma z_i} \right) & \text{if } (y, z) \in C, \\ -\infty & \text{otherwise,} \end{cases} \tag{2.10}$$

where $h_0(z) := \inf_x \ell_0(x, z)$ for $z \geq 0$ and $-\infty$ otherwise is the ordinary dual objective associated with problem (P). Hence the dual problem (2.9) is

$$(D_\gamma) \qquad \sup_{(y,z) \in C} \left\{ h_0(z) + \gamma \sum_{i=1}^m z_i q_* \left(\frac{y_i}{\gamma z_i} \right) \right\}. \tag{2.11}$$

3. Derivation of a differentiable dual problem. Given the dual problem (D_γ), consider the following *relaxed* dual problem:

$$(\hat{D}_\gamma) \quad \sup_{y>0} \sup_{z>0} \left\{ h_0(z) + \gamma \sum_{i=1}^m z_i q_* \left(\frac{y_i}{\gamma z_i} \right) \right\}. \tag{3.1}$$

We will prove (see Theorem 3.3(b)) that under a mild assumption on the data of problem (P), the relaxed dual problem has in fact the same optimal value and the same set of optimal solutions as the original dual problem (D_0), and therefore is a natural dual candidate.

Hidden in the formulation (3.1) is another dual problem given in terms of the dual variable y *only*, whose objective function $h_\gamma(y)$ is the optimal value function of the inner maximization problem in (3.1), i.e.,

$$h_\gamma(y) = \sup_{z>0} \left\{ h_0(z) + \gamma \sum_{i=1}^m z_i q_* \left(\frac{y_i}{\gamma z_i} \right) \right\}. \tag{3.2}$$

The main result of this paper is that the dual problem thus obtained,

$$(\hat{D}_\gamma) \qquad \sup\{h_\gamma(y) : \ y \in \mathbb{R}^m_{++}\},$$

has a *continuously differentiable* objective function on the positive orthant.

We first express the function h_γ in terms of the function

$$\varphi(t) := -t q_* \left(\frac{1}{t} \right), \tag{3.3}$$

and the rescaled variables $y_i \leftarrow \gamma^{-1} y_i$ as

$$h_\gamma(y) = \sup_{z>0} \left\{ h_0(z) - \gamma \sum_{i=1}^{m} y_i \varphi\left(\frac{z_i}{y_i}\right) \right\}. \tag{3.4}$$

The next result shows properties of the function φ inherited by those of q.

LEMMA 3.1. *If the function $q(\cdot)$ satisfies conditions* (Q1)–(Q4), *then the function φ defined in* (3.3) *is a strictly convex and differentiable function on the positive real line satisfying*

$$\varphi(1) = \varphi'(1) = 0, \tag{3.5}$$

$$\lim_{t \to 0^+} \varphi'(t) = -\infty. \tag{3.6}$$

Proof. The function q_* is concave and hence it is easy to verify that $tq_*\left(\frac{1}{t}\right)$ is concave for $t > 0$, so φ is convex (the strict convexity following from (3.9)). Since q is strictly concave on its domain $(a, +\infty)$, it is essentially strictly concave. Hence by [6, Thm. 26.3] its conjugate q_* is essentially smooth and by [6, Thm. 26.1] differentiable on $\mathbb{R}_{++}$; consequently $\varphi(t)$ is also differentiable for $t > 0$. Now, to prove (3.5), we compute

$$\begin{aligned} \varphi(1) &= -q_*(1) = -\inf(t - q(t)) = q(0) \quad \text{by (Q3)} \\ &= 0 \quad \text{by (Q2)}. \end{aligned}$$

Also

$$\varphi'(t) = -q_*\left(\frac{1}{t}\right) + \frac{1}{t} q_*'\left(\frac{1}{t}\right), \tag{3.7}$$

$$\begin{aligned} \varphi'(1) &= -q_*(1) + q_*'(1) \\ &= 0 + (q')^{-1}(1) = 0 \quad \text{by (Q3)}. \end{aligned}$$

To prove (3.6) we use a well-known relation for conjugate (concave) functions:

$$q(q_*'(s)) = s\, q_*'(s) - q_*(s) \tag{3.8}$$

(see [6, p. 257]). This combined with (3.7) implies

$$\varphi'(t) = q\left(q_*'\left(\frac{1}{t}\right)\right), \tag{3.9}$$

so

$$\varphi'(0) = q(q_*'(\infty)) = q((q')^{-1}(\infty)) = q(a) = -\infty.$$

Here we used the well-known relation $q_*' = (q')^{-1}$ and conditions (Q1), (Q4). □

Examples. For $q(t) = 1 - e^{-t}$, $\varphi(s) = s - \log s - 1$, $(s > 0)$, and for $q(t) = \log(1+t)$, $\varphi(s) = s \log s - s + 1$, $(s \geq 0)$.

Let Φ be the family of functions $\varphi : \mathbb{R}_{++} \to \mathbb{R}$ as in Lemma 3.1. Given $\varphi \in \Phi$, we define $d_\varphi : \mathbb{R}_+^n \times \mathbb{R}_{++}^n \to \mathbb{R}$ as

$$d_\varphi(x, y) = \sum_{j=1}^{n} y_j \varphi\left(\frac{x_j}{y_j}\right). \tag{3.10}$$

The function d_φ is called the φ-divergence and was introduced by Csiszar [2] as a measure of distance between probability measures. We use it here as a kind of "distance" between two vectors in $\mathbb{R}^n_+$. This is justified by the following facts:

(a) $d_\varphi(x, y) \geq 0$ and equality holds iff $x = y$,

(b) $d_\varphi(x, y)$ is *jointly convex* in (x, y),

which are easily verified from the properties of $\varphi \in \Phi$.

Using (3.10), the dual objective function (3.4) of $(\hat{D}_\gamma)$ can be written as

$$h_\gamma(y) = \sup_{z>0}\{h_0(z) - \gamma d_\varphi(z, y)\}. \tag{3.11}$$

The right-hand side resembles the *Moreau–Yosida smoothing* of the function h_0 [4] except that in (3.11) the usual quadratic distance is replaced by the φ-divergence d_φ. This type of smoothing was recently introduced and studied by Teboulle [9] under the name *entropic proximal regularization.*

Returning to the convex program (P), from now on we assume that problem (P) is feasible with a finite optimal value and the Slater constraint qualification holds, i.e., there exists $\hat{x} \in \mathbb{R}^n$ such that

$$g_i(\hat{x}) < 0, \quad i = 1, \ldots, m.$$

Under this assumption, the dual objective h_0 is a closed proper concave function and there exists an optimal Lagrange multiplier z^* solving the associated dual problem (D). Furthermore, one has

$$0 \in ri(\text{dom } (h_0)_*), \tag{3.12}$$

where $(h_0)_*$ denotes the (concave) conjugate of h_0.

To establish the last part of the proof of Theorem 3.3(b), we will also need the following assumption:

(A) There exists $\bar{z} > 0$ such that $\inf_x\{f(x) + \bar{z}^T g(x)\} > -\infty$.

Note that assumption (A) is a mild one. For example, in the case of a convex problem (P) with linear constraints $Ax \geq b$, assumption (A) is required to verify that there exists $\bar{z} > 0$, such that $A'z \in \text{dom } f^*$, and which is very similar to a standard constraint qualification for this class of problems, which would read as follows: there exists $\bar{z} \geq 0 : A'z \in ri \text{ dom } f^*$.

Before establishing the main result of the paper, we need the following technical result.

LEMMA 3.2. *Let H be a closed proper concave function on $\mathbb{R}^m$. Assume that the supremum of H over $\mathbb{R}^m_+$ is attained and that there exists $\bar{z} > 0$ such that $\bar{z} \in \text{dom } H$. Then*

$$\sup_{z \in \mathbb{R}^m_+} H(z) = \sup_{z \in \mathbb{R}^m_{++}} H(z).$$

Proof. Clearly we have

$$\sup_{z \geq 0} H(z) \geq \sup_{z>0} H(z). \tag{3.13}$$

Let z^* solve $\sup_{z\geq 0} H(z)$ and $z^* \not> 0$ (otherwise the result trivially holds). Consider the sequence $z_n \equiv (1-\frac{1}{n})z^* + \frac{1}{n}\bar{z}$, $n \geq 1$. Clearly $z_n > 0$, $z_n \in \text{dom } H$, and $z_n \to z^*$. Since H is concave, we have

$$H(z_n) \geq \left(1-\frac{1}{n}\right) H(z^*) + \frac{1}{n} H(\bar{z})$$

and hence

$$\liminf_{n\to\infty} H(z_n) \geq H(z^*). \tag{3.14}$$

But, H is closed concave (i.e., upper semicontinuous), which means

$$\limsup_{n\to\infty} H(z_n) \leq H(z^*). \tag{3.15}$$

Combining (3.14)–(3.15) thus proved that z_n is a supremizing sequence and $\sup_{z\geq 0} H(z) = \sup_{z>0} H(z)$. □

Remark 1. The assumption in Lemma 3.2 that the supremum of H over $\mathbb{R}^m_+$ is achieved is in fact redundant, but it is fulfilled in our application (see the proof of Theorem 3.3 (b)). Indeed, note that since dom H and $B := \mathbb{R}^m_{++}$ are convex and B is open, it is easy to check that the assumption dom $H \cap B \neq \emptyset$ is equivalent to $ri(\text{dom})H \cap B \neq \emptyset$. Then, invoking [6, Thm. 9.3], we get $\text{cl}(H - \delta_B) = H - \text{cl}(\delta_B) = H - \delta_{\text{cl}(B)}$, where cl denotes the (upper) closure of a function and δ_B the indicator function of the set B. Since the suprema of a proper function and its (upper) closure are equal, the result follows.

THEOREM 3.3. *Let $\varphi \in \Phi$. For every $\gamma > 0$ one has the following.*

(a) *The dual objective function h_γ of problem $(\hat{D}_\gamma)$ is concave and continuously differentiable on $\mathbb{R}^m_{++}$ with gradient given by*

$$[\nabla h_\gamma(y)]_i = \gamma\varphi^*\left(\varphi'\left(\frac{z^i_\varphi(y)}{y_i}\right)\right) \quad \forall\, y \in \mathbb{R}^m_{++},$$

where $z^i_\varphi(y)$ denotes the ith component of the unique point $z \equiv z_\varphi(y)$ at which the supremum in (3.11) *is attained.*

(b) *The dual problems (D_0) and $(\hat{D}_\gamma)$ have the same set of optimal solutions, and under assumption* (A) *have the same optimal value.*

Proof. Recall that the dual objective function h_0 is a concave function and d_φ is jointly convex in (z, y); thus, $h_0(z) - \gamma d_\varphi(z, y)$ is also jointly concave in (z, y). Hence, $h_\gamma(y)$ defined in (3.11), being the pointwise supremum of a jointly concave function, is concave in y (see [8, Thm. 1, p. 2]. Now, for every $y \in \mathbb{R}^m_{++}$, we have from (3.12) that

$$ri\ (\text{dom } h_0^*) \cap ri\ \text{dom } d^*_\varphi(\cdot, y)) \neq \emptyset.$$

(Here $d^*_\varphi(\cdot, y)$ denotes the conjugate of d_φ with respect to the first variable for fixed $y \in \mathbb{R}^m_{++}$, and it can be verified that $0 \in ri\,\text{dom} d^*_\varphi(\cdot, y)$.) Then, applying Fenchel's duality theorem [6, Thm. 31.1], one deduces that the supremum of $h_0(z) - \gamma d_\varphi(z, y)$ is finitely attained and thus the subdifferential set $\partial h_\gamma(y)$ is nonempty. Moreover, since $z \to -d_\varphi(z, y)$ is strictly concave, so is $h_0(z) - \gamma d_\varphi(z, y)$ and hence the supremum in (3.11) is uniquely attained at $z \equiv z_\varphi(y)$ and

$$h_\gamma(y) = h_0(z_\varphi(y)) - \gamma d_\varphi(z_\varphi(y), y). \tag{3.16}$$

For every $\hat{y} \in \mathbb{R}^n_{++}$

$$h_\gamma(\hat{y}) \geq h_0(z_\varphi(y)) - \gamma d_\varphi(z_\varphi(y), \hat{y}). \tag{3.17}$$

Combining (3.16) and (3.17), we obtain

$$-\gamma[d_\varphi(z_\varphi(y), \hat{y}) - d_\varphi(z_\varphi(y), y)] \leq h_\gamma(\hat{y}) - h_\gamma(y). \tag{3.18}$$

Let $w \in \partial h_\gamma(y)$. Then, for any $\hat{y} \in \mathbb{R}^m_{++}$ we have

$$h_\gamma(\hat{y}) - h_\gamma(y) \leq \langle w, \hat{y} - y \rangle, \tag{3.19}$$

which together with (3.18) implies $-\gamma[d_\varphi(z_\varphi(y), \hat{y}) - d_\varphi(z_\varphi(y), y)] \leq \langle w, \hat{y} - y \rangle$, i.e.,

$$d_\varphi(z_\varphi(y), \hat{y}) - d_\varphi(z_\varphi(y), y) \geq \langle -\gamma^{-1} w, \hat{y} - y \rangle$$

and therefore $-\gamma^{-1} w \in \partial_2 d_\varphi(z_\varphi(y), y)$, where ∂_2 denotes the subgradient with respect to the second variable of d_φ. Since φ is continuously differentiable on $\mathbb{R}_{++}$, $\partial_2 d_\varphi$ is a singleton and we have

$$w = -\gamma \nabla_2 d_\varphi(z_\varphi(y), y). \tag{3.20}$$

Consequently, h_γ is continuously differentiable for every $y \in \mathbb{R}^m_{++}$. Moreover, from (3.20) we obtain the following using the definition of d_φ:

$$\begin{aligned} w_i = [\nabla h_\gamma(y)]_i &= -\gamma \left\{ \varphi\left(\frac{z^i_\varphi(y)}{y_i} \right) - \frac{z^i_\varphi(y)}{y_i} \varphi'\left(\frac{z^i_\varphi(y)}{y_i} \right) \right\} \\ &= \gamma \varphi^*\left(\varphi'\left(\frac{z^i_\varphi(y)}{y_i} \right) \right), \end{aligned} \tag{3.21}$$

where in the later equality we use the well-known relation for conjugate functions [6, p. 257]:

$$\varphi(t) - t\varphi'(t) = -\varphi^*(\varphi'(t)).$$

We now turn to the proof of part (b) of the theorem.

Let Y_γ be the optimal solution set of problem $(\hat{D}_\gamma)$, i.e.,

$$Y_\gamma = \{y : 0 \in \partial h_\gamma(y)\} = \{y : \nabla h_\gamma(y) = 0\}.$$

Then using (3.21), $\bar{y} \in Y_\gamma$ if and only if

$$\varphi^*\left(\varphi'\left(\frac{z^i_\varphi(\bar{y})}{\bar{y}_i} \right) \right) = 0, \quad i = 1, \ldots, m. \tag{3.22}$$

From $\varphi \in \Phi$, it follows that

$$\varphi^*(s) = 0 \Leftrightarrow s = 0;$$

hence from (3.22)

$$\varphi'\left(\frac{z^i_\varphi(\bar{y})}{\bar{y}_i} \right) = 0$$

and using the fact that φ is strictly convex with $\varphi'(1) = 0$, we conclude that

$$\bar{y} \in Y_\gamma \text{ if and only if } z_\varphi^i(\bar{y}) = \bar{y}_i \ \forall i. \tag{3.23}$$

On the other hand, the optimality condition for $z_\varphi(\bar{y})$ is

$$\gamma\varphi'\left(\frac{z_\varphi^i(\bar{y})}{\bar{y}_i}\right) \in [\partial h_0(z_\varphi(\bar{y}))]_i \ \forall i. \tag{3.24}$$

Combining (3.23), (3.24), and $\varphi'(1) = 0$, it follows that

$$\begin{aligned}&\bar{y} \in Y_\gamma \text{ if and only if } 0 \in \partial h_0(\bar{y}),\\ &\text{i.e., if and only if } \bar{y} \text{ solves problem } (D_0).\end{aligned}$$

Finally, to prove the last part of (b), using the definition of h_γ given in (3.11), we have

$$\begin{aligned}\sup_{y>0} h_\gamma(y) &= \sup_{y>0}\sup_{z>0}\{h_0(z) - \gamma d_\varphi(z,y)\}\\ &= \sup_{z>0}\{h_0(z) - \gamma \inf_{y>0} d_\varphi(z,y)\}\\ &= \sup_{z>0} h_0(z) \qquad (3.25)\end{aligned}$$

since $d_\varphi(z,y) \geq 0$ and equals zero if and only if $z = y$. Invoking Lemma 3.2 with $H := h_0$ (which is closed proper concave under the Slater condition), the assumption that there exists $\bar{z} > 0$ such that $\bar{z} \in \operatorname{dom} H = \operatorname{dom} h_0$ is nothing else but assumption (A). Therefore we conclude that $\sup_{z>0} h_0(z) = \sup_{z\geq 0} h_0(z)$, and hence together with (3.25) $\sup(\hat{D}_\gamma) = \sup(D_0)$ is proved. □

Remark 2. If S is any locally simplicial subset of dom h_0, then by [6, Thm. 10.2, p. 84], h_0 is continuous relative to S, and assumption (A) in that case is redundant. The class of locally simplicial sets includes, for example, polytopes and polyhedral convex sets.

Acknowledgments. We thank the referees for helpful comments which improved the presentation of the results.

REFERENCES

[1] D. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[2] I. CSISZAR, *Information-type measures of difference of probability distributions and indirect observations*, Studia Sci. Math. Hungar., 2 (1967), pp. 299–318.

[3] E. G. GOL'SHTEIN AND N. V. TRET'YAKOV, *Modified Lagrangian Functions (Theory and Methods Optimization)*, Nauka, Moscow, USSR, 1989.

[4] J. J. MOREAU, *Proximité et dualité dans un espace Hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[5] R. A. POLYAK, *Modified barrier functions, theory and methods*, Math. Programming, 54 (1992), pp. 177–222.

[6] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[7] ———, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Math. Programming, 5 (1973), pp. 354–373.

[8] ———, *Conjugate Duality and Optimization*, CBMS, Vol. 16, Society for Industrial and Applied Mathematics Publications, Philadelphia, PA, 1974.

[9] M. TEBOULLE, *Entropic proximal mappings with application to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–690.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 626–637, August 1996
© 1996 Society for Industrial and Applied Mathematics
006

ASYMPTOTIC CONVERGENCE ANALYSIS OF SOME INEXACT PROXIMAL POINT ALGORITHMS FOR MINIMIZATION*

CIYOU ZHU†

Abstract. In this paper, we prove that the inexact proximal point algorithm (PPA) in the form of Bonnans–Gilbert–Lemaréchal–Sagastizábal's "general algorithmic pattern" (GAP–1) converges linearly under mild conditions. We also propose another variant (GAP–2) of inexact PPA that shares the same convergence property as GAP–1 but makes more sense numerically.

Based on this essential result, we further prove the linear convergence for the outer iteration of the bundle method without requiring the differentiability of the objective function or the uniqueness of the solution. We also prove the linear convergence for the outer iteration of Correa–Lemaréchal's "implementable form" of PPA and derive its rate.

Key words. proximal point algorithm, bundle method, convex minimization, linear convergence

AMS subject classifications. 65K05, 65K10, 90C20

1. Introduction.

Consider the convex programming problem

$$(\mathcal{P}) \qquad \text{minimize } f(x) \text{ over all } x \in \mathbb{R}^n,$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed proper convex function [8]. As a *blanket assumption*, we assume that the solution set of $(\mathcal{P})$, denoted by $\bar{X}$, is nonempty. Let $\langle \cdot, \cdot \rangle_M$ be the inner product associated with a symmetric positive definite matrix $M \in \mathbb{R}^{n \times n}$

$$\langle x, y \rangle_M = x^T M y, \tag{1.1}$$

where "T" indicates the transpose, and let $|\cdot|_M$ be the norm induced by this inner product. A general iteration scheme for solving $(\mathcal{P})$ is the proximal point algorithm (PPA) [5].

Proximal point algorithm (exact PPA). Specify a starting point x^0. For $k = 0, 1, 2, \ldots$, generate the sequence $\{x^k\}$ as

$$x^{k+1} := \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2} |x - x^k|_M^2 \right\}. \tag{1.2}$$

The PPA is especially useful as a "regularization" procedure for problems where the objective function is not strongly convex. Some commonly used minimization algorithms may lose their convergence properties or even become undefined when f is not strongly convex. With the PPA as an "outer loop" of the iteration, however, these algorithms can still be used in solving the "inner loop" subproblems of calculating the

*Received by the editors May 31, 1994; accepted for publication (in revised form) January 18, 1995. This research was supported by the Office of Scientific Computing, U. S. Department of Energy contract W-31-109-Eng-38, and National Science Foundation contract ASC-9213149.

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439. Current Address: Electrical Engineering and Computer Science Department, 2145 Sheridan Road, Northwestern University, Evanston, IL 60208 (ciyou@eecs.nwu.edu).

argmin on the right-hand side of (1.2). The objective functions in these subproblems are strongly convex because of the augmented term $\frac{1}{2}|x - x^k|_M^2$.

In a more general sense, the PPA can be formulated as a procedure for solving the multivalue equation $0 \in T_M(x)$ related to a *maximal monotone operator* (associated with the M-inner product) $T_M : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ [9]. The operator

$$J_M = (I + T_M)^{-1} \tag{1.3}$$

is single valued from all of $\mathbb{R}^n$ into $\mathbb{R}^n$ [7]. Hence the iteration

$$x^{k+1} := J_M(x^k) \tag{1.4}$$

is well defined. Moreover, $J_M(x) = x$ if and only if $0 \in T_M(x)$. The iteration (1.4) converges to a solution of $0 \in T_M$ [5, 6]. Let T_M be the subdifferential mapping $\partial_M f$ related to f in the M-inner product. Then problem $(\mathcal{P})$ is equivalent to the multivalue equation $0 \in T_M(x)$, and iteration (1.4) reduces precisely to (1.2). See [9] for detailed discussions.

In notation, we shall always use subscript "M" to indicate any concept associated with the inner product $\langle \cdot, \cdot \rangle_M$, such as $|\cdot|_M$, $\partial_M f$, $\mathbb{B}_M(z, \delta)$ (the closed ball centered at z with radius δ), and $\mathrm{dist}_M(\cdot, \cdot)$ (the distance function). When M is the identity matrix I, this M-inner product reduces to the ordinary Euclidean inner product and the subscript will be dropped. To simplify the notation, the ordinary Euclidean inner product will underlie most parts of the paper unless otherwise specified.

One problem in the implementation of PPA is the calculation of argmin on the right-hand side of (1.2). Usually, it is also an iterative process. Since this subproblem must be solved for each outer-loop iteration, it is computationally expensive to take x^{k+1} as the exact argmin in (1.2). In [9], Rockafellar proposed a more realistic version of the PPA, where an approximation, instead of the exact argmin in (1.2), is taken as the next iterate x^{k+1}.

Asymptotically accurate proximal point algorithm (see [9]). Choose a positive sequence $\{\delta_k\}$ such that $\sum \delta_k < +\infty$. Specify a starting point $x^0 \in X$. For $k = 0, 1, 2, \ldots$, generate the sequence $\{x^k\}$ in such a way that

$$|x^{k+1} - \bar{x}^{k+1}| \leq \delta_k, \tag{1.5}$$

where

$$\bar{x}^{k+1} := \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2}|x - x^k|^2 \right\}. \tag{1.6}$$

This iteration scheme converges to an optimal solution of $(\mathcal{P})$ [9]. If, in addition, x^{k+1} satisfies

$$|x^{k+1} - \bar{x}^{k+1}| \leq \delta_k |x^{k+1} - x^k|, \tag{1.7}$$

then the convergence will be linear under the assumptions that the optimal solution set of $(\mathcal{P})$ is a singleton and that the inverse of the mapping $T = \partial f$ is Lipschitz continuous in some neighborhood of 0. Other convergence results regarding the asymptotically accurate PPA may be found in Luque [4], where conditions similar to (1.7) are also imposed.

Unfortunately, the criterion (1.7) with $\sum \delta_k < +\infty$ will eventually drive

$$\frac{|x^{k+1} - \bar{x}^{k+1}|}{|x^{k+1} - x^k|} \to 0 \quad \text{as } k \to \infty.$$

Hence, the subproblem of finding the argmin in (1.6) will still have to be solved to high accuracy, thus posing potential difficulty for the inner loop. Intuitively it might not be wise to spend much computational effort in solving a single subproblem to very high accuracy while leaving the "base point" x^k of the inner-loop iteration unchanged.

For any fixed $z \in \mathbb{R}^n$, let $\tilde{f}_M(\cdot; z) : \mathbb{R}^n \to \mathbb{R}$ be the function defined as

$$\tilde{f}_M(x; z) = f(x) + \frac{1}{2}|x - z|_M^2. \tag{1.8}$$

Denote

$$\tilde{f}^*_{M,z} = \min_{x \in \mathbb{R}^n} \tilde{f}_M(x; z). \tag{1.9}$$

In [1], Bonnans, Gilbert, Lemaréchal, and Sagastizábal proposed a "general algorithmic pattern" (GAP–1) for inexact PPA, where f is simply required to decrease by a fixed fraction of $f(z) - \tilde{f}^*_{M,z}$ in the inner iteration based at z. With the simplified notation

$$\tilde{f}_k(x) = \tilde{f}_{M_k}(x; x^k) \quad \text{and} \quad \tilde{f}^*_k = \min_{x \in \mathbb{R}^n} \tilde{f}_k(x) \tag{1.10}$$

for any given M_k and x^k, GAP–1 can be written in the following form.

General algorithmic pattern – 1 (GAP–1) (see [1]). Choose some descent parameter $\eta \in (0, 1)$. Specify a starting point x^0 and a symmetric positive definite matrix M^0. For $k = 0, 1, 2, \ldots$, generate the sequence $\{x^k\}$ as follows. If $x^k \in \bar{X}$, then stop; otherwise do the following.

Step 1. Compute x^{k+1} satisfying

$$\frac{f(x^{k+1}) - \tilde{f}^*_k}{f(x^k) - \tilde{f}^*_k} \le \eta. \tag{1.11}$$

Step 2. Choose a symmetric positive definite matrix M_{k+1}.

The convergence of GAP–1 was established in [1, Thm. 2.3]. Let $\lambda_{\max}(M_k)$ be the largest eigenvalue of M_k.

THEOREM 1 (convergence of GAP–1; (see [1, Thm. 2.3]). *Assume the solution set $\bar{X}$ of $(\mathcal{P})$ is nonempty and bounded. Then the sequence $\{x^k\}$ generated by GAP–1 is bounded. Moreover, if*

$$\sum_{k=1}^{\infty} \left(\lambda_{\max}(M_k)\right)^{-1} = \infty, \tag{1.12}$$

then any accumulation point of $\{x^k\}$ belongs to $\bar{X}$. The same properties also hold for the sequence $\{\bar{x}^k\}$ of the minimizer in (1.10).

However, two important questions remain unanswered.

(i) How fast would the inexact PPA in the pattern of GAP–1 converge?

(ii) How much computational effort does it cost numerically to meet the inner-loop stopping criterion (1.11) of GAP–1?

In this paper, we analyze the asymptotic rate of convergence for several inexact PPAs, where our results on GAP–1 will play a central role. Before proceeding further, we first propose another variant of inexact PPA, from which the answer to the second question will become clearer.

General algorithmic pattern – 2 (GAP–2). Choose an inner-loop algorithm for minimizing the regularized objective function $\tilde{f}(\cdot; z)$ for fixed z as well as a descent parameter $\eta \in (0, 1)$. Specify a starting point x^0 and a symmetric positive definite matrix M^0. For $k = 0, 1, 2, \ldots$, generate the sequence $\{x^k\}$ as follows. If $x^k \in \bar{X}$, then stop; otherwise do the following.

Step 1. Minimize $\tilde{f}(y; x^k)$ starting from $y^{k,0} = x^k$ to generate $y^{k,1}, y^{k,2}, \ldots$, until a point y^{k,m_k} satisfying

$$\frac{\tilde{f}_k(y^{k,m_k}) - \tilde{f}_k^*}{\tilde{f}_k(y^{k,0}) - \tilde{f}_k^*} \leq \eta \tag{1.13}$$

is obtained; take

$$x^{k+1} = y^{k,m_k}. \tag{1.14}$$

Step 2. Choose a symmetric positive definite matrix M_{k+1}.

Now the stopping criterion (1.13) in GAP–2 is related to the values of the regularized functions $\tilde{f}_k$ only. Obviously, this criterion can be satisfied in a finite number of iterations if the inner-loop algorithm itself has at least a linear rate on $\tilde{f}_k$. Therefore, people can make the decision as whether to terminate the inner-loop iteration by simply counting the number of iterations. On the other hand, by noting that $\tilde{f}_k(y^{k,0}) = f(x^k)$ and $\tilde{f}_k(y^{k,m_k}) = \tilde{f}_k(x^{k+1}) \geq f(x^{k+1})$, it is easy to see that the inner-loop stopping criterion of GAP–2 implies (1.11) in GAP–1. Hence GAP–2 shares the same convergence property of GAP–1, as stated in Theorem 1.

A prominent feature of GAP–1 and GAP–2 is that no asymptotically stringent stopping criteria were imposed on the inner-loop iteration. It is clear from the stopping criterion (1.13) that the regularized subproblems only need to be solved to finite precision constantly. An even more surprising fact is that both GAP–1 and GAP–2 will converge linearly under mild local growth conditions on the inverse of the subgradient mapping ∂f. We will prove this result in §2 without any differentiability assumption or strong convexity assumption on the objective function f.

As applications of this essential result, we will further prove the linear convergence for the outer iteration of the bundle method and for the outer iteration of Correa–Lemaréchal's "implementable form" of PPA [2] in the last two sections of the paper.

2. Rate of convergence for GAP–1 and GAP–2. In this section, we investigate the rate of convergence for the sequence $\{f(x^k)\}$ generated by either GAP–1 or GAP–2. We show that they converge at least linearly with a rate related to the descent parameter η as well as a certain growth condition on the inverse of the subdifferential mapping ∂f near 0.

Keep the notation developed in §1. Let f^* be the optimal value of $(\mathcal{P})$:

$$f^* = f(\bar{X}). \tag{2.1}$$

Denote

$$\bar{x}^{k+1} = \operatorname*{argmin}_{x \in \mathbb{R}^n} \tilde{f}_k(x). \tag{2.2}$$

The following lemma gives a general result on the relationship between the outer-loop residue $f(x^k) - f^*$ and the inner-loop residue $\tilde{f}_k(x^k) - \tilde{f}_k^*$.

LEMMA 2 (relation between inner- and outer-loop residues). *Let $\{x^k\}$ be a sequence in $\mathbb{R}^n \setminus \bar{X}$ such that $\operatorname{dist}(\bar{X}, x^k) \to 0$ as $k \to \infty$. Let $\{M_k\}$ be a sequence of symmetric positive definite matrices in $\mathbb{R}^{n\times n}$ with*

$$\lambda_{\max}(M_k) \le \Lambda_{\max} \tag{2.3}$$

for some $\Lambda_{\max} > 0$ and sufficiently large k. Suppose there exist constants $\alpha \ge 0$, $\delta > 0$, and $r \ge 1$ such that the inverse of the subdifferential mapping $T = \partial f$ satisfies the growth condition

$$\forall w \in \mathbb{B}(0,\delta), \quad \forall x \in T^{-1}(w), \qquad \operatorname{dist}(\bar{X}, x) \le \alpha |w|^r. \tag{2.4}$$

If the sequence $\{\lambda_{\min}(M_k)\}$ has a lower bound $\Lambda_{\min} > 0$, then

$$\tilde{f}_k(x^k) - \tilde{f}_k^* \le f(x^k) - f^* \le 2\big(1 + \alpha\Lambda_{\max}^r \operatorname{dist}^{r-1}(\bar{X}, x^k)\big)\big(\tilde{f}_k(x^k) - \tilde{f}_k^*\big) \tag{2.5}$$

for sufficiently large k. Hence for any $\varepsilon > 0$, there exists k' such that for all $k \ge k'$

$$f(x^k) - f^* \le \begin{cases} 2(1 + \alpha\Lambda_{\max})\big(\tilde{f}_k(x^k) - \tilde{f}_k^*\big) & \text{if } r = 1, \\ 2(1+\varepsilon)\big(\tilde{f}_k(x^k) - \tilde{f}_k^*\big) & \text{if } r > 1. \end{cases} \tag{2.6}$$

Proof. The first half of (2.5) follows directly from $\tilde{f}_k(x^k) = f(x^k)$ and $\tilde{f}_k^* \ge f^*$. Now we prove the second half of (2.5). Recall that $\bar{x}^{k+1} = J_{M_k}(x^k)$ and $\bar{x}^{k+1} \ne x^k$ unless $x^k \in \bar{X}$, where $J_{M_k} = (I + T_{M_k})^{-1}$ is the proximal mapping related to the operator $T_{M_k} = \partial_{M_k} f$. According to Güler [3, Lem. 2.2],

$$f(\bar{x}^{k+1}) - f(z) \le \frac{1}{2}|z - x^k|_{M_k}^2 - \frac{1}{2}|z - \bar{x}^{k+1}|_{M_k}^2 - \frac{1}{2}|x^k - \bar{x}^{k+1}|_{M_k}^2 \quad \forall z. \tag{2.7}$$

For any $\bar{x} \in \bar{X}$, $z = \bar{x}$ in (2.7) yields

$$\tilde{f}_k(\bar{x}^{k+1}) - f(\bar{x}) \le \frac{1}{2}|x^k - \bar{x}|_{M_k}^2 - \frac{1}{2}|\bar{x}^{k+1} - \bar{x}|_{M_k}^2. \tag{2.8}$$

On the other hand, $z = x^k$ in (2.7) yields

$$\tilde{f}_k(x^k) - \tilde{f}_k(\bar{x}^{k+1}) \ge \frac{1}{2}|x^k - \bar{x}^{k+1}|_{M_k}^2. \tag{2.9}$$

Combining (2.8) and (2.9), we get

$$\frac{\tilde{f}_k(\bar{x}^{k+1}) - f(\bar{x})}{\tilde{f}_k(x^k) - \tilde{f}_k(\bar{x}^{k+1})} \le \frac{|x^k - \bar{x}|_{M_k}^2 - |\bar{x}^{k+1} - \bar{x}|_{M_k}^2}{|x^k - \bar{x}^{k+1}|_{M_k}^2}. \tag{2.10}$$

But

$$\begin{aligned} \frac{|x^k - \bar{x}|_{M_k}^2 - |\bar{x}^{k+1} - \bar{x}|_{M_k}^2}{|x^k - \bar{x}^{k+1}|_{M_k}^2} &= \frac{|x^k - \bar{x}^{k+1}|_{M_k}^2 + 2\langle x^k - \bar{x}^{k+1}, \bar{x}^{k+1} - \bar{x}\rangle_{M_k}}{|x^k - \bar{x}^{k+1}|_{M_k}^2} \\ &\le \frac{|x^k - \bar{x}^{k+1}|_{M_k}^2 + 2|x^k - \bar{x}^{k+1}|_{M_k}|\bar{x}^{k+1} - \bar{x}|_{M_k}}{|x^k - \bar{x}^{k+1}|_{M_k}^2} \\ &= \frac{|x^k - \bar{x}^{k+1}|_{M_k} + 2|\bar{x}^{k+1} - \bar{x}|_{M_k}}{|x^k - \bar{x}^{k+1}|_{M_k}}. \end{aligned} \tag{2.11}$$

Substituting (2.11) in (2.10), we get

$$\frac{\tilde{f}_k(\bar{x}^{k+1}) - f(\bar{x}_)}{\tilde{f}_k(x^k) - \tilde{f}_k(\bar{x}^{k+1})} \leq \frac{|x^k - \bar{x}^{k+1}|_{M_k} + 2|\bar{x}^{k+1} - \bar{x}|_{M_k}}{|x^k - \bar{x}^{k+1}|_{M_k}}$$

for any $\bar{x} \in \bar{X}$. Hence

$$\frac{\tilde{f}_k(\bar{x}^{k+1}) - f^*}{\tilde{f}_k(x^k) - \tilde{f}_k(\bar{x}^{k+1})} \leq 1 + \frac{2\mathrm{dist}_{M_k}(\bar{X}, \bar{x}^{k+1})}{|x^k - \bar{x}^{k+1}|_{M_k}}. \tag{2.12}$$

Now, according to [9, Prop. 1],

$$|J_{M_k}(z) - J_{M_k}(z')|^2_{M_k} + |(I - J_{M_k})(z) - (I - J_{M_k})(z')|^2_{M_k} \leq |z - z'|_{M_k}, \tag{2.13}$$

$$(I - J_{M_k})(z) \in T_{M_k}\big(J_{M_k}(z)\big) \tag{2.14}$$

for all z and z'. Let $z = x^k$ and $z' = \bar{x}$ in (2.13) and (2.14). By (2.13),

$$|\bar{x}^{k+1} - \bar{x}|^2_{M_k} + |x^k - \bar{x}^{k+1}|^2_{M_k} \leq |x^k - \bar{x}|^2_{M_k} \quad \forall \bar{x} \in \bar{X}.$$

Hence

$$|x^k - \bar{x}^{k+1}|_{M_k} \leq \mathrm{dist}_{M_k}(\bar{X}, x^k) \leq \Lambda_{\max}^{\frac{1}{2}} \mathrm{dist}(\bar{X}, x^k). \tag{2.15}$$

Therefore

$$|x^k - \bar{x}^{k+1}| \leq \Lambda_{\min}^{\frac{1}{2}} |x^k - \bar{x}^{k+1}|_{M_k} \to 0 \quad \text{as } \mathrm{dist}(\bar{X}, x^k) \to 0.$$

But $x^k - \bar{x}^{k+1} \in T_{M_k}(\bar{x}^{k+1})$ by (2.14). Then

$$f(z) \geq f(\bar{x}^{k+1}) + \langle x^{k+1} - \bar{x}^{k+1}, z - \bar{x}^{k+1} \rangle_{M_k} \quad \forall z$$

or, equivalently,

$$f(z) \geq f(\bar{x}^{k+1}) + \langle M^k(x^{k+1} - \bar{x}^{k+1}), z - \bar{x}^{k+1} \rangle \quad \forall z,$$

which yields $M^k(x^k - \bar{x}^{k+1}) \in T(\bar{x}^{k+1})$ or, in other words,

$$\bar{x}^{k+1} \in T^{-1}\big(M^k(x^k - \bar{x}^{k+1})\big).$$

Hence it follows from (2.4) that

$$\Lambda_{\max}^{-\frac{1}{2}} \mathrm{dist}_{M_k}(\bar{X}, \bar{x}^{k+1}) \leq \mathrm{dist}(\bar{X}, \bar{x}^{k+1}) \leq \alpha |M^k(x^k - \bar{x}^{k+1})|^r \leq \alpha \Lambda_{\max}^{\frac{r}{2}} |(x^k - \bar{x}^{k+1})|^r_{M^k} \tag{2.16}$$

for sufficiently large k. Combining (2.16) and (2.12), we obtain

$$\frac{\tilde{f}_k(\bar{x}^{k+1}) - f^*}{\tilde{f}_k(x^k) - \tilde{f}_k(\bar{x}^{k+1})} \leq 1 + 2\alpha \Lambda_{\max}^{\frac{r+1}{2}} |x^k - \bar{x}^{k+1}|^{r-1}_{M_k},$$

from which the second half of (2.5) follows by (2.15). □

Lemma 2 may be viewed as a quantization of the relationship

$$J(x) = x \iff 0 \in T(x).$$

The result can also be used in the optimality test of the algorithm. If an estimation on $\tilde{f}_k(x^k) - \tilde{f}_k^*$ for the subproblem is available, then an estimation on $f(x^k) - f^*$ for the original problem can be obtained via (2.6).

The growth condition (2.4) is in fact a local upper Lipschitz condition with modulus α and order r on T^{-1}. It is a common requirement when convergence rates are under consideration. Both Rockafellar [9] and Luque [4] used similar conditions in their analysis of PPAs. Actually, the condition in [9] on the convergence of the asymptotically accurate PPA implies the growth condition with $r = 1$. See Zhu [10] for more discussion on the growth conditions.

THEOREM 3 (rate of convergence for GAP–1 and GAP–2). *Assume that the solution set of* $(\mathcal{P})$ *is nonempty and bounded and there exist constants* $\alpha \geq 0$, $\delta > 0$, *and* $r \geq 1$ *such that the inverse of the subdifferential mapping* $T = \partial f$ *satisfies the growth condition*

$$\forall w \in \mathbb{B}(0,\delta) \quad \forall x \in T^{-1}(w), \qquad \operatorname{dist}(\bar{X}, x) \leq \alpha |w|^r. \tag{2.17}$$

Let $\{x^k\} \subset \mathbb{R}^n \setminus \bar{X}$ *be a sequence generated by GAP–1 (or GAP–2). If the eigenvalues of the symmetric positive definite matrices* $\{M_k\}$ *in the algorithm satisfy*

$$\Lambda_{\max} \geq \lambda_{\max}(M_k) \geq \lambda_{\min}(M_k) \geq \Lambda_{\min} > 0 \tag{2.18}$$

with some $\Lambda_{\max} > \Lambda_{\min} > 0$ *for sufficiently large* k, *then the sequence* $\{x^k\}$ *satisfies*

$$\frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{1-\eta}{2\big(1 + \alpha \Lambda_{\max}^r \operatorname{dist}^{r-1}(\bar{X}, x^k)\big)} \tag{2.19}$$

for sufficiently large k, *where* $\eta \in (0,1)$ *is the descent parameter in the stopping criterion of the algorithm. Hence the sequence* $\{f(x^k)\}$ *of the objective value converges to the optimal value of* $(\mathcal{P})$ *linearly in the sense that*

$$\frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{1-\eta}{2(1 + \alpha \Lambda_{\max}^r)} \tag{2.20}$$

for all sufficiently large k, *and*

$$\limsup_{k \to \infty} \frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq \frac{1+\eta}{2} \quad \text{if } r > 1. \tag{2.21}$$

Proof. We only need to prove the theorem for GAP–1 since the stopping criterion (1.13) in GAP–2 implies the stopping criterion (1.11) in GAP–1. Observe that $\operatorname{dist}(\bar{X}, x^k) \to 0$ as $k \to \infty$ by Theorem 1. Now we prove the conclusions on the rate of convergence by using Lemma 2.

It follows from (1.11) that

$$f(x^k) - f(x^{k+1}) \geq (1-\eta)\big(\tilde{f}_k(x^k) - \tilde{f}_k^*\big) \tag{2.22}$$

for sufficiently large k. However, by the second half of (2.5),

$$f(x^k) - f^* \leq \big(2 + (2\alpha \Lambda_{\max}^r) \operatorname{dist}^{r-1}(\bar{X}, x^k)\big)\big(\tilde{f}_k(x^k) - \tilde{f}_k^*\big) \tag{2.23}$$

for sufficiently large k. Observe that $f(x^k) - f^* > 0$ for $x^k \notin \bar{X}$. Hence dividing (2.22) by (2.23), we have

$$\frac{f(x^k) - f(x^{k+1})}{f(x^k) - f^*} \geq \frac{1-\eta}{2\big(1 + \alpha\Lambda_{\max}^r \mathrm{dist}^{r-1}(\bar{X}, x^k)\big)}$$

for sufficiently large k, which yields (2.19). The inequalities (2.20) and (2.21) follow from (2.19) immediately. □

According to Theorem 3, a linear rate of the inner-loop iteration on the "regularized" subproblem will induce a linear rate on the outer-loop iteration, provided the growth condition (2.17) is satisfied. Hence, algorithms possessing a linear or superlinear rate of convergence on "regular" problems can be extended to solve the "singular" problems via the inexact PPA.

Theorem 3 also throws light on the choice of the *proximal parameter* c when M_k in the algorithm takes the form $M_k = \frac{1}{c}I$. For $r = 1$, which is the common case corresponding to an upper Lipschitz continuous inverse of the subdifferential mapping ∂f in some neighborhood of 0, the rate given by (2.20) depends on the proximal parameter c as well as on the inner-loop rate represented by η. Once the inner-loop algorithm is chosen, the relationship between η and c can be determined. Then the selection of c can be optimized by minimizing the right-hand side of (2.20).

3. Linear convergence on the proximal form of bundle methods. In this section, we assume the objective function f is finite valued. Correa–Lemaréchal's proximal form of bundle methods [2] for finding the minimum of f may be written as follows.

Proximal form of bundle methods (see [2]). Choose some tolerance $\mu \in (0,1)$ and a positive sequence $\{c_k\}$. Specify a starting point x^0 and a convex function $\varphi^0 \leq f$. Set $k = j = 0$.

Step 1. Solve for y in

$$\min\left\{\varphi^j(y) + \frac{1}{2c_k}|y - x^k|^2\right\} \tag{3.1}$$

to obtain the unique optimal solution y^j as well as $\gamma^j := (x^k - y^j)/c_k \in \partial\varphi^j(y^j)$.

Step 2. Compute $f(y^j)$. If

$$f(x^k) - f(y^j) \geq \mu[f(x^k) - \varphi^j(y^j)], \tag{3.2}$$

then take a *descent-step* by letting $x^{k+1} := y^j$ and $k := k+1$; otherwise take a *null-step* by doing nothing here.

Step 3. If a descent-step was made in Step 2, then choose a convex function $\varphi^{j+1} : \mathbb{R}^n \to \mathbb{R}$ satisfying $\varphi^{j+1} \leq f$; otherwise choose φ^{j+1} satisfying

$$\varphi^{j+1} \leq f, \tag{3.3}$$

$$\varphi^j(y^j) + \langle \gamma^j, \cdot - y^j\rangle \leq \varphi^{j+1}, \tag{3.4}$$

$$f(y^j) + \langle g^j, \cdot - y^j\rangle \leq \varphi^{j+1}, \tag{3.5}$$

where $g^j \in \partial f(y^j)$. Go to Step 1 with $j := j + 1$.

Correa and Lemaréchal proved in [2, Thm. 4.4] that if c_k is bounded and $\sum c_k = +\infty$, then this iteration scheme

(i) either reaches some $x^k \in \bar{X}$ in finite iterations

(ii) or generates $\{x^k\}$ converging to some $\bar{x} \in \bar{X}$.

Now we prove further that if the proximal parameters c_k are bounded away from zero, then the outer iteration of the algorithm converges at least linearly for the problems satisfying the growth conditions (2.17) with a nonempty and bounded solution set.

PROPOSITION 4 (rate of convergence on Correa–Lemaréchal's bundle methods). *Suppose the optimal solution set $\bar{X}$ is nonempty and bounded and that $x^k \notin \bar{X}$ in Correa–Lemaréchal's proximal form of bundle methods for all k. Assume that there exist constants $\alpha \geq 0$, $\delta > 0$, and $r \geq 1$ such that the inverse of the subdifferential mapping $\mathcal{T} = \partial f$ satisfies the growth condition* (2.17) *in Theorem* 3. *If there exist positive constants $c_{\max}$ and $c_{\min}$ such that*

$$c_{\max} \geq c_k \geq c_{\min} > 0 \tag{3.6}$$

for sufficiently large k, then the sequence $\{x^k\}$ generated by Correa–Lemaréchal's proximal form of the bundle method satisfies

$$\frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{\mu}{2\big(1 + (\alpha/c_{\min}^r)\mathrm{dist}^{r-1}(\bar{X}, x^k)\big)} \tag{3.7}$$

for all sufficiently large k. Hence the sequence $\{f(x^k)\}$ of the objective value converges to the optimal value f^* linearly as

$$\frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{\mu}{2(1 + \alpha/c_{\min}^r)} \tag{3.8}$$

for all sufficiently large k, and

$$\limsup_{k\to\infty} \frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{\mu}{2} \quad \text{if } r > 1. \tag{3.9}$$

Proof. According to [1, Prop. 2.2] condition (3.2) implies (1.11). Hence the iteration falls in the category of GAP–1 with $M_k = I/c_k$, and the conclusion follows immediately from Theorem 3. □

4. Linear convergence on Correa–Lemaréchal's implementable PPA. In this section, we prove the linear convergence for the outer iteration of Correa–Lemaréchal's "implementable form" of PPA [2, Alg. 3.3]. Let $\partial_\varepsilon f$ denote the ε-subdifferential of f. Recall that

$$g \in \partial_\varepsilon f(x) \iff f(z) \geq f(x) + \langle z - x, g\rangle - \varepsilon \quad \forall\, z.$$

With the notation in this paper, Correa–Lemaréchal's implementable form of PPA for minimizing $f(x)$ goes as follows.

Correa–Lemaréchal's implementable form of PPA (see [2]). Choose a positive sequence $\{c_k\}$ and an algorithm to generate a minimizing sequence $\{w^j\}$ of $\tilde{f}(\cdot; z)$ for fixed z. Choose $\kappa > 1$ and $\mu \in (0, 1)$. Start from x^0, set $k = 0$.

Step 1. Set $j = 0$; start from some $w^{k,j} = w^{k,0}$.

Step 2. Set

$$\varepsilon(x^k, w^{k,j}) = \kappa\left[f(x^k) - f(w^{k,j}) - \frac{\mu}{c_k}|w^{k,j} - x^k|^2\right]. \tag{4.1}$$

If

$$\frac{x^k - w^{k,j}}{c_k} \in \partial_{\varepsilon(x^k, w^{k,j})} f(x^k), \tag{4.2}$$

then go to Step 3; otherwise compute $w^{k,j+1}$, increase j by 1, and execute Step 2 again.

Step 3. Set $x^{k+1} = w^{k,j}$, increase k by 1, and loop to Step 1.

Correa and Lemaréchal pointed out in [2] that (4.1) and (4.2) together imply

$$\varepsilon(x^k, w^{k,j}) = \kappa \left[f(x^k) - f(w^{k,j}) - \frac{\mu}{c_k} |w^{k,j} - x^k|^2 \right] \geq 0. \tag{4.3}$$

They also proved that if f is strongly coercive, that is,

$$f(z)/|z| \to +\infty \text{ when } |z| \to \infty, \tag{4.4}$$

then

(i) for each k, Step 2 eventually exit to Step 3, unless x^k is already optimal [2, Thm. 3.2];

(ii) $\liminf_{k\to\infty} f(x^k) = f^*$ [2, Prop. 3.2].

Here we prove further that, if the proximal parameters c_k are bounded above and bounded away from zero, then the outer iteration of the algorithm converges at least linearly for the problems satisfying the growth conditions (2.17) with a nonempty and bounded solution set.

PROPOSITION 5 (rate of convergence on Correa–Lemaréchal's form of PPA). *Suppose the optimal solution set $\bar{X}$ is nonempty and bounded and that $x^k \notin \bar{X}$ in Correa–Lemaréchal's implementable PPA for all k. Assume that f is strongly coercive in the sense of* (4.4), *and there exist constants $\alpha \geq 0$, $\delta > 0$, and $r \geq 1$ such that the inverse of the subdifferential mapping $T = \partial f$ satisfies the growth condition* (2.17) *in Theorem* 3. *If there exist positive constants $c_{\max}$ and $c_{\min}$ such that*

$$c_{\max} \geq c_k \geq c_{\min} > 0 \tag{4.5}$$

for sufficiently large k, then the sequence $\{x^k\}$ generated by Correa–Lemaréchal's implementable PPA satisfies

$$\frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{1}{(2\kappa + 1/\mu)\left(1 + (\alpha/c_{\min}^r)\operatorname{dist}^{r-1}(\bar{X}, x^k)\right)} \tag{4.6}$$

for sufficiently large k. Hence the sequence $\{f(x^k)\}$ of the objective value converges to the optimal value f^* linearly as

$$\frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{1}{(2\kappa + 1/\mu)(1 + \alpha/c_{\min}^r)} \tag{4.7}$$

for all sufficiently large k, and

$$\limsup_{k\to\infty} \frac{f(x^{k+1}) - f^*}{f(x^k) - f^*} \leq 1 - \frac{1}{2\kappa + 1/\mu} \quad \text{if } r > 1. \tag{4.8}$$

Proof. We prove the proposition by showing that the iteration scheme falls in the category of GAP–1 with

$$\eta = 1 - \frac{1}{\kappa + 1/(2\mu)} \quad \text{and} \quad M_k = \frac{1}{c_k} I. \tag{4.9}$$

Then the conclusions follow directly from Theorem 3.

Observe that (4.2) in the algorithm with $\varepsilon(x^k, w^j)$ given by (4.1) yields

$$\begin{aligned} f(z) \geq & f(x^k) + \frac{1}{c_k}\langle x^k - x^{k+1}, z - x^k\rangle - \varepsilon(x^k, x^{k+1}) \quad \forall\, z \\ = & f(x^k) + \frac{1}{c_k}\langle x^k - x^{k+1}, z - x^k\rangle - \kappa\left[f(x^k) - f(x^{k+1}) - \frac{\mu}{c_k}|x^{k+1} - x^k|^2\right] \quad \forall\, z \end{aligned}$$

or, equivalently,

$$\begin{aligned} \tilde{f}_k(z) \geq & f(x^k) + \frac{1}{2}|z - x^k|_{M_k} + \langle x^k - x^{k+1}, z - x^k\rangle_{M_k} \\ & - \kappa\big[f(x^k) - f(x^{k+1}) - \mu|x^{k+1} - x^k|^2_{M_k}\big] \quad \forall\, z, \end{aligned}$$

where $M_k = I/c_k$ and $\tilde{f}_k(z) = \tilde{f}_{M_k}(z; x^k)$. Let $z = \bar{x}^{k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^n} \tilde{f}_k(x)$. We have

$$\begin{aligned} \tilde{f}_k(\bar{x}^{k+1}) \geq & f(x^k) + \frac{1}{2}|\bar{x}^{k+1} - x^k|_{M_k} + \langle x^k - x^{k+1}, \bar{x}^{k+1} - x^k\rangle_{M_k} \\ & - \kappa\big[f(x^k) - f(x^{k+1}) - \mu|x^{k+1} - x^k|^2_{M_k}\big]. \end{aligned}$$

Therefore, with the notation $\tilde{f}_k^* = \min_{x\in\mathbb{R}^n} \tilde{f}_k(x)$,

$$\begin{aligned} f(x^k) - \tilde{f}_k^* \leq & \kappa\big[f(x^k) - f(x^{k+1}) - \mu|x^{k+1} - x^k|^2_{M_k}\big] \\ & - \frac{1}{2}|\bar{x}^{k+1} - x^k|_{M_k} - \langle x^k - x^{k+1}, \bar{x}^{k+1} - x^k\rangle_{M_k}. \end{aligned} \tag{4.10}$$

However

$$|\bar{x}^{k+1} - x^k|^2_{M_k} + 2\langle x^k - x^{k+1}, \bar{x}^{k+1} - x^k\rangle_{M_k} = -|x^k - x^{k+1}|^2_{M_k} + |\bar{x}^{k+1} - x^{k+1}|^2_{M_k}. \tag{4.11}$$

Substituting (4.11) in (4.10), we obtain

$$\begin{aligned} f(x^k) - \tilde{f}_k^* \leq & \kappa\big[f(x^k) - f_k(x^{k+1}) - \mu|x^{k+1} - x^k|^2_{M_k}\big] \\ & + \frac{1}{2}|x^{k+1} - x^k|^2_{M_k} - \frac{1}{2}|\bar{x}^{k+1} - x^{k+1}|^2_{M_k}. \end{aligned}$$

Hence, for $\mu \in (0, 1)$, there holds

$$f(x^k) - \tilde{f}_k^* \leq \kappa\big[f(x^k) - f(x^{k+1})\big] + \frac{1}{2}|x^{k+1} - x^k|^2_{M_k}. \tag{4.12}$$

On the other hand,

$$f(x^k) - f(x^{k+1}) - \mu|x^{k+1} - x^k|^2_{M_k} \geq 0$$

by (4.3). Hence

$$\frac{1}{\mu}\big(f(x^k) - f_k(x^{k+1})\big) \geq |x^{k+1} - x^k|^2_{M_k}. \tag{4.13}$$

Substituting (4.13) in (4.12), we obtain

$$f(x^k) - \tilde{f}_k^* \leq \left(\kappa + \frac{1}{2\mu}\right)\big[f(x^k) - f(x^{k+1})\big].$$

Then

$$\frac{f(x^k) - \tilde{f}_k^*}{f(x^k) - f(x^{k+1})} \leq \kappa + \frac{1}{2\mu}$$

or, equivalently,

$$\frac{f(x^{k+1}) - \tilde{f}_k^*}{f(x^k) - \tilde{f}_k^*} \leq 1 - \frac{1}{\kappa + 1/(2\mu)}.$$

Therefore condition (1.11) in GAP–1 holds with η given by (4.9). □

Acknowledgments. The author is indebted to Professor C. Lemaréchal for his insightful comments and encouragement. The preprints sent to the author by him were very helpful in preparing the revised version of this paper.

REFERENCES

[1] J. F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *A family of variable metric proximal methods*, Math. Programming, 68 (1995), pp. 15–47.

[2] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Programming, 62 (1993), pp. 261–275.

[3] O. Güler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[4] F. J. Luque, *Asymptotic convergence analysis of the proximal point algorithm*, SIAM J. Control Optim., 22 (1984), pp. 277–293.

[5] B. Martinet, *Regularisation d'inéquations variationelles par approximations successives*, Rev. Française Inf. Rech. Oper. (1970), pp. 154–159.

[6] ———, *Determination approchée d'un point fixe d'une application pseudo-contractante*, C. R. Acad. Sci. Paris Sér. I Math., 274 (1972), pp. 163–165.

[7] G. J. Minty, *Monotone (nonlinear) operators in Hilbert space*, Duke Math. J., 29 (1962), pp. 341–346.

[8] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[9] ———, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[10] C. Zhu, *Asymptotic convergence analysis of the forward-backward splitting algorithm*, Math. Oper. Res., 20 (1995), pp. 449–464.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 638–652, August 1996
© 1996 Society for Industrial and Applied Mathematics
007

COMPLEXITY ANALYSIS OF AN INTERIOR CUTTING PLANE METHOD FOR CONVEX FEASIBILITY PROBLEMS*

JEAN-LOUIS GOFFIN†, ZHI-QUAN LUO‡, AND YINYU YE§

Dedicated to the memory of Gyorgy Sonnevend.

Abstract. We further analyze the convergence and the complexity of a dual column generation algorithm for solving general convex feasibility problems defined by a separation oracle. The oracle is called at an *approximate* analytic center of the set given by the intersection of the linear inequalities which are the previous answers of the oracle. We show that the algorithm converges in finite time and is in fact a fully polynomial approximation algorithm, provided that the feasible region has a nonempty interior.

Key words. convex feasibility problem, potential reduction, column generation, cutting planes

AMS subject classifications. 90C25, 90C26, 90C60

1. Introduction.

The problem studied is that of finding an interior point in a convex set Γ, where we assume $\Gamma \subset \mathsf{R}^m$ has a nonempty interior and is contained in the cube $\Omega^0 = \{y \in \mathsf{R}^m : 0 \le y \le e\} = [0,1]^m$.

The set Γ is defined implicitly by a *separating oracle* which for every $\bar{y} \in \Omega^0$ either answers that $\bar{y}$ belongs to $\mathrm{int}(\Gamma)$ or generates a separating hyperplane $\{y \in \mathsf{R}^m : a^T y \le a^T \bar{y}\} \supset \Gamma$. Without loss of generality, we assume that a is normalized so that $\|a\| = 1$.

The problem of finding an interior point to a convex set defined by a system (finite or infinite) of convex inequalities

$$\Gamma = \{y \in \mathsf{R}^m : f_i(y) \le 0, \quad i = 1, 2, \ldots\},$$

where each $f_i : \mathsf{R}^m \mapsto \mathsf{R}$ is convex, can be cast in this manner. In particular, the separating oracle just needs to select a to be $g_i/\|g_i\|$, where g_i is an arbitrary subgradient of any of the functions f_i satisfying $f_i(\bar{y}) \ge 0$, i.e., $g_i \in \partial f_i(\bar{y})$ (the subdifferential of f_i). Note that for any $g_i \in \partial f_i(\bar{y})$, $f_i(y) \le f_i(\bar{y})$ implies $g_i^T(y - \bar{y}) \le 0$. Thus, if $f_i(\bar{y}) \ge 0$, $f_i(y) \le 0$, then $g_i^T(y - \bar{y}) \le 0$. In fact, the requirement that $f_i(\bar{y})$ be computed exactly and that $g_i \in \partial f_i(\bar{y})$ can be significantly relaxed. All that is required from the oracle is that the following query be answered: find any i such that $f_i(\bar{y}) \ge 0$ and a such that $a^T(y - \bar{y}) \le 0$ for all $y \in \{y \in \mathsf{R}^m : f_i(y) \le 0\}$. This also works if the functions f_i are pseudoconvex.

The problem under investigation may be cast as that of finding the solution to an infinite system of linear inequalities, which is defined implicitly by the oracle.

* Received by the editors November 18, 1993; accepted for publication (in revised form) March 29, 1995.

† Faculty of Management, McGill University, 1001 Sherbrooke Street West, Montreal, Quebec H3A 1G5, Canada (ma56@musica.mcgill.ca). The research of this author was supported by Natural Sciences and Engineering Research Council of Canada grant OPG0004152 and by the FCAR of Quebec.

‡ Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario L8S 4L7, Canada. The research of this author was supported by Natural Sciences and Engineering Research Council of Canada grant OPG0090391.

§ Department of Management Sciences, The University of Iowa, Iowa City, IA 52242. The research of this author was supported by National Science Foundation grant DDM-9207347, an Iowa Business School summer grant, and K. C. Wong Educational Foundation, Hong Kong, through Academia Sinica, Beijing, China.

The column generation method of Dantzig-Wolfe [3] or its dual equivalent, Kelley's cutting plane method [9], could be used. The classical centering methods that have been suggested for the above convex feasibility problem include the center of gravity method of Levin [12], the largest sphere method of Elzinga and Moore [5], the ellipsoid method of Yudin and Nemirovsky [16, 10, 18], the maximum volume ellipsoid method of Tarasov, Khacian, and Erlich [20, 11], and the method of volumetric centers of Vaidya [21], among others.

Sonnevend [19]; Goffin, Haurie, and Vial [7]; and Ye [22] proposed a column generation—or cutting plane—algorithm that computes $\bar{y}$ as the analytic center of the system of inequalities generated so far. In [22], Ye showed that this algorithm is a potential-reduction algorithm for solving the linear feasibility problem where $\Gamma = \{y \in \mathsf{R}^m : A^T y \leq c\}$ for some $A \in \mathsf{R}^{m \times d}$ and $c \in \mathsf{R}^d$. If d is finite, the algorithm finds a feasible solution in polynomial time of n and d. This algorithm was developed and tested for a variety of large-scale problems when d is infinite, where it performed quite well (see, for example, Burke et al. [2]; Goffin, Haurie, and Vial [7]; Mitchell [13]; and Mitchell and Todd [14]); but proofs of convergence and of complexity remained at large.

Recently, Atkinson and Vaidya developed a modified version of the algorithm that uses a column elimination scheme and proved a polynomial complexity result [1]; Nesterov [17] proved a complexity result for the minimization of a convex function using a variant of the algorithm which uses a potential function consisting of the logarithm of a quadratic term representing the initial sphere localization.

In this paper, we show that for any given convex feasibility problem with nonempty interior, the algorithm of [7, 22] is a fully polynomial approximation scheme for the original version of the algorithm that uses only linear inequalities to approximate the solution set and does not use column elimination. A fully polynomial approximation scheme means that for every ϵ, the accuracy at termination, the method is polynomial in the dimension of the problem m.

2. Preliminaries. Before we use approximate analytic centers, we need to develop some technical results to be used in the subsequent analysis. We start with a measure of proximity to the analytic center. Again, let Ω be a (bounded) polytope in R^m defined by n ($> m$) linear inequalities; i.e.,

$$\Omega = \{y \in \mathsf{R}^m : c - A^T y \geq 0\}.$$

Recall that for a point y in the interior of Ω, the potential function is

$$\phi(y) := \sum_{j=1}^{n} \ln s_j,$$

where

$$s = c - A^T y,$$

and the max-potential of Ω is defined as the value of ϕ at the analytic center $\bar{y}$:

$$P(\Omega) := \max_{y \in \Omega} \phi(y).$$

Consider a normalized gradient vector of ϕ at y:

$$\Delta s(y) = S^{-1} A^T (AS^{-2}A^T)^{-1} \nabla \phi(y) = -S^{-1} A^T (AS^{-2}A^T)^{-1} AS^{-1} e$$

and

$$\delta(y) = \|\Delta s(y)\| = \|S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}e\|$$

or

$$\delta(y)^2 = e^T S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}e.$$

Note that $AS^{-2}A^T$ is symmetric positive definite, where S (or X) represents the diagonal matrix of vector s (or x). Thus, this quantity $\delta(y)$ is a norm of the gradient vector. Let

$$x(y) = S^{-1}(I - S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1})e;$$

we have

$$\Delta s(y) = Sx(y) - e \quad \text{and} \quad \delta(y) = \|Sx(y) - e\|.$$

Another measure would explicitly use the primal variable $x \geq 0$ and $Ax = 0$, the Lagrangian multiplier for Ω, where Ω is usually called the dual system. For a primal feasible x, a residual is defined as

$$\eta(x, y) = \|X(c - A^T y) - e\| = \|Xs - e\|.$$

We now establish some interesting inequalities relating $\delta(y)$ to $\eta(x, y)$. Some of these inequalities will be used in the subsequent analysis. Most results of this lemma were proved in Gonzaga [8] and den Hertog, Roos, and Vial [4].

LEMMA 2.1. *Let (y, s) be an interior point in Ω.*

(i) *If there is $x > 0$ with $Ax = 0$ such that $\eta(x, y) < 1$ then*

$$\delta(y) \leq \eta(x, y).$$

Conversely, if $\delta(y) < 1$ then there is $x > 0$ with $Ax = 0$ such that

$$\eta(x, y) = \delta(y).$$

(ii) *Let $\Delta(y) = (AS^{-2}A^T)^{-1}\nabla\phi(y)$ and let $y^+ = y + \Delta(y)$. Then if $\delta(y) < 1$,*

$$\delta(y^+) \leq \delta(y)^2.$$

(iii) *Let $\bar{y}$ be the analytic center of Ω. Then if $\delta(y) < 1$,*

$$\phi(\bar{y}) - \phi(y) \leq \frac{\delta(y)^2}{1 - \delta(y)^2}$$

or

$$P(\Omega) - \phi(y) \leq \frac{\delta(y)^2}{1 - \delta(y)^2}.$$

(iv) *Let $\bar{s} = c - A^T\bar{y}$ where $\bar{y}$ is the analytic center of Ω. Then if $\delta(y) < 1$,*

$$\|S^{-1}\bar{s} - e\| \leq \frac{\delta(y)}{1 - \delta(y)}.$$

Proof. (i) Given (y, s) in the interior of Ω, we can verify that $\delta(y)$ is the minimal value and $x(y)$ is the minimal solution of the least squares problem

$$\begin{array}{ll} \text{minimize} & \|Sx - e\| \\ \text{subject to} & Ax = 0. \end{array}$$

Since x is a feasible point for this problem, we must have

$$\delta(y) \leq \eta(x, y).$$

Conversely, let $x = x(y)$. Then we have $Ax = 0$ and $\eta(x, y) = \delta(y)$. Moreover, $x > 0$ since $\delta(y) < 1$.

(ii) Define $s^+ = s - S\Delta s(y)$ and note that from the proof of (i)

$$\delta(y^+) = \|S^+ x(y^+) - e\| \leq \|S^+ x(y) - e\|.$$

But

$$\begin{aligned} \|S^+ x(y) - e\|^2 &= \|(2S - S^2 X(y))x(y) - e\|^2 \\ &= \sum_{j=1}^{n} (s_j x(y)_j - 1)^4 \\ &\leq \left(\sum_{j=1}^{n} (s_j x(y)_j - 1)^2 \right)^2 \\ &= \|Sx(y) - e\|^4 \\ &= \delta(y)^4. \end{aligned}$$

(iii) Let y^+ be generated in (ii). Since ϕ is a concave function,

$$\phi(y^+) - \phi(y) \leq \nabla\phi(y)(y^+ - y) = \delta(y)^2.$$

Consider the sequence generated by this process with $y^0 = y$ and $y^{k+1} = y^k + \Delta(y^k)$. Then from (ii), $y^k \to \bar{y}$ and from the above relation,

$$\phi(y^{k+1}) - \phi(y^k) \leq \delta(y^k)^2.$$

Thus, for any $k \geq 1$ we have

$$\phi(y^k) - \phi(y^0) \leq \sum_{j=1}^{k} \delta(y)^{2^j} \leq \sum_{j=1}^{k} (\delta(y)^2)^j \leq \frac{\delta(y)^2}{1 - \delta(y)^2}.$$

(iv) Let $y^0 = y$ and the sequence $\{y^k, s^k\}$ be generated by the process in (iii). Then we show by induction that

$$\|(S^0)^{-1} s^k - e\| \leq \sum_{j=1}^{2^k - 1} \delta(y)^j.$$

Obviously, this relation is true for $k = 1$. Now, assume it is true for k. Note that

$$s^{k+1} = s^k - S^k (S^k x(y^k) - e).$$

We have

$$\begin{aligned}\|(S^0)^{-1}(s^{k+1}-s^k)\| &= \|(S^0)^{-1}S^k(S^k x(y^k)-e)\| \\ &\le \|(S^0)^{-1}S^k\|\|S^k x(y^k)-e\| \\ &\le \left(\sum_{j=0}^{2^k-1}\delta(y)^j\right)\|S^k x(y^k)-e\| \\ &\le \left(\sum_{j=0}^{2^k-1}\delta(y)^j\right)\delta(y)^{2^k}.\end{aligned}$$

Thus,

$$\begin{aligned}\|(S^0)^{-1}s^{k+1}-e)\| &\le \|(S^0)^{-1}s^k-e)\| + \|(S^0)^{-1}(s^{k+1}-s^k)\| \\ &\le \sum_{j=1}^{2^k-1}\delta(y)^j + \left(\sum_{j=0}^{2^k-1}\delta(y)^j\right)\delta(y)^{2^k} \\ &= \sum_{j=1}^{2^{k+1}-1}\delta(y)^j.\end{aligned}$$

This proves (iv). □

3. Newton procedures for computing centers. We review two Newton procedures to compute the analytic center. Either of them can be used in our cutting plane algorithm.

3.1. Dual Newton procedure. The dual Newton iteration for finding the analytic center is as follows: given an interior point $y \in \mathsf{R}^m$, slacks $s = c - A^T y$, and a primal feasible $x = x(y)$ satisfying

$$\|Xs - e\| \le \eta < 1, \tag{3.1}$$

we can apply the dual Newton method to generate a new iterate

$$y^+ = y + \Delta(y),$$

where $\Delta(y)$ is given by part (ii) of Lemma 2.1. The residual at (x^+, y^+, s^+), where $s^+ = c - A^T y^+$ and $x^+ = x(y^+)$, satisfies the following lemma.

LEMMA 3.1. *Let (x^+, y^+, s^+) be generated as above. Then*

$$\|X^+ s^+ - e\| \le \eta^2.$$

3.2. Primal-dual Newton procedure. Given an interior point $y \in \mathsf{R}^m$, $s = c - A^T y$, and a primal feasible $x \in \mathsf{R}^n$ satisfying

$$\|Xs - e\| \le \eta < 1, \tag{3.2}$$

we can apply the primal-dual Newton method to generate a new iterate x^+ and (y^+, s^+) as follows.

Solve for d_x, d_y, and d_s from the system of linear equations

$$\begin{aligned} Sd_x + Xd_s &= \tfrac{x^T s}{n}e - Xs, \\ Ad_x &= 0, \\ A^T d_y + d_s &= 0. \end{aligned} \tag{3.3}$$

Then let

$$x^+ = x + d_x, \quad y^+ = y + d_y, \quad s^+ = s + d_s.$$

We have a lemma for the residual at (x^+, y^+, s^+).

LEMMA 3.2. *Let (x^+, y^+, s^+) be generated as above. Then*

$$\|X^+ s^+ - e\| \leq \frac{\sqrt{2}\eta^2}{4(1-\eta)}.$$

Proof. See Mizuno [15].

4. Max-potential reduction. Now we use an approximate center y^k to generate a cut, where (y^k, s^k) is an interior point in Ω and there is $x^k > 0$ with $Ax^k = 0$ such that

$$\|X^k s^k - e\| \leq \eta \tag{4.1}$$

for some $0 < \eta < 1$. Let us place a cut exactly at y^k; that is, we add a new inequality $a^T y \leq a^T y^k$ to Ω and consider the new set

$$\Omega^+ = \{y : A^T y \leq c, \quad a^T y \leq a^T y^k\}.$$

We remark that our definition of Ω^+ is different from that of [22], which requires the new inequality to be $a^T y \leq a^T y^k + \beta r$, for some $0 < \beta < 1$, where

$$r = \sqrt{a^T (A(S^k)^{-2} A^T)^{-1} a}.$$

Thus, "more" is cut away from Ω in the current definition of Ω^+. This deep cut was first developed by Nesterov in his cutting plane method [17].

We now prove a lemma resembling Theorem 2 of Ye [22].

LEMMA 4.1. *Let $(\bar{y}, \bar{s})$ be the analytic center of Ω and let*

$$\bar{r} = \sqrt{a^T (A\bar{S}^{-2} A^T)^{-1} a}.$$

Then

$$P(\Omega^+) \leq P(\Omega) + \ln(\bar{r}) - \alpha$$

for some constant α depending only on η. Moreover, if $0 < \eta \leq 1/100$, then we have $\alpha > 0$.

Proof. Let $\bar{y}$ and $\bar{y}^+$ be the analytic centers for Ω and Ω^+, respectively. Let $\bar{s}^+ = c - A^T \bar{y}^+ > 0$ and $\bar{s}^+_{n+1} = a^T y^k - a^T \bar{y}^+$. Then we have

$$\begin{aligned}
\bar{s}^+_{n+1} &= a^T (y^k - \bar{y}^+) \\
&= a^T (A\bar{S}^{-2} A^T)^{-1} (A\bar{S}^{-2} A^T)(y^k - \bar{y}^+) \\
&= a^T (A\bar{S}^{-2} A^T)^{-1} A\bar{S}^{-2} (A^T y^k - A^T \bar{y}^+) \\
&= a^T (A\bar{S}^{-2} A^T)^{-1} A\bar{S}^{-2} (-c + A^T y^k + c - A^T \bar{y}^+) \\
&= a^T (A\bar{S}^{-2} A^T)^{-1} A\bar{S}^{-2} (\bar{s}^+ - s^k) \\
&= a^T (A\bar{S}^{-2} A^T)^{-1} A\bar{S}^{-1} (\bar{S}^{-1}\bar{s}^+ - \bar{S}^{-1} s^k) \\
&\leq \|a^T (A\bar{S}^{-2} A^T)^{-1} A\bar{S}^{-1}\| \|\bar{S}^{-1}\bar{s}^+ - \bar{S}^{-1} s^k\| \\
&= \bar{r} \|\bar{S}^{-1}\bar{s}^+ - e + e - \bar{S}^{-1} s^k\| \\
&\leq \bar{r} (\|\bar{S}^{-1}\bar{s}^+ - e\| + \|e - \bar{S}^{-1} s^k\|) \\
&\leq \bar{r} \left(\|\bar{S}^{-1}\bar{s}^+ - e\| + \frac{\eta}{(1-\eta)^2} \right) \quad \text{(from Lemma 2.1).}
\end{aligned}$$

Thus,

$$\frac{\exp P(\Omega^+)}{\bar{r}\exp P(\Omega)} = \frac{\bar{s}^+_{n+1}}{\bar{r}}\prod_{j=1}^{n}\frac{\bar{s}^+_j}{\bar{s}_j}$$
$$\leq \left(\|\bar{S}^{-1}\bar{s}^+ - e\| + \frac{\eta}{(1-\eta)^2}\right)\prod_{j=1}^{n}\frac{\bar{s}^+_j}{\bar{s}_j}.$$

Note that we still have, as in [22],

$$e^T\bar{S}^{-1}\bar{s}^+ = n.$$

This can be seen as follows:

$$\begin{aligned} e^T\bar{S}^{-1}\bar{s}^+ &= e^T\bar{X}\bar{s}^+ \\ &= e^T\bar{X}(c - A^T\bar{y}^+) \\ &= e^T\bar{X}c \\ &= e^T\bar{X}(c - A^T\bar{y}) \\ &= e^T\bar{X}\bar{s} \\ &= e^Te \\ &= n, \end{aligned}$$

where the first and sixth equalities are due to $\bar{X}\bar{s} = e$ and the second and fourth equalities follow from $A\bar{x} = 0$.

Thus, from the proof of Theorem 2 of [22],

$$\left(\|\bar{S}^{-1}\bar{s}^+ - e\| + \frac{\eta}{(1-\eta)^2}\right)\prod_{j=1}^{n}\frac{\bar{s}^+_j}{\bar{s}_j} \leq 4\exp\left(\frac{8\eta-3}{2(1-\eta)^2}\right).$$

Consequently,

$$P(\Omega^+) - P(\Omega) \leq \ln(\bar{r}) + \ln(4) + \frac{8\eta-3}{2(1-\eta)^2}.$$

Let

$$\alpha = -\ln(4) - \frac{8\eta-3}{2(1-\eta)^2}.$$

Let $\eta = 1/100$. Then we have $\alpha > 0$ and the desired result. □

5. Update to a new approximate center. In this section, we show how to construct a pair (x, y, s) from (x^k, y^k, s^k) such that $(A, a)x = 0$ with $x > 0$, (y, s) is in the interior of Ω^+, and

$$\|Xs - e\| < \eta < 1.$$

Suppose a pair (x^k, y^k, s^k) is given which satisfies (4.1). We use the *dual* scaling for the construction of (x, y, s). Let

$$r^k = \sqrt{a^T(A(S^k)^{-2}A^T)^{-1}a}, \tag{5.1}$$

$$\Delta y = -(\beta/r^k)(A(S^k)^{-2}A^T)^{-1}a, \tag{5.2}$$

$$\Delta s = (\beta/r^k)A^T(A(S^k)^{-2}A^T)^{-1}a, \tag{5.3}$$

$$\Delta x = -(\beta/r^k)(S^k)^{-2}A^T(A(S^k)^{-2}A^T)^{-1}a. \tag{5.4}$$

Then we set

$$y = y^k + \Delta y$$

and

$$x = \begin{bmatrix} x^k + \Delta x \\ \beta/r^k \end{bmatrix} = \begin{bmatrix} x^k - (\beta/r^k)(S^k)^{-2}A^T(A(S^k)^{-2}A^T)^{-1}a \\ \beta/r^k \end{bmatrix}.$$

It can be readily verified that

$$s = \begin{bmatrix} c - A^T(y^k + \Delta y) \\ a^T y^k - a^T(y^k + \Delta y) \end{bmatrix} = \begin{bmatrix} s^k + \Delta s \\ \beta r^k \end{bmatrix} = \begin{bmatrix} s^k + (\beta/r^k)A^T(A(S^k)^{-2}A^T)^{-1}a \\ \beta r^k \end{bmatrix}.$$

First, we have

$$(A, a)x = Ax^k - (\beta/r^k)a + (\beta/r^k)a = 0.$$

Second, we have

$$s^k + (\beta/r^k)A^T(A(S^k)^{-2}A^T)^{-1}a = (S^k)(e + p^k)$$

and

$$x^k - (\beta/r^k)(S^k)^{-2}A^T(A(S^k)^{-2}A^T)^{-1}a = (S^k)^{-1}(X^k S^k e - p^k),$$

where

$$p^k = (\beta/r^k)(S^k)^{-1}A^T(A(S^k)^{-2}A^T)^{-1}a.$$

Note that we have

$$\|p^k\| = \beta. \tag{5.5}$$

On the other hand, we have

$$x_j^k s_j^k \geq 1 - \eta.$$

Thus, if we select η and β such that

$$1 - \eta - \beta > 0, \tag{5.6}$$

then

$$s = \begin{bmatrix} (S^k)(e + p^k) \\ \beta r^k \end{bmatrix} > 0$$

and

$$x = \begin{bmatrix} (S^k)^{-1}(X^k S^k e - p^k) \\ \beta/r^k \end{bmatrix} > 0.$$

A simple calculation yields

$$Xs - e = \begin{bmatrix} X^k s^k - e \\ 0 \end{bmatrix} - \begin{bmatrix} (p^k)^2 \\ 1 - \beta^2 \end{bmatrix} + \begin{bmatrix} (X^k S^k - I)p^k \\ 0 \end{bmatrix},$$

where the vector

$$(p^k)^2 = ((p_1^k)^2, (p_2^k)^2, \ldots, (p_n^k)^2)^T.$$

Therefore, we have

$$\begin{aligned}\|Xs - e\| &\le \|X^k s^k - e\| + \sqrt{\|(p^k)^2\|^2 + (1-\beta^2)^2} + \|X^k s^k - e\|\|p^k\| \\ &\le \eta + \sqrt{\beta^4 + (1-\beta^2)^2} + \eta\beta,\end{aligned}$$

where the last step follows from (4.1) and (5.5). Let $\beta = 1/\sqrt{2}$ and $\eta = 0.15$; then

$$\|Xs - e\| \le \gamma = 0.15 + 1/\sqrt{2} + 0.15/\sqrt{2} < 1.$$

Or, we can let

$$\eta = 1/100 \quad \text{and have} \quad \gamma := 1/100 + 1.01/\sqrt{2} < 1.$$

Furthermore, it can also be easily verified that (5.6) holds.

Hence, using this (y, s) as a starting pair, we can apply the *dual* Newton procedure of §3.1 to generate a pair (y^{k+1}, s^{k+1}) and $x^{k+1} = x(y^{k+1})$ such that

$$\begin{aligned}(A, a)x^{k+1} &= 0, \quad x^{k+1} > 0, \\ s^{k+1} &= (c^T, a^T y^k)^T - (A, a)^T y^{k+1} > 0,\end{aligned}$$

and

$$\|X^{k+1}s^{k+1} - e\| \le \eta.$$

By Lemma 3.2 and the above-given values of γ and η, this can be accomplished in four *dual* Newton steps due to the fact that $\gamma^{16} \le \eta$. This column generation process can be repeated, and from Lemma 4.1 the nested sequence of polyhedral sets Ω^k generated by the algorithm satisfies

$$P(\Omega^{k+1}) \le P(\Omega^k) + \ln(\bar{r}^k) - \alpha, \tag{5.7}$$

where α is some constant,

$$\bar{r}^k = \sqrt{a_{k+1}^T (A(\bar{S}^k)^{-2} A^T)^{-1} a_{k+1}},$$

$(\bar{y}^k, \bar{s}^k)$ is the analytic center of Ω^k, and a_{k+1} is the cut generated at the kth iteration. Note that $(\bar{y}^k, \bar{s}^k)$ is solely used for analysis, and the algorithm does not need any knowledge of $(\bar{y}^k, \bar{s}^k)$.

We remark that, instead of using the *dual* Newton iterations of §3.1, one can use the *primal-dual* Newton iterations prescribed in §3.2 to obtain an approximate analytic center. Such an updating procedure works with primal as well as dual variables.

Finally, the updating step (5.1)–(5.4) uses dual scaling. Alternatively, we could use the *primal* scaling or the *primal-dual* scaling. Specifically, in the *primal* scaling we use $A(X^k)^2 A^T$ to replace $A(S^k)^{-2}A^T$ in (5.1)–(5.4), while in the *primal-dual* scaling we use $A(S^k)^{-1}X^k A^T$ to replace $A(S^k)^{-2}A^T$. The proofs and the conditions on η and β are similar; in particular, they both require four Newton iterations to recenter.

6. The cutting plane method from analytic centers: Convergence and complexity. The analysis of Goffin, Luo, and Ye [6] relies on two things: a bound identical to (5.7) and the assumption that the original feasibility problem has an interior point. Using the analysis above, we show that the technical results of [6] hold without much change for the algorithm using approximate centers. Thus, we derive the identical iteration complexity for the algorithm, and each iteration uses a constant number of primal-dual Newton steps. We extend here the proof of [6] for completeness.

Let the solution set Γ be contained in $\Omega^0 = \{y \in \mathsf{R}^m : 0 \leq y \leq e\}$ and $\text{int}(\Gamma)$ contain a full-dimensional closed ball with $\epsilon < \frac{1}{2}$ radius. We also assume that there exists an oracle which for every $\bar{y} \in \Omega^0$ either returns that $\bar{y} \in \text{int}(\Gamma)$ or generates a separating hyperplane $\{y : a^T y \leq a^T \bar{y}\} \supset \Gamma$, with $\|a\| = 1$ being assumed.

The interior-point column generation—or cutting plane—algorithm [19, 7, 22] from approximate analytic centers is as follows:

- **Step 0.**
  Let

$$A^0 = (I, -I) \in \mathsf{R}^{m \times 2m},$$

$$c^0 = \begin{pmatrix} e \\ 0 \end{pmatrix} \in \mathsf{R}^{2m},$$

$$y^0 = \frac{1}{2}e \in \mathsf{R}^m,$$

$$s^0 = c^0 - (A^0)^T y^0 = \frac{1}{2}e \in \mathsf{R}^{2m},$$

$$x^0 = 2e \in \mathsf{R}^{2m}$$

  and

$$k = 1.$$

- **Step 1.**
  Compute (y^k, s^k, x^k) with y^k an η-approximate analytic center of $\Omega^k = \{y \in \mathsf{R}^m : c^k - (A^k)^T y \geq 0\}$, using either the primal-dual procedure or the dual procedure of §3.2 with the updating scheme of §5, with the appropriate values of η, starting at $(y^{k-1}, s^{k-1}, x^{k-1})$, an η-approximate "primal-dual" center of Ω^{k-1}.
  Query the oracle to see if $y^k \in \text{int}(\Gamma)$ or not.
  If yes, stop; otherwise generate a hyperplane $\{y : a_{k+1}^T y \leq a_{k+1}^T y^k\} \supset \Gamma$ with $\|a_{k+1}\| = 1$, and let

$$\Omega^{k+1} = \{y \in \mathsf{R}^m : c^{k+1} - (A^{k+1})^T y \geq 0\},$$

  where

$$A^{k+1} = (A^k, a_{k+1})$$

  and

$$c^{k+1} = \begin{pmatrix} c^k \\ a_{k+1}^T y^k \end{pmatrix}.$$

- **Step 2.** Set $k := k + 1$ and return to Step 1.

Let the potential function computed at the *exact* analytic center $\bar{y}^k$ be

$$P(\Omega^k) = \sum_{j=1}^{2m+k} \ln(c^k - (A^k)^T \bar{y}^k)_j.$$

Clearly, the following relations hold, provided that termination has not occurred:

$$\Gamma \subset \Omega^k \quad \forall k, \tag{6.1}$$

$$P(\Omega^{k+1}) \leq P(\Omega^k) + \frac{1}{2}\ln(\bar{r}^k)^2 - \alpha \quad \text{(by Lemma 4.1)}, \tag{6.2}$$

where

$$(\bar{r}^k)^2 = a_{k+1}^T (A^k (\bar{S}^k)^{-2} (A^k)^T)^{-1} a_{k+1}$$

and

$$\bar{s}^k = c^k - (A^k)^T \bar{y}^k, \ \bar{S}^k = \operatorname{diag}(\bar{s}^k).$$

LEMMA 6.1. *For all* $k \geq 0$,

$$P(\Omega^k) \geq (2m + k) \ln \epsilon.$$

Proof. From (6.1), $\Gamma \subset \Omega^k$. Thus Ω^k contains a full-dimensional ball with ϵ radius. Let the center of this ball be $\bar{y}$.

Then $c^k - (A^k)^T \bar{y} \geq \epsilon e$; thus

$$P(\Omega^k) = \sum_{j=1}^{2m+k} \ln(c^k - (A^k)^T \bar{y}^k)_j \geq \sum_{j=1}^{2m+k} \ln(c^k - (A^k)^T \bar{y})_j \geq \sum_{j=1}^{2m+k} \ln \epsilon,$$

where $\bar{y}^k$ denotes the analytic center of Ω^k. □

LEMMA 6.2. *Let* $s = c^k - (A^k)^T y$ *for any* $y \in \Omega^k$. *Then*

1. $0 \leq s_j \leq 1, \quad j = 1, \ldots, 2m,$
2. $0 \leq s_j \leq \sqrt{m}, \quad j = 2m + 1, \ldots, 2m + k.$

Proof. The proof given here is the same as [6]; we include it here for completeness.

For $j = 1, \ldots, m$, $s_j = 1 - y_j$, since $0 \leq y_j \leq 1$, $0 \leq s_j \leq 1$.

For $j = m + 1, \ldots, 2m$, $s_j = y_{j-m}$, since $0 \leq y_{j-m} \leq 1$, $0 \leq s_j \leq 1$.

For $j = 2m + 1, \ldots, 2m + k$,

$$s_j = a_{j-2m}^T y^{j-2m} - a_{j-2m}^T y \leq \|a_{j-2m}\| \ \|y^{j-2m} - y\| = \|y^{j-2m} - y\| \leq \sqrt{m}.$$

The last inequality is due to the fact that $0 \leq y^{j-2m} \leq e$ and $0 \leq y \leq e$ or $y^{j-2m} \in \Omega^0$ and $y \in \Omega^0$. □

Lemma 6.1 indicates that in order to prove finite convergence, one needs to show that $P(\Omega^k)$ grows more slowly than $2m+k$. By Lemma 4.1, this means finding upper bounds on $\bar{r}^k$. In the following lemma, this is achieved by using a construction from Nesterov [17] which bounds

$$A^k (\bar{S}^k)^{-2} (A^k)^T$$

from below by using a certain matrix B^k which is simple enough to handle.

LEMMA 6.3. *Let* $s = c^k - (A^k)^T y$ *for any* $y \in \Omega^k$ *and* $B^0 = 8I$, $B^{k+1} = B^k + \frac{1}{m} a_{k+1} a_{k+1}^T$. *Then*

$$A^k (S^k)^{-2} (A^k)^T \succeq B^k;$$

that is,

$$A^k (S^k)^{-2} (A^k)^T - B^k$$

is positive definite.

Proof. Let $Y^k = \operatorname{diag}(y^k)$. Then

$$\begin{aligned} A^k (S^k)^{-2} (A^k)^T &= (Y^k)^{-2} + (I - Y^k)^{-2} + \sum_{j=1}^{k} \frac{a_j a_j^T}{(s_{2m+j}^k)^2} \\ &\succeq (Y^k)^{-2} + (I - Y^k)^{-2} + \frac{1}{m} \sum_{j=1}^{k} a_j a_j^T \quad \text{(by Lemma 6.2)} \\ &\succeq 8I + \frac{1}{m} \sum_{j=1}^{k} a_j a_j^T \quad \text{(as } 0 \le y^k \le e\text{)} \\ &= B^k. \qquad \square \end{aligned}$$

LEMMA 6.4. *Let* $\bar{s}^k = c^k - (A^k)^T \bar{y}^k$ *be the slacks at the analytic center* $\bar{y}^k$ *of* Ω^k *and* $(\omega^k)^2 = a_{k+1}^T (B^k)^{-1} a_{k+1}$; *then*

$$(\omega^k)^2 \ge a_{k+1}^T (A^k (\bar{S}^k)^{-2} (A^k)^T)^{-1} a_{k+1} = (\bar{r}^k)^2. \tag{6.3}$$

Proof. Using Lemma 6.3,

$$(\omega^k)^2 = a_{k+1}^T (B^k)^{-1} a_{k+1} \ge a_{k+1}^T (A^k (\bar{S}^k)^{-2} (A^k)^T)^{-1} a_{k+1} = (\bar{r}^k)^2. \tag{6.4}$$ $\square$

This implies that upper bounds on the series of $(\omega^k)^2$ will lead to upper bounds on the series $(\bar{r}^k)^2$.

The following lemma adapts a result of Nesterov [17] to the technical hypotheses of this paper.

LEMMA 6.5.

$$\sum_{j=0}^{k} (\omega^j)^2 \le 2m^2 \ln \left(1 + \frac{k+1}{8m^2}\right).$$

Proof. Note that

$$\det B^{k+1} = \det \left(B^k + \frac{1}{m} a_{k+1} a_{k+1}^T\right) = \left(1 + \frac{(\omega^k)^2}{m}\right) \det B^k.$$

Thus

$$\ln \det B^{k+1} = \ln \det B^k + \ln \left(1 + \frac{(\omega^k)^2}{m}\right).$$

But

$$\frac{(\omega^k)^2}{m} \leq \frac{1}{8} a_{k+1}^T a_{k+1} = \frac{1}{8};$$

hence

$$\begin{aligned}
\ln\left(1+\frac{(\omega^k)^2}{m}\right) &\geq \frac{(\omega^k)^2}{m} - \frac{\left(\frac{(\omega^k)^2}{m}\right)^2}{2\left(1-\frac{(\omega^k)^2}{m}\right)} \\
&= \frac{(\omega^k)^2}{m}\left(1 - \frac{\frac{(\omega^k)^2}{m}}{2\left(1-\frac{(\omega^k)^2}{m}\right)}\right) \\
&\geq \frac{(\omega^k)^2}{2m}.
\end{aligned}$$

Thus we have

$$\ln\det B^{k+1} \geq \ln\det B^0 + \sum_{j=0}^{k} \frac{(\omega^j)^2}{2m} = m\ln 8 + \sum_{j=0}^{k} \frac{(\omega^j)^2}{2m}.$$

But

$$\frac{1}{m}\ln\det B^{k+1} \leq \ln\frac{\text{trace } B^{k+1}}{m} = \ln\left(8+\frac{k+1}{m^2}\right).$$

Thus

$$\sum_{j=0}^{k} \frac{(\omega^j)^2}{2m} \leq m\ln\left(8+\frac{k+1}{m^2}\right) - m\ln 8$$

or

$$\sum_{j=0}^{k} (\omega^j)^2 \leq 2m^2 \ln\left(1+\frac{k+1}{8m^2}\right). \qquad \square$$

THEOREM 6.6. *The column generation scheme stops with a feasible solution as soon as k satisfies*

$$\frac{\epsilon^2}{m} \geq \frac{\frac{1}{2} + 2m\ln(1+\frac{k+1}{8m^2})}{2m+k+1} e^{-2\alpha\frac{k+1}{k+1+2m}}.$$

Proof. From relation (6.2) and Lemma 6.1,

$$\begin{aligned}
(2m+k+1)\ln\epsilon &\leq P(\Omega^{k+1}) \\
&\leq P(\Omega^0) + \frac{1}{2}\sum_{j=0}^{k} \ln(\bar{r}^j)^2 - (k+1)\alpha \\
&= 2m\ln\frac{1}{2} + \frac{1}{2}\sum_{j=0}^{k} \ln(\bar{r}^j)^2 - (k+1)\alpha.
\end{aligned}$$

Thus

$$\ln \epsilon + \frac{k+1}{2m+k+1}\alpha \le \frac{1}{2(2m+k+1)}\left[2m \ln \frac{1}{4} + \sum_{j=0}^{k} \ln(\bar{r}^j)^2\right]$$
$$\le \frac{1}{2} \ln \frac{2m\frac{1}{4} + \sum_{j=0}^{k}(\bar{r}^j)^2}{2m+k+1} \quad \text{(from the concavity of ln)}$$
$$\le \frac{1}{2} \ln \frac{\frac{m}{2} + \sum_{j=0}^{k}(\omega^j)^2}{2m+k+1} \quad \text{(from Lemma 6.4)}$$
$$\le \frac{1}{2} \ln \frac{\frac{m}{2} + 2m^2 \ln(1+\frac{k+1}{8m^2})}{2m+k+1} \quad \text{(from Lemma 6.5)}$$

or

$$\frac{\epsilon^2}{m} \le \frac{\frac{1}{2} + 2m \ln(1+\frac{k+1}{8m^2})}{2m+k+1} e^{-2\alpha \frac{k+1}{k+1+2m}}. \qquad \square$$

Theorem 6.6 implies that the complexity of the column generation scheme, counted by the calls to the oracle, is $O^*(\frac{m^2}{\epsilon^2})$; the notation O^* means that lower-order terms are ignored. Notice that whether or not α is positive does not alter the estimate of k significantly.

THEOREM 6.7. *The approximate analytic center algorithm, which uses the updating scheme of* §5 *and either the dual method of* §3.1 *or the primal-dual method of* §3.2, *measured by the number of Newton steps or by the number of factorizations, is, for appropriate values of* η *and* β *which depend on the exact mix of recentering and updating steps, a fully polynomial approximation algorithm.*

Proof. For the prescribed values of η and β, the number of Newton steps is at most four. One additional factorization may be required for the updating procedure. $\square$

The largest value of η that guarantees $\gamma^{16} \le \eta$ (so that four dual Newton steps are enough to recenter) is about $\eta = .09$ with $\beta = .691$. In this case α may be negative, but the algorithm will still terminate after $O^*(\frac{m^2}{\epsilon^2})$ of iterations; this can be seen from the proof of Theorem 6.6.

REFERENCES

[1] D. S. ATKINSON AND P. M. VAIDYA, *A cutting plane algorithm that uses analytic centers*, Math. Programming, 69 (1995), pp. 1–43.

[2] J. V. BURKE, A. A. GOLDSTEIN, P. TSENG, AND Y. YE, *Translation cuts for minimization*, in Complexity in Numerical Optimization, P.M. Pardalos, ed., World Scientific, 1993, pp. 57–73.

[3] G. B. DANTZIG AND P. WOLFE, *The decomposition algorithm for linear programming*, Econometrica, 29 (1961), pp. 767–778.

[4] D. DEN HERTOG, C. ROOS, AND J. PH. VIAL, *A complexity reduction for the long-step path-following algorithm for linear programming*, SIAM J. Optim., 2 (1992), pp. 71–87.

[5] J. ELZINGA AND T. MOORE, *A central cutting plane algorithm for convex programming*, Math. Programming, 8 (1975), pp. 134–145.

[6] J.-L. GOFFIN, Z. LUO, AND Y. YE, *On the complexity of a column generation algorithm for convex or quasiconvex feasibility problems*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1993, pp. 187–196.

[7] J.-L. GOFFIN, A. HAURIE, AND J.-P. VIAL, *Decomposition and nondifferentiable optimization with the projective algorithm*, Management Sci., 38 (1992), pp. 284–302.

[8] C. C. GONZAGA, *Path-following methods for linear programming*, SIAM Rev., 34 (1992), pp. 167–224.

[9] J. E. KELLEY, *The cutting plane method for solving convex programs*, J. SIAM, 8 (1960), pp. 703–712.

[10] L. G. KHACIAN, *Polynomial algorithms in linear programming*, Zh. Vychisl. Mat. i Mat. Fiz., 20 (1980), pp. 51–68; translated in U.S.S.R. Comput. Math. and Math. Phys., 20 (1980), pp. 53–72.

[11] L. G. KHACIAN AND M. J. TODD, *On the complexity of approximating the maximal inscribed ellipsoid for a polytope*, Math. Programming, 61 (1993), pp. 137–160.

[12] A. LEVIN, *An algorithm of minimization of convex functions*, Soviet Math. Dokl., 160 (1965), pp. 1244–1247.

[13] J. E. MITCHELL, *Karmarkar's Algorithm and Combinatorial Optimization Problems,* Ph.D. thesis, Department of ORIE, Cornell University, Ithaca, NY, 1988.

[14] J. E. MITCHELL AND M. J. TODD, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Math. Programming, 56 (1992), pp. 245–284.

[15] S. MIZUNO, *A new polynomial time method for a linear complementarity problem*, Math. Programming, 56 (1992), pp. 31–43.

[16] A. NEMIROVSKY AND D. YUDIN, *Problem complexity and method efficiency*, in Optimization, Wiley-Interscience, New York, 1983.

[17] Y. NESTEROV, *Cutting plane algorithms from analytic centers: Efficiency estimates*, Math. Programming, 69 (1995), pp. 149–176.

[18] N. Z. SHOR, *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag, Berlin, 1985.

[19] G. SONNEVEND, *New algorithms in convex programming based on a notion of "centre" (for systems of analytic inequalities) and on rational extrapolation*, in Trends in Mathematical Optimization: Proceedings of the 4th French–German Conference on Optimization in Irsee, West–Germany, April 1986, International Series of Numerical Mathematics, Vol. 84, K. H. Hoffmann, J. B. Hiriat-Urruty, C. Lemarechal, and J. Zowe, eds., Birkhäuser-Verlag, Basel, Switzerland, 1988, pp. 311–327.

[20] S. TARASOV, L. G. KHACHIYAN, AND I. ERLIKH, *The method of inscribed ellipsoids*, Soviet Math. Dokl., 37 (1988), pp. 226–230.

[21] P. VAIDYA, *A new algorithm for minimizing convex functions over convex sets*, Math. Programming, to appear.

[22] Y. YE, *A potential reduction algorithm allowing column generation*, SIAM J. Optim., 2 (1992), pp. 7–20.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 653–673, August 1996
© 1996 Society for Industrial and Applied Mathematics
008

TENSOR METHODS FOR EQUALITY CONSTRAINED OPTIMIZATION*

DAN FENG† AND ROBERT B. SCHNABEL‡

Abstract. This paper introduces tensor methods for nonlinear equality constrained optimization problems. These are general purpose methods intended especially for problems where the constraint gradient matrix at the solution is rank deficient or ill conditioned. The new methods are adapted from the standard successive quadratic programming method by augmenting the linear model of the constraints with a simple second-order term. The second-order term is selected so that the model of the constraints interpolates constraint function values from one or more previous iterations, as well as the current constraint function value and gradients. Similar to tensor methods for nonlinear equations, the tensor methods for constrained optimization require no more function and derivative evaluations, and hardly more storage or arithmetic per iteration, than the standard SQP methods. Test results indicate that the tensor methods are more efficient than SQP methods on singular and nonsingular nonlinear equality constrained optimization problems, with a particularly large advantage on singular problems.

Key words. constrained optimization, tensor methods, singular problems, successive quadratic programming

AMS subject classifications. 65K05, 90C30

1. Introduction.

This paper describes a new class of methods for solving the nonlinear equality constrained optimization problem

$$\begin{aligned} &\min_{x \in \Re^n} f(x) \\ &\text{subject to } c(x) = 0, \end{aligned} \tag{1}$$

where $f : \Re^n \to \Re$ is a smooth function and $c : \Re^n \to \Re^m$ is a smooth nonlinear function. Problems of this type, i.e., with only equality constraints, form a very important class of constrained optimization problems for two reasons. First, they arise directly in a variety of practical applications ranging from aircraft design [16] and weather forecasting [19] to orthogonal distance regression [1]. Second, for any nonlinearly constrained optimization problem that may also have nonlinear inequality constraints, close to the solution only a fixed set of inequality constraints is active and consequently the problem of interest becomes (1).

Various methods have been developed for solving (1). Among them, methods based on successive quadratic programming (SQP) have been particularly successful. SQP is an iterative method. At each iteration, a quadratic approximation to the Lagrangian function of (1),

$$l(x, \lambda) = f(x) + c(x)^T \lambda, \tag{2}$$

* Received by the editors June 30, 1994; accepted for publication (in revised form) February 21, 1995.

† Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035-1000 (feng@riacs.edu). The research of this author was supported in part by NASA contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA).

‡ Department of Computer Science, Campus Box 430, University of Colorado, Boulder, CO 80309-0430 (bobby@cs.colorado.edu). The research of this author was supported by Air Force Office of Scientific Research grants AFOSR-90-0109 and F49620-94-1-0101, Army Research Office grants DAALO3-91-G-0151 and DAAH04-94-0228, and National Science Foundation grant CCR-9101795.

is minimized subject to a linearization of the constraints. This gives the subproblem

$$\begin{aligned} &\min_{d\in\Re^n} g(x^k)^T d + \frac{1}{2} d^T H_k d \\ &\text{subject to } c(x^k) + A(x^k)^T d = 0, \end{aligned} \tag{3}$$

where $g(x) \in \Re^n$ is the gradient of $f(x)$, $A(x) = [\nabla c_1(x), \ldots, \nabla c_m(x)]$ is the $n \times m$ matrix of constraint gradients, and $H_k \in \Re^{n\times n}$ is a matrix that equals or approximates the Hessian of the Lagrangian function. The new iterate is given by $x^{k+1} = x^k + \alpha_k d_k$, where the step length α_k is chosen to force progress toward the solution to (1). The progress is usually judged by a "merit function."

In practice, SQP methods perform reasonably well for most problems with n of moderate size. Theoretically, for f and c smooth enough, if started close enough to the solution, the sequence produced by the SQP method converges locally and q-quadratically to the solution (a point that satisfies the Kuhn–Tucker conditions), provided that the reduced Hessian (the projection of the Hessian of the Lagrangian into the null space of the constraint gradient matrix) is positive definite at the solution and the constraint gradient matrix has full rank at the solution.

Standard SQP methods are not as promising, however, for solving degenerate problems where either $A(x)$ is rank deficient at the solution or the reduced Hessian of the Lagrangian is singular (positive semidefinite) at the solution. Degenerate constrained optimization problems occur in many applications, such as structural optimization problems with degenerate feasible regions [17], some fluid flow problems involving shocks, and the orthogonal distance regression problems from [18] that are discussed in §5. In practice, SQP methods are not usually quickly locally convergent for degenerate problems. One often observes linear convergence of SQP methods when solving degenerate problems, if they converge at all. One difficulty is that when $A(x)$ is rank deficient at the solution, the Lagrange multipliers in the standard SQP method can easily blow up [23] and as a consequence, the penalty parameter in the merit function will not be bounded above, which makes it much harder for an algorithm to achieve global convergence. There is also a lack of a thorough study of the practical performance of the SQP methods on degenerate problems or of research on general purpose constrained optimization methods that improves the local convergence of SQP methods on degenerate problems. This paper will consider some of these issues.

Theoretically, the behavior of the standard SQP methods on degenerate problems is not well understood. When a problem is degenerate (such as rank deficiency in the constraint gradient matrix at the solution), very often the normal Kuhn–Tucker conditions are not satisfied. Work on characterizing the optimality conditions for such degenerate constrained minimizers is scant. These issues are considered in [8] and in a forthcoming paper [3].

The tensor methods introduced in this paper are general purpose methods that are primarily intended to improve standard SQP methods in cases when the constraint gradient matrix is singular or very ill conditioned at the solution. They are not intended to improve the standard methods when the reduced Hessian of the Lagrangian function is singular or ill conditioned at the solution. In addition, the new methods will have similar efficiency in arithmetic cost per iteration as standard methods. They are related in a number of ways to the tensor methods for nonlinear equations introduced in [25].

The new methods are based upon expanding the linearization of the constraints

in the SQP model to a quadratic approximation and forming the subproblem

$$\begin{aligned} &\min_{d\in\Re^n} g(x^k)^T d + \frac{1}{2} d^T \hat{H}_k d \\ &\text{subject to } c(x^k) + A(x^k)^T d + \frac{1}{2} T_k dd = 0, \end{aligned} \tag{4}$$

where $\hat{H}_k$ is slightly different from H_k in the SQP model for reasons that are discussed in §3. For the same reason as for the tensor method for nonlinear equations, the natural choice of $\nabla^2 c(x^k)$ for T_k is entirely impractical since it is far too expensive to form and store and would make the model problem far too expensive to solve. Instead, our choice of T_k will cause the second-order term $T_k dd$ in (4) to have a simple useful form. T_k will not contain actual second derivative information. Instead, the second-order term in (4) will allow the model to interpolate one or more additional values of the constraint function $c(x)$ that have been computed previously in a way that still allows the model (4) to be formed and solved efficiently.

In this paper, we introduce two tensor methods for equality constrained optimization: the direct tensor method and the projected tensor method. They differ both in how they form the tensor term T_k and in how they solve the augmented SQP model (4). However, they are closely related. In the direct tensor method, the tensor model of the constraints is formed similarly to the tensor model for nonlinear equations. A disadvantage of this approach is that the separation of variables into the range space and the null space of A_k, which is often used in solving the standard SQP model, no longer yields the same advantage. The projected tensor method is devised to overcome this disadvantage of the direct tensor method for solving the augmented SQP model (4). It achieves this by imposing an extra condition on T_k—that it should contain information only along the range space of A_k. This restriction is helpful computationally and is justified theoretically by the local convergence analysis of tensor methods for nonlinear equations in [9]. The reason for including both approaches in this paper is that the first is the natural one based on previous work on tensor methods for nonlinear equations, while the second is simpler and less expensive but less obvious, and so far both appear to be about equally useful computationally. Thus both appear to warrant further consideration.

This paper derives these new tensor methods, discusses their incorporation into a complete equality constrained optimization algorithm, and summarizes computational comparisons between the new methods and an analogous SQP method. The comparisons are based on an implementation in which the same global algorithm, a simple line search method, is used for all the methods. The test problems include both small and large and nonsingular and singular problems. The computational results indicate that both the direct and projected tensor methods are more efficient than the standard method on both nonsingular and singular problems, with a particularly large advantage on singular problems. We also show that the tensor methods require no more function and gradient information per iteration, and hardly more storage or arithmetic per iteration, than a standard SQP method.

The remainder of this paper is organized as follows. Since the new tensor methods for constrained optimization are closely related to the tensor methods for nonlinear equations, §2 provides a very brief review of these methods. Tensor methods for constrained optimization are discussed in §§3–6. The details of the formation and solution of the tensor model are given in §3. Then in §4 an overall tensor algorithm is introduced, along with the discussion of the merit function and the descent properties of the tensor step. Comparative test results for our implementation of two versions

of tensor model based methods versus an analogous method based on the standard SQP model are summarized in §5. Finally, a summary and a discussion of areas for future related research are given in §6.

Throughout this paper, for any vector or matrix object B, $\|B\|$ is assumed to be the l_2 norm of B unless another norm is explicitly indicated. Subscripts are used to denote components of vectors, matrices, and three-dimensional tensors. Superscripts denote iteration indices. We often abbreviate $c(x^k)$, $g(x^k)$, and $A(x^k)$ by c_k, g_k, and A_k, respectively. For a vector $v = [v_1, \ldots, v_n]^T \in \Re^n$, we define $v^2 \equiv [v_1^2, \ldots, v_n^2]^T \in \Re^n$.

2. Tensor methods for nonlinear equations. The tensor model for nonlinear equations introduced by Schnabel and Frank [25] is a quadratic model of $F(x)$ formed by adding a second-order term to the linear Taylor series model, giving

$$M_T(x^k + d) = F(x^k) + F'(x^k)d + \frac{1}{2}T_k dd, \tag{5}$$

where $T_k \in \Re^{n \times n \times n}$ is intended to supply second-order information about $F(x)$ around x^k, without appreciably increasing the cost of forming, storing, or solving the model. Schnabel and Frank [25] form T_k by requiring the tensor model to interpolate the function values at a very small number, p, of past iterates. This requires no additional function or derivative evaluations. By choosing the smallest T_k, in the Frobenuis norm, that meets these conditions, T_k has rank p, the term $T_k dd$ has a simple form, and the cost of forming and storing T_k is very small.

Since (5) may or may not have a root, the goal is to solve

$$\min_{d \in \Re^n} \|M_T(x^k + d)\|_2.$$

Schnabel and Frank [25] show that this can be done nearly as efficiently as solving the standard linear model, usually by solving p quadratic equations in p unknowns and $n - p$ linear equations in $n - p$ unknowns. If $F'(x^k)$ is large and sparse, the tensor model solution still costs very little more than the standard sparse Newton iteration; see [2].

Computational results in [25] and [2] show that the tensor method is more efficient than an analogous standard method based upon Newton's method on both nonsingular and singular problems, with a particularly large advantage on singular problems. In tests in [25] on a standard set of nonsingular test problems, the tensor method is almost always more efficient than the standard method and is never significantly less efficient, with average improvements over 20% for all problems and over 35% for harder problems. The average improvements in iterations and function evaluations for singular problems is large, generally in the range of 30% to 65%. More recent computational experiments in [2], including experiments on much larger problems, show similar advantages for tensor methods.

It is shown in [9] that under mild conditions the sequence of iterates generated by the tensor method based upon an ideal tensor model converges locally and two-step Q-superlinearly to the solution with Q-order $\frac{3}{2}$, and that the sequence of iterates generated by the tensor method based upon a practical tensor model converges locally and three-step Q-superlinearly to the solution with Q-order $\frac{3}{2}$. In the same situation, standard methods converge linearly with constant converging to $\frac{1}{2}$. Hence, tensor methods have theoretical advantages over standard methods on one class of singular problems. The analysis in [9] also confirms that tensor methods converge at least quadratically on problems where the Jacobian matrix at the root is nonsingular.

3. Tensor model formation and solution for equality constrained optimization. This section covers the derivation, formation, and solution of the tensor model for equality constrained optimization. An immediate option one could try in forming the tensor model for constrained optimization is to directly apply the tensor model for a system of equations to the first-order necessary conditions for the constrained minimizer, i.e.,

$$g(x) + A(x)\lambda = 0, \tag{6}$$

$$c(x) = 0. \tag{7}$$

However, this approach is intrinsically flawed. To explain why, define $F(x, \lambda) : \Re^{n+m} \rightarrow \Re^{n+m}$ as

$$F(x, \lambda) = \begin{pmatrix} g(x) + A(x)\lambda \\ c(x) \end{pmatrix}. \tag{8}$$

Then

$$\nabla F(x, \lambda) = \begin{pmatrix} \nabla^2 f(x) + \lambda \nabla^2 c(x) & A(x) \\ A(x)^T & 0 \end{pmatrix}. \tag{9}$$

Suppose $A(x)$ has rank deficiency one at the solution x^*. Then $\nabla F(x, \lambda)$ has rank deficiency at least one at the solution. Let u be the right null vector of $A(x^*)$; then $\bar{u} = [0, u^T]^T$ is a right null vector of $\nabla F(x^*, \lambda^*)$. Simple calculation shows that $\nabla^2 F(x^*, \lambda^*)\bar{u}\bar{u} \equiv 0$. Since this violates the conditions in the assumptions that guarantee the fast convergence of the tensor methods [9], we would not expect tensor methods for nonlinear equations applied to (6)–(7) to be helpful in overcoming the slow convergence of the normal SQP method in cases when $A(x^*)$ is rank deficient. This difficulty comes from the fact that since $F(x, \lambda)$ is linear in λ, the bottom right hand blocks of all the derivatives of F, including $\nabla^2 F(x^*, \lambda^*)$, are zero.

Another simple proposal that is closely related to the approach we will take is to augment the SQP model by adding a second-order tensor term to the linear model of the constraints, while keeping the quadratic model of the Lagrangian function in the SQP model unchanged. As we will see in §3.1, it turns out that this does not quite give the correct tensor model either. The correct tensor model requires a slight but important modification to the Hessian of the Lagrangian that is used in the objective function in the SQP model in conjunction with the addition of a second-order term to the model of the constraints. Section 3.1 gives a derivation of the tensor model and explains these issues. In §3.2, two possible quadratic models of the constraints, the direct tensor model and the projected tensor model, are introduced, and the cost of forming each tensor model is discussed. In §3.3, the solution of the augmented SQP models formed by adding each of these tensor terms to the constraints of the standard SQP model is discussed. In addition, we discuss how to obtain a standard SQP step as a by-product of the tensor step solution procedure. Finally, a comparison of the total costs per iteration of the standard SQP method, the direct tensor method, and the projected tensor method is given and local convergence results are very briefly summarized.

3.1. Derivation of the tensor model. The Kuhn–Tucker conditions are widely used in the design and analysis of algorithms for constrained optimization. However, (6) is often not satisfied by constrained optimization problems where the constraint gradient matrix has rank deficiency at the solution. Byrd, Feng, and Schnabel [3]

introduce and discuss several optimality conditions for singular constrained optimization. These conditions are a generalization of the traditional Kuhn–Tucker conditions for constrained optimization. The theory in [3] indicates that the Kuhn–Tucker conditions can still be satisfied in a "limit" sense for many singular problems. An informal reason is that for many singular problems, for x^k arbitrarily near x^*, we still have $g_k + A_k\lambda^k = O(\|x^k - x^*\|)$ even though $\lim_{k\to\infty} \|\lambda^k\| = \infty$. This is true in part because although $\lim_{k\to\infty} \|\lambda^k\| = \infty$, $A_k\lambda^k = -A_k(A_k^T A_k)^{-1}A_k^T g_k$ has a well-defined finite value everywhere away from the singular manifold of $A(x)$, since $A_k(A_k^T A_k)^{-1}A_k^T$ is a projection matrix. Thus a model derived from (6)–(7) is still a worthwhile basis for a constrained optimization algorithm arbitrarily close to the solution x^*. An interesting class of singular problems covered by the theory in [3] consists of problems where the subspace spanned by the singular vectors corresponding to the vanishing singular values is continuous. A simple instance is problems that result from transforming a nondegenerate problem by squaring some of its constraints. In these cases, the optimality condition becomes $g_* + \hat{A}_*\hat{\lambda}^* = 0$, where $\hat{A}_*$ and $\hat{\lambda}^*$ are the limits of $\{A_k\}$ and $\{\lambda^k\}$ for a sequence $\{x_k\}$ that only intersects the singular manifold at x^*.

This analysis indicates that whether the problems are singular or nonsingular, it is desirable for a constrained optimization algorithm to approximately solve the system of nonlinear equations (6)–(7) at each iteration. From this point of view, the standard SQP method can be viewed as being based on solving a linear model of this nonlinear system, since the first-order conditions for the standard SQP model at the current point are the Newton equations for (6)–(7). One of our interests in developing tensor methods is to improve SQP methods when the constraint gradient matrix is singular at the solution. In this case, the main weakness in the linear SQP model of (6)–(7) comes from the linear model of the constraints (7). Hence, to deal with problems where $A(x^*)$ is singular, it is reasonable to use a quadratic (tensor) model of (7) while keeping a linear model of (6).

From Taylor series expansion, at a current point x^k, the linear model of (6) is given by

$$g_k + A_k\lambda + H_k d = 0, \tag{10}$$

where $H_k = \nabla^2 f(x^k) + \lambda^k \nabla^2 c(x^k)$. If a tensor model is used for (7), the model for (6)–(7) that results is

$$g_k + A_k\lambda + H_k d = 0, \tag{11}$$

$$c_k + A_k^T d + \frac{1}{2}T_k dd = 0, \tag{12}$$

where T_k is the tensor term at the current iteration which is intended to carry some second-order information about the constraints. Equations (11)–(12) will be the basis of our approach. The formation of the tensor term will be discussed in §3.2.

Now we discuss how (11)–(12) corresponds to a constrained optimization model problem. Let $\bar{\lambda}$, $\bar{d}$ be a solution to (11)–(12). Then it is easy to see that $\bar{\lambda}$, $\bar{d}$ satisfies the Kuhn–Tucker conditions for

$$\begin{aligned} &\min_{d\in\Re^n} g_k^T d + \frac{1}{2}d^T(H_k - \bar{\lambda}T_k)d \\ &\text{subject to } c_k + A_k^T d + \frac{1}{2}T_k dd = 0, \end{aligned} \tag{13}$$

i.e., $(\lambda, d) = (\bar{\lambda}, \bar{d})$ satisfies

$$g_k + (H_k - \bar{\lambda}T_k)d + A_k\lambda + \lambda T_k d = 0, \tag{14}$$

$$c_k + A_k^T d + \frac{1}{2} T_k dd = 0. \tag{15}$$

Hence if λ^k is the current estimate of the Lagrange multiplier, by choosing

$$\hat{H}_k = \nabla^2 f(x^k) + \lambda^k \nabla^2 c_k - \bar{\lambda}^k T_k \tag{16}$$

we see that the tensor model problem that corresponds to (11)–(12) is

$$\begin{aligned} &\min_{d \in \Re^n} g_k^T d + \frac{1}{2} d^T \hat{H}_k d \\ &\text{subject to } c_k + A_k^T d + \frac{1}{2} T_k dd = 0. \end{aligned} \tag{17}$$

This derivation indicates that using H_k from the SQP model without modification in (17) would not lead to a correct model problem for (1). Instead, the model of the Hessian of the Lagrangian needs to be modified to reflect the second-order information in the model of the constraints. Note also that equation (17) is only a conceptual model, since one cannot form (17) before $\bar{\lambda}$ is obtained from solving (17). However, we will see later that we can still utilize this model computationally, in part by using (11)–(12) directly.

3.2. Two approaches to forming the tensor term for the constraints. This subsection presents two approaches, the direct method and the projected method, to forming the second-order term T_k in the model of the constraint equation (12). In both cases, this term will cause (12) to interpolate the constraint values $c(x^{-i})$ at p not necessarily consecutive iterates $x^{-1}, \ldots, x^{-p}$. As in tensor methods for nonlinear equations, these p iterates are chosen to be strongly linearly independent, meaning that the step $x^{-i} - x^k$ makes a large angle with the subspace spanned by previous selected steps. This usually results in p being 1 or 2. The two methods differ in how they select T_k to satisfy the p interpolation conditions. Both will be seen to be effective computationally.

Direct tensor model. In the direct tensor model, the tensor term in the model of the constraint equations is formed in exactly the same manner as in the tensor model of nonlinear equations. For $i = 1, \ldots, p$, define $s_i = x^{-i} - x^k$; $z_i = 2(c(x^{-i}) - c(x^k) - A_k^T s_i)$. Then T_k is chosen to satisfy

$$\begin{aligned} &\min_{T_k \in \Re^{m \times n \times n}} \|T_k\|_F \\ &\text{subject to } T_k s_i s_i = z_i, \ i = 1, \ldots, p. \end{aligned} \tag{18}$$

This causes the tensor model of the constraints, $m(x^k + s) = c(x) + A_k^T s + \frac{1}{2} T_k ss$, to interpolate $c(x^{-i})$, $i = 1, \ldots, p$. Define $Z \in \Re^{m \times p}$ with column i of $Z = z_i$, $i = 1, \ldots, p$, and $M \in \Re^{p \times p}$ by $M[i,j] = (s_i^T s_j)^2, 1 \le i, j \le p$. By the same techniques as those for forming the tensor term for nonlinear equations, the solution to the previous minimization problem is

$$T_k = \sum_{i=1}^{p} b_i s_i s_i, \tag{19}$$

where b_i is the ith column of $B \in \Re^{m \times p}$, B defined by $B = ZM^{-1}$. It is easy to verify that forming the tensor term requires $pmn + O(p^2(m+n))$ multiplications and additions. The leading cost comes from calculating the p matrix-vector products

$A_k^T s_i, i = 1, \ldots, p$; the cost of forming M is $O(p^2 n)$ and the cost of solving $B = ZM^{-1}$ is $O(p^2 m)$. Note that $z_i = \nabla^2 c(x^k) s_i s_i + O(\|s_i\|^3)$. If $\nabla^2 c(x^k)$ is available, one may use $z_i = \nabla^2 c(x^k) s_i s_i$ instead. In the projected tensor model discussed next, the same statements are true if we replace s_i with $\hat{s}_i$.

Projected tensor model. An important feature in the efficient solution of the SQP model is the separation of the variable space into orthogonal subspaces spanning the range and null spaces of the model constraints and a corresponding clean separation in the solution for these variables. We will see in §3.3 that the direct tensor model does not allow as clean a separation of the solution process. The projected tensor method is a variant that allows this.

To form the tensor term in the projected tensor model, for $i = 1, \ldots, p$, let

$$s_i = x^{-i} - x^k; \; P_k = A_k (A_k^T A_k)^{-1} A_k^T; \; \hat{s}_i = P_k s_i; \tag{20}$$

$$z_i = 2(c(x^{-i}) - c(x^k) - A_k^T \hat{s}_i) = 2(c(x^{-i}) - c(x^k) - A_k^T s_i). \tag{21}$$

That is, P_k is the projection matrix onto the range space of the linearized constraints, and $\hat{s}_i$ is the projection of s_i onto this subspace. Then T_k is chosen to satisfy

$$\begin{aligned} &\min_{T_k \in \Re^{m \times n \times n}} \|T_k\|_F \\ &\text{subject to } T_k \hat{s}_i \hat{s}_i = z_i, \; i = 1, \ldots, p, \text{ and } T_k P_k P_k = T_k, \end{aligned} \tag{22}$$

meaning that T_k is restricted to lie in the range space of the linearized constraints, and the interpolation conditions are posed in this subspace. However, since $P_k s_i = \hat{s}_i$, $T_k \hat{s}_i \hat{s}_i = z_i$, and $T_k P_k P_k = T_k$, (22) implies that $z_i = T_k \hat{s}_i \hat{s}_i = T_k (P_k s_i)(P_k s_i) = (T_k P_k P_k) s_i s_i = T_k s_i s_i$, $i = 1, \ldots, p$, so that the same interpolation conditions $m(x^k + s_i) = c(x^{-i})$, $i = 1, \ldots, p$ as those for the direct tensor model also are satisfied. Also, the analysis in [8, 9] indicates that the restriction of T_k that is made in the projected tensor method is consistent with the desired convergence properties on singular problems. This analysis indicates that only information incorporated into T_k from the space corresponding to the rank deficiency in $A(x^*)$ (i.e., the space spanned by the columns of $A(x)$ near the solution x^* but not at x^*) can help overcome the slow convergence caused by the rank deficiency of $A(x^*)$. This information is contained in the range space of $A(x)$ for x close to, but not at, x^*.

To solve (22), define Z as above and $\hat{M} \in \Re^{p \times p}$ by $\hat{M}[i,j] = (\hat{s}_i^T \hat{s}_j)^2, 1 \le i, j \le p$. By the techniques for forming the tensor term for the model of nonlinear equations, we can solve the relaxed form of (22),

$$\begin{aligned} &\min_{T_k \in \Re^{m \times n \times n}} \|T_k\|_F \\ &\text{subject to } T_k \hat{s}_i \hat{s}_i = z_i, \; i = 1, \ldots, p, \end{aligned}$$

to give

$$T_k = \sum_{i=1}^{p} \hat{b}_i \hat{s}_i \hat{s}_i, \tag{23}$$

where $\hat{b}_i$ is the ith column of $\hat{B} \in \Re^{m \times p}$, $\hat{B}$ defined by $\hat{B} = Z\hat{M}^{-1}$. Since T_k given by (23) also satisfies $T_k P_k P_k = T_k$ because $P_k \hat{s}_i = \hat{s}_i$, (23) is the solution to (22).

Given the QR factorization of A_k, $A_k = QR$,

$$\hat{s}_i^T \hat{s}_j = s_i^T P_k P_k s_j = s_i^T P_k s_j = s_i^T A_k (A_k^T A_k)^{-1} A_k^T s_j = s_i^T Y_k Y_k^T s_j, \tag{24}$$

where Y_k is the first m columns of Q. Calculating $Y_k^T s_i$, $i = 1, \ldots, p$, costs pmn multiplications. Afterwards, the cost of forming $\hat{M}$ and solving $\hat{B} = Z\hat{M}^{-1}$ is $O(p^2 m)$. Hence the total cost of forming the tensor term for the projected tensor method is $pmn + O(p^2 m)$, which has the same leading cost as that of the direct tensor method, provided that the QR factorization of A_k is available.

If A_k is rank deficient, then a QR decomposition of A_k with column pivoting such that $r_{11} \geq r_{22} \geq \cdots \geq r_{mm}$ must be computed. In this case, if A_k has rank deficiency q, then we can choose Y_k to be the first $m - q$ columns of Q in (24) and replace P_k in (20) with $Y_k Y_k^T$, and the remainder of the derivation and computational procedure is unchanged.

3.3. Solving the SQP model with tensor term added to the constraints. After T_k is formed, the next question is how to solve the tensor model efficiently. This subsection discusses this issue.

The standard SQP model is not well posed if the reduced Hessian matrix $Z_k^T H_k Z_k$ is not positive definite. In solving the tensor model, we would also like to have the convenience of keeping this matrix positive definite. We will assume that $Z_k^T H_k Z_k$ is positive definite throughout this section and deal with the case when it is not in §4.

We first review the solution of the standard SQP model since the solution of the tensor model builds upon some of these techniques. Then the solution of the direct tensor model is discussed. Afterwards, the solution of the projected tensor model is given. In both cases, we show how to also obtain the standard SQP step as a by-product of solving the tensor model, since this step will be used occasionally in the global portion of our tensor methods. Finally, we summarize and compare the costs per iteration of tensor methods and the SQP method.

We use the notation of §3.2 for $B_k \in \Re^{m \times p}$ and $\hat{B}_k \in \Re^{m \times p}$. In addition we define $S_k \in \Re^{n \times p}$ with column i of $S_k = s_i, i = 1, \ldots, p$ and $\hat{S}_k \in \Re^{n \times p}$ with column i of $\hat{S}_k = \hat{s}_i, i = 1, \ldots, p$.

Solving the standard SQP model. The standard SQP model is given by (3). The model can be solved by means of separation of variables. Assume that A_k has full rank and that the QR factorization of A_k, $A_k = Q_k R_k$ is given. Let $d = Y_k d_1 + Z_k d_2$, where Y_k is the first m columns of Q_k and Z_k is the last $n - m$ columns of Q_k. Clearly $Y_k^T Z_k = 0$, $Y_k^T Y_k = I$, and $Z_k^T Z_k = I$, and $A_k = Y_k \bar{R}_k$, where $\bar{R}_k$ is the triangular matrix consisting of the first m rows of R_k. Then d_1 can be solved directly from the constraint equations by $d_1 = -\bar{R}_k^{-T} c_k$, using the fact that $A_k^T Z_k = 0$ and $A_k^T Y_k = \bar{R}_k$. After d_1 is obtained, plugging d_1 into the objective function and dropping the constant terms yields an unconstrained minimization problem in terms of d_2 only, which is solved by $d_2 = -(Z_k^T H_k Z_k)^{-1} Z_k^T (g_k + H_k Y_k d_1)$ when $Z_k^T H_k Z_k$ is positive definite.

When A_k is rank deficient with $\text{rank}(A_k) = m - q$, in order in carry out the previous procedure, we choose Y_k as the first $m - q$ columns of Q_k and Z_k as the last $n - m + q$ columns of Q_k. The constraint equation is solved in the least squares sense to give $d_1 = -(\bar{R}_k^T \bar{R}_k)^{-1} \bar{R}_k^T c_k$, where $\bar{R}_k$ is the first $m - q$ rows of R_k, and then d_2 is obtained as before.

Solving the direct tensor model. We solve the direct tensor model by solving (11) and (12). Similar to the scheme for solving the SQP model discussed previously, assume first that A_k has full rank, and let $d = Y_k d_1 + Z_k d_2$, where Y_k, Z_k, and $\bar{R}_k$ are previously defined. Premultiplying (11) by Z_k^T, while leaving (12) unchanged, eliminates λ and results in a system of n quadratic equations in terms of the n unknowns

d only, that is,

$$Z_k^T H_k d = -Z_k^T g_k,$$

$$A_k^T d + \frac{1}{2} B_k \beta^2 = -c_k,$$

where $T_k = B_k S_k S_k$ (i.e., (19)) and $\beta = S_k^T d$. We solve this system by solving for β first. Using $d = Y_k d_1 + Z_k d_2$ and $Y_k^T Z_k = 0$, we have

$$Z_k^T H_k Z_k d_2 + Z_k^T H_k Y_k d_1 = -Z_k^T g_k, \tag{25}$$

$$\bar{R}_k^T d_1 + \frac{1}{2} B_k \beta^2 = -c_k. \tag{26}$$

From the second equation

$$d_1 = -\bar{R}_k^{-T} \left(c_k + \frac{1}{2} B_k \beta^2 \right). \tag{27}$$

Plugging this into the first equation gives

$$d_2 = -(Z_k^T H_k Z_k)^{-1} Z_k^T \left(g_k - H_k Y_k \bar{R}_k^{-T} \left(c_k + \frac{1}{2} B_k \beta^2 \right) \right). \tag{28}$$

Premultiplying (27) by $S_k^T Y_k$, (28) by $S_k^T Z_k$, and adding up the results yields

$$\begin{aligned} & S_k^T Y_k d_1 + S_k^T Z_k d_2 = S_k^T d = \beta \\ &= -S_k^T Y_k \bar{R}_k^{-T} \left(c_k + \frac{1}{2} B_k \beta^2 \right) \\ &\quad -S_k^T Z_k (Z_k^T H_k Z_k)^{-1} Z_k^T \left(g_k - H_k Y_k \bar{R}_k^{-T} \left(c_k + \frac{1}{2} B_k \beta^2 \right) \right) \\ &= (-S_k^T + S_k^T Z_k (Z_k^T H_k Z_k)^{-1} Z_k H_k) Y_k \bar{R}_k^{-T} \left(c_k + \frac{1}{2} B_k \beta^2 \right) \\ &\quad -S_k^T Z_k (Z_k^T H_k Z_k)^{-1} Z_k^T g_k. \end{aligned} \tag{29}$$

This is a system of p quadratic equations in the p unknowns β. Given the QR factorization of A_k and the Cholesky factorization of $Z_k^T H_k Z_k$, the formation of (29) involves a series of backsolves of $(n-m) \times (n-m)$ or $m \times m$ triangular systems and matrix-vector multiplications. The total cost of forming (29) is $O(pn^2)$ multiplications. Once it is formed, this nonlinear system can be solved either exactly or in the least squares sense, with $O(p^4) \leq O(m^2)$ multiplications, assuming that the number of iterations is $O(p)$ since the cost per iteration is $O(p^3)$. After β is obtained, d_1 can be solved from (27) and d_2 can be solved from (28).

Compared to the solution of the standard SQP model, the total extra expense of solving the direct tensor model is at most $O(pn^2)$, given the QR factorization of A_k and the Cholesky factorization of $Z_k^T H_k Z_k$, which is not significant compared to $O(n^3)$, the total cost of solving the standard SQP model in the same fashion.

It is also easily seen that the standard SQP step can be obtained from the previous procedure by letting $\beta = 0$ in (27) and (28). The additional cost of this once the tensor step has been computed as described previously is one backsolve of an $m \times m$ triangular system and two backsolves of $(n-m) \times (n-m)$ triangular systems, not more expensive than $O((n-m)^2 + m^2)$ operations.

When A_k is rank deficient with $\text{rank}(A_k) = m - q$, Y_k and Z_k can be chosen in the same way as in the similar situation for the SQP method, i.e., $Y_k \in \Re^{n\times(m-q)}$ is an orthogonal basis for the column space of A_k and Z_k is an orthogonal basis for the complement of this space. The new relation between d_1 and β becomes $d_1 = -(\bar{R}_k^T \bar{R}_k)^{-1} \bar{R}_k^T (c_k + \frac{1}{2} B_k \beta^2)$, where $A_k = Y_k \bar{R}_k$. This is the solution of the constraint equation in the least squares sense. With this replacing (27), the remainder of the derivation is unchanged.

Solving the projected tensor model. The projected tensor model is solved by applying a separation of variables to (17). Assume first that A_k has full rank, and let $d = Y_k d_1 + Z_k d_2$, with Y_k, Z_k, and $\bar{R}_k$ as defined previously. Then, plugging the projected tensor term $T_k = \hat{B}_k \hat{S}_k \hat{S}_k$ given by (23) into the tensor model of the constraints gives

$$\begin{aligned} & c_k + A_k^T Y_k d_1 + \frac{1}{2} \hat{B}_k \{\hat{S}_k^T d\}^2 \\ &= c_k + \bar{R}_k^T d_1 + \frac{1}{2} \hat{B}_k \{S_k^T P_k (Y_k d_1 + Z_k d_2)\}^2 \\ &= c_k + \bar{R}_k^T d_1 + \frac{1}{2} \hat{B}_k \hat{\beta}^2 = 0, \end{aligned} \tag{30}$$

where $\hat{\beta} = S_k^T Y_k d_1$, because $\hat{S}_k = P_k S_k$, $P_k Y_k = Y_k$, and $P_k Z_k = 0$. That is, the model of the constraints is a system of m quadratic equations in the m unknowns d_1. This is the sense in which solving the projected tensor model is easier and cleaner than the direct tensor model and more like SQP; the constraint equations are m (quadratic) equations in the m variables d_1 only.

Now, as in the tensor method for nonlinear equations, the unknowns d_1 can be expressed in terms of $\hat{\beta}$ with

$$d_1 = -\bar{R}_k^{-T} \left(c_k + \frac{1}{2} \hat{B}_k \hat{\beta}^2 \right). \tag{31}$$

Premultiplying this equation by $S_k^T A_k$ gives

$$S_k^T Y_k d_1 = \hat{\beta} = -S_k^T Y_k R^{-T} \left(c_k + \frac{1}{2} \hat{B}_k \hat{\beta}^2 \right), \tag{32}$$

which is a system of p quadratic equations in the p unknowns $\hat{\beta}$ only. Given the QR factorization of A_k, the formation of (32) involves $p + 1$ backsolves of an $m \times m$ triangular system and several matrix-vector multiplications. The total cost of forming (32) is $pmn + \frac{1}{2}pm^2 + p(p+1)m = O(pmn)$ operations. As in the direct tensor method, the nonlinear system can then be solved, either exactly or in the least squares sense, in $O(p^4) \leq O(m^2)$ operations, assuming that no more than $O(p)$ iterations are needed. After $\hat{\beta}$ is obtained, d_1 can be solved from (31) in $O(m^2)$ operations.

After d_1 is available, plugging it into the objective of (17) and dropping the constant terms yields an unconstrained minimization problem in d_2 only, i.e.,

$$\min_{d_2 \in \Re^{(n-m)}} \frac{1}{2} d_2^T Z_k^T \hat{H}_k Z_k d_2 + (g_k + \hat{H}_k Y_k d_1)^T Z_k d_2. \tag{33}$$

Recall that the tensor term for the projected tensor model is only in the range space of A_k. Hence $\hat{H}_k Z_k = H_k Z_k$ due to $T_k Z_k = 0$ so that (33) is equivalent to

$$\min_{d_2 \in \Re^{(n-m)}} \frac{1}{2} d_2^T Z_k^T H_k Z_k d_2 + (g_k + H_k Y_k d_1)^T Z_k d_2, \tag{34}$$

which is solved at

$$d_2 = -(Z_k^T H_k Z_k)^{-1} Z_k^T (g_k + H_k Y_k d_1) \tag{35}$$

when $Z_k^T H_k Z_k$ is positive definite.

Compared to the solution of the standard SQP model discussed previously, the extra work required in solving the projected tensor model is also minimal. The leading cost comes from the calculation of $S_k^T Y_k R^{-T} B_k$ ($pmn + \frac{1}{2}pm^2 + p^2 m = O(pmn)$ multiplications as analyzed previously) and the solution of (32) ($O(m^2)$ operations), which is not significant compared to $O(n^3)$, the total cost of solving the standard SQP model.

Similar to the situation in the direct tensor model solution, the standard SQP step can be obtained as a by-product of the previous process by letting $\hat{\beta} = 0$ in (31) and using (35). This costs $O((n-m)^2 + m^2)$ additional operations, similar to the situation in the direct tensor step calculation.

The situation when A_k is rank deficient can be treated in a similar way as in the standard SQP method and the direct tensor method. Y_k, Z_k, and $\bar{R}_k$ can be chosen in the same way as in a similar situation for the SQP method, and (31) is replaced by

$$d_1 = -(\bar{R}_k^T \bar{R}_k)^{-1} \bar{R}_k^T \left(c_k + \frac{1}{2} \hat{B}_k \hat{\beta}^2 \right). \tag{36}$$

Again, this is the solution of the constraint equation in the least squares sense. The rest of the solution procedure is unchanged.

Summary of work comparison between tensor and SQP methods. Using the solution procedures discussed previously, each of the two tensor methods and the SQP method require a QR factorization of A_k which costs $2(mn(n-m)+m^3/3)$ multiplications, a Cholesky factorization of $Z_k^T H_k Z_k$ which costs $(n-m)^3/6$ multiplications, and the formation of $Z_k^T H_k Z_k$ which costs $n^2(n-m) + (n-m)^2 n$ multiplications. Thus the cost is $O(n^3)$ operations. Tensor methods require some extra work to form and solve the tensor model, but it is not significant in comparison to this $O(n^3)$ cost. The leading term in the extra work to solve either the direct or the projected tensor model is pmn multiplications, while the leading terms in the extra work to solve the direct and projected tensor models are $4pn^2$ and pmn extra operations, respectively. (The major additional work for the direct tensor method comes from the calculation of $S_k^T Z_k (Z_k^T H_k Z_k)^{-1} Z_k^T H_k$, which is not needed for the projected tensor method.) Both tensor model solutions can provide a standard SQP step as a by-product. They use the same amount of work ($O((n-m)^2 + m^2)$ operations) in doing so. Finally, both tensor models require a small multiple of pn additional storage locations, which is insignificant in comparison to the $O(n^2)$ locations required to store the SQP model.

Local convergence of tensor model based methods. An interesting question is whether tensor model based methods have a local convergence advantage over the standard SQP method. This is studied by Feng and Schnabel [10]. They analyze local convergence properties of a tensor method based on a slightly idealized rank-one tensor model (close to the projected tensor model). Using the optimality conditions introduced in [3], they show that on the class of singular problems mentioned in §3.1, the sequence of iterates generated by the tensor method converges locally and two-step Q-superlinearly to the solution with Q-order $\frac{3}{2}$. They also show that the standard SQP method converges linearly with constant converging to $\frac{1}{2}$ on these problems. To the best of our knowledge, this result is the first of this kind for the SQP method. The analysis in [10] also confirms that the tensor method converges with at least the

same rate as the standard SQP method on problems where the constraint gradient matrix at the solution has full rank.

4. An overall tensor algorithm for equality constrained optimization. The previous section presented methods for forming and solving tensor models for equality constrained optimization. This section discusses the incorporation of these models into full equality constrained optimization algorithms. The merit function that monitors the progress of the iterates toward the solution is discussed, as well as the issue of descent of the directions generated by the tensor models on the merit function. Also, we know that when the reduced Hessian matrix $Z_k^T H_k Z_k$ is not positive definite, the standard SQP model does not have a feasible solution. A similar situation occurs for the tensor model. We also discuss how to deal with this situation.

We first present a general framework for a full tensor algorithm and discuss some issues related to its implementation. Then the effect of a step produced by this tensor algorithm on the l_1 merit function is analyzed. Finally, the global strategy used in the tensor algorithms is introduced. An important part of this strategy is that sometimes the standard SQP step is used instead of the tensor step.

4.1. Framework for a full tensor algorithm. This subsection describes the framework of a complete tensor model based method for the equality constrained optimization problem (1). Algorithm 4.1 is the top level description of this framework for methods based on either the direct tensor method or the projected tensor method. It involves a subalgorithm, Algorithm S, that is discussed in §4.3 and that is based upon a merit function φ that is discussed in §4.2.

ALGORITHM 4.1. An iteration of the tensor methods.
Given x^k, $c(x^k)$, $f(x^k)$

(A-1) Calculate A_k, Z_k, g_k and decide whether to stop. If not:
(A-2) Compute H_k.
(A-3) Form the tensor term T_k from up to $\sqrt{m}$ previous iterates.
(A-4) Obtain d_t, the solution to the tensor model, by solving (17) or solve (17) approximately if it has no solution.
(A-5) In some cases (see Algorithm S) compute d_s, the solution to the SQP model (3).
(A-6) Use Algorithm S (see §4.3) to obtain x^{k+1}.
(A-7) Decide whether to stop; if not, set $x^k = x^{k+1}$, and go to (A-1). □

Note that steps (A-3) and (A-4) can be performed only after the first iteration; at the first iteration, the standard SQP step is calculated instead. Note also that we can use either the direct tensor term or the projected tensor term T_k in (17).

There are several effective ways to estimate the Lagrange multipliers that are required in the Hessian of the Lagrangian H_k. When A_k is nonsingular, λ^k is usually required by convergence theory (see, e.g., [4]) to be chosen so that $\|\lambda^k - \lambda^*\| \leq \gamma\|x^k - x^*\|$ is satisfied for some constant γ. One common choice that satisfies this condition is $\lambda^k = -(A_k^T A_k)^{-1} A_k^T g_k$ and this is used in our algorithm.

When $Z_k^T H_k Z_k$ is not positive definite, neither the tensor model (17) nor the SQP model (3) may have a feasible solution. So, in our methods, when $Z_k^T H_k Z_k$ is not positive definite, it is modified into a positive-definite matrix. One way to do this is to perform a modified Cholesky factorization of $Z_k^T H_k Z_k$ as in [24, 13]. This implicitly results in replacing $Z_k^T H_k Z_k$ by $Z_k^T H_k Z_k + D$, where D is a nonnegative diagonal matrix which is zero when $Z_k^T H_k Z_k$ is safely positive definite and which is large enough so that $Z_k^T H_k Z_k + D$ is positive definite otherwise. Another possibility is to

replace $Z_k^T H_k Z_k$ with $Z_k^T H_k Z_k + \epsilon I$, where ϵ is a positive number whose magnitude is greater than the most negative eigenvalue of $Z_k^T H_k Z_k$. This modification does not affect the general solution procedure of either the direct tensor method or the projected tensor method.

Finally, to ensure that d_k is a descent direction on the merit function φ_{μ_k}, we choose d_k and update the penalty parameter μ_k according to the algorithm given in §4.3. Before describing these algorithms, we discuss the l_1 merit function and conditions that ensure the descent of a direction generated by Algorithm 4.1 for the merit function.

4.2. Descent of a direction generated by Algorithm 4.1. Merit functions are necessary for the global convergence of many constrained optimization algorithms [12]. Many merit functions have been proposed. The first merit function used in an SQP algorithm was the l_1 merit function proposed by Han [14],

$$\varphi_\mu(x) = f(x) + \mu \|c(x)\|_1. \tag{37}$$

The penalty parameter μ is adjusted in a nondecreasing fashion and the increment is usually bounded by a constant. Local convergence analysis shows that for many problems, μ is not increased after a certain point in the algorithm.

We base our global algorithm on the l_1 merit function because it is simple to analyze and inexpensive to implement. Although using this merit function may lead to the Maratos effect [5], we do not deal with the Maratos effect in our algorithms in this paper because it occurs less frequently with algorithms that use exact second derivatives.

To assure global convergence, it is essential that the step generated by Algorithm 4.1 defines a descent direction for the merit function φ and the penalty parameter μ_k used, i.e., that the directional derivative of the merit function in the direction d_k, $D\varphi_{\mu_k}(x^k; d_k)$, is negative. There are many ways to ensure that d_k generated by Algorithm 4.1 is a descent direction for φ_{μ_k}. We discuss one after first reviewing the corresponding strategy for SQP methods, which is closely related.

It is easy to show that for the l_1 merit function,

$$D\varphi_{\mu_k}(x^k; d_k) = g_k^T d_k + \mu_k \sum_{i=1}^{m} \operatorname{sign}((c_k)_i a_i^T d_k) |a_i^T d_k|. \tag{38}$$

If d_k is given by solving the standard SQP model (3), it is then easy to verify that since $(c_k)_i + a_i^T d_k = 0$, we have $(c_k)_i a_i^T d_k \leq 0$ for each $i = 1, \ldots, m$, and thus

$$\mu_k \sum_{i=1}^{m} \operatorname{sign}((c_k)_i a_i^T d_k) |a_i^T d_k| = -\mu_k \|c_k\|_1.$$

As a consequence, $D\varphi_{\mu_k}(x^k; d_k) = g_k^T d_k - \mu_k \|c_k\|_1$, which is less than zero if $g_k^T d_k < 0$. If $g_k^T d_k > 0$, a simple way to ensure that $D\varphi_{\mu_k}(x^k; d_k) \leq 0$ is to require $\mu_k \geq g_k^T d_k / \|c_k\|_1 + \rho$, where ρ is a positive quantity that we will elaborate upon in §4.3. Furthermore, it can be shown that μ_k is bounded above as a function of $\|g_k\|$, $\|(Z_k^T H_k Z_k)^{-1}\|$, $\|H_k\|$, and the reciprocal of the smallest singular value of A_k, independent of $\|c_k\|$.

Similarly, if d_k is given by solving the tensor model (17), (38) still holds. If $D\varphi_{\mu_k}(x^k; d_k) > 0$, one can adjust μ_k so that

$$\mu_k \geq -g_k^T d_k / \left(\sum_{i=1}^{m} \operatorname{sign}((c_k)_i a_i^T d_k) |a_i^T d_k| \right) + \rho, \tag{39}$$

provided that the denominator in (39) is negative. This is highly likely since for each $1 \leq i \leq m$, $(c_k)_i a_i^T d_k = -(c_k)_i (c_k + \frac{1}{2} T_k d_k d_k)_i$, which will be negative whenever $\frac{1}{2} T_k d_k d_k$ is a higher-order term compared to c_k. This in turn is likely close to the solution. The situation when the denominator is not safely negative can be handled by the overall global strategy, such as using the standard SQP step instead in this situation. This is discussed in §4.3.

4.3. Global strategies. This subsection presents the algorithm for choosing d_k and updating the penalty parameter μ_k in order to ensure that d_k is a descent direction on φ_{μ_k}. The strategy that is used involves sometimes choosing the SQP step instead of the tensor step. The heuristics that are used to do this are related to those that have proven successful in implementing tensor methods for nonlinear equations.

Two constants are used in the global strategy. The first, ρ, is a positive quantity used to increase the penalty parameter μ_k and the second, θ, is a positive constant used to compare candidate penalty parameters for the tensor and SQP steps. We use $\rho = \max\{\|\lambda^k\|, .05\}$ and $\theta = 5$. One should notice that there is a major difference between our definition of ρ and the definition that is traditionally used for SQP methods, where ρ is chosen as a small constant such as .05. Essentially, if $\|\lambda^k\|$ is large, our choice of ρ causes μ_k to be increased by a multiplicative rather than an additive factor. The reason for this difference is that from our computational experience and our theoretical analysis, $\|\lambda^k\|$ often goes to infinity for problems where the constraint gradient matrix is rank deficient at the solution. Hence, when close to the solution for these problems, a small ρ is negligible compared to $\|\lambda^k\|$ so that u_k would not necessarily be increased on the same scale as $\|\lambda^k\|$. As a consequence, the full step d_k would often be rejected by the merit function, which would result in very frequent updating of the penalty parameter μ and very slow convergence. Our choice of ρ overcomes this difficulty by keeping ρ proportional to $\|\lambda^k\|$. We also noticed from our computational experience that this choice of ρ does not affect the performance of Algorithm 4.1 on nonsingular problems. Also, this strategy of choosing ρ (increase by a multiplicative factor) would not change the theoretical results that have been established for algorithms that use the traditional strategy of choosing ρ (increase by an additive factor) for nonsingular problems. This is because a typical theory for nonsingular problems requires only that ρ be a finite positive number (not necessarily fixed). Our way of choosing ρ certainly guarantees this when λ^* is finite.

Algorithm S (Step strategy) uses the strategy that the tensor step should be taken only if it does not cause the penalty parameter to increase much more than the SQP step does.

ALGORITHM S.

(S-1) If the tensor step gives a sufficient descent on $\varphi_{\mu_{k-1}}$, then set $d_k = d_t$ and $\mu_k = \mu_{k-1}$. Go to (S-5).

(S-2) Calculate μ_k'', the penalty parameter for the SQP step, by letting $\mu_k'' = \mu_{k-1}$ if $\mu_{k-1} \geq \omega_k + \rho$; otherwise $\mu_k'' = \omega_k + 2\rho$, where $\omega_k = g_k^T d_s / \|c_k\|_1$ if $c_k \neq 0$; $\omega_k = 0$ otherwise.

(S-3) If the denominator of (39) is negative, then set $d_k = d_s$ and $\mu_k = \mu_k''$ and go to (S-5); otherwise compute μ_k' from (39).

(S-4) If $\mu_k' \leq \theta \mu_k''$ then set $d_k = d_t$, and $\mu_k = \mu_k'$; otherwise set $d_k = d_s$ and $\mu_k = \mu_k''$.

(S-5) Do a line search on the pair d_k, φ_{μ_k}. □

An interesting question is whether our tensor algorithm is globally convergent.

We have not attempted a global convergence analysis of the tensor methods for constrained optimization developed in this paper. The main difficulty of such an analysis arises from the possible singularity of the constraint gradient at the solution. This singularity would usually result in both the penalty parameter μ_k and the norm of the Lagrange multiplier λ^k going to infinity. This situation is not covered by available global convergence theory for algorithms based on the SQP model. Our speculation is that the global convergence analysis is possible for certain classes of problems in spite of this difficulty. One reason is that although the penalty parameter could go to infinity for singular problems, our experience tells us that in this situation, the iterates usually converge along the direction of the singular vector corresponding to the smallest singular value of the constraint gradient matrix at the solution. This implies that close to the solution, $\|c(x^k)\|_1$ is a second-order term. Hence the effect of $\mu_k\|c(x^k)\|_1$ together in the l_1 merit function would be similar to a bounded constant times a first-order term, which is the case for nonsingular problems. Another reason is that although $\|\lambda^k\|$ going to infinity leads to the unboundedness of the Hessian of the Lagrangian H_k (remember that $H_k = \nabla^2 f_k + \lambda^k \nabla^2 c_k$), near the solution the term $H_k Z_k$ is bounded for certain problem classes (see [10]). Hence the difficulty associated with μ_k and $\|\lambda^k\|$ going to infinity could be computational rather than theoretical.

On the other hand, for nonsingular problems, it would likely be fairly easy to adapt the available theory for SQP methods to tensor methods. This is because the tensor methods always use the SQP step as a safeguard. Hence a relevant question is whether our tensor algorithms are globally convergent for nonsingular problems if the SQP step is always chosen. It is easy to see that the only difference between this induced SQP method and the traditional SQP method is the way of choosing ρ for increasing the penalty parameter μ_k. Because of a consideration of singular problems, our ρ is chosen dependent on $\|\lambda^k\|$ in a way that is bounded for nonsingular problems. Hence, the difference between our way and the traditional way for updating the penalty parameter is whether a fixed number or a changing but bounded number is used as the increment for the penalty parameter. Using a changing but bounded number does not affect the results of most of the global convergence analyses for methods based on the SQP model. Therefore we would expect our algorithms to have the same global behavior as traditional algorithms based on the SQP model on singular problems.

5. Implementation and testing of tensor methods for equality constrained optimization. This section gives some highlights of the implementation and testing of tensor methods for equality constrained optimization. Since we are still in the early stages of implementing these methods, we purposely designed our algorithms to be as simple as possible. This strategy is used both in the formation of the tensor model and the implementation of the linear algebraic calculations, which favor simplicity and transparency over efficiency. In particular, we use tensor models that interpolate only one past iterate. (This means that the tensor term has rank one.) In previous experiments with tensor methods for nonlinear equations and unconstrained optimization, such tensor models were nearly as effective as those that interpolate more past iterates.

We describe comparative testing for three methods—the direct tensor method, the projected tensor method, and the standard SQP method—based on the same implementation. We present summary statistics of the test results in this section. For details on the implementation, test problems, complete test results, and detailed analysis and discussion of the test results, see [8] or the initial expanded version of

this paper [11].

Implementation. The projected tensor method and the direct tensor method that are used in our tests are derived from the framework of Algorithm 4.1. A standard SQP method is derived in an analogous fashion. In each method, the variables are separated into components in the range and null spaces of the current constraint gradient matrix $A(x^k)$. The bases for the range and null spaces are obtained from the singular value decomposition of $A(x^k)$. Second, for either an SQP model or a tensor model to be bounded below, the reduced Hessian matrix must be positive definite at every iteration. This implementation uses a very simple strategy to maintain the positive definiteness of the reduced Hessian matrix, which is based upon calculating the smallest and largest eigenvalues of the reduced Hessian and using the second strategy mentioned in §4.2 to do the adjustment. Third, in our implementations, the new step length α_k in the line search is chosen by halving α_k, although more sophisticated strategies such as quadratic interpolation could be used.

As discussed in §4.1, the l_1 merit function is used to monitor the progress of iterates. When a trial step generated by these methods does not give descent on the merit function, the penalty parameter is increased sufficiently to assure descent, using the multiplicative strategy described in §4.3. For a tensor iteration, we assume that there is a previous iterate x^{k-1}; at the initial iteration both tensor methods use the SQP step instead.

For the tensor methods we use the exact second-order derivative of the constraint, $\nabla^2 c(x^k)$, to form the tensor term T_k, instead of using the interpolation approach. This means that $z_i = \nabla^2 c(x^k) s_i s_i$ or $\nabla^2 c(x^k) \hat{s}_i \hat{s}_i$ for the direct tensor method or the projected tensor method, respectively. As mentioned in §3.2, the two values of T_k are asymptotically equivalent as $(x^{k-1} - x^k)$ approaches 0. The reason for making this simplification, in the early stage of testing, is to rule out the possibility that approximation or round off errors from the interpolation approach might disguise the basic behavior of the tensor approach. In our previous experience, using this ideal choice versus the interpolation approach makes little practical difference.

Small problems. The small test problems that we use are divided into six classes. In the first class, small nonsingular (SN) problems, the majority of the test problems, were chosen from the standard test set of Hock and Schittkowski [15]. A few more problems were chosen from [20, 21, 7, 6, 22]. The dimensions of these problems range from two to ten variables or constraints.

Since it is difficult to find a large collection of singular constrained optimization test problems in the literature, we constructed almost all the singular test problems ourselves. Two approaches were used in the construction of singular problems. The first approach is to square one (SS1) or two (SS2) constraints of problems in the nonsingular test set, which results in problems with rank deficiency one or two, respectively. The second approach for generating singular problems is to add a linear term to the constraints of problems in the nonsingular problem test set. This linear term is chosen so that the new problems have the same solution as the original problems and the constraint Jacobian at the solution has rank deficiency one (SL*1) or two (SL*2). At the solution the normal Kuhn–Tucker conditions may (* =K) or may not (* =NK) be satisfied.

Table 1 gives a summary of the test results for small problems. We list the improvements of tensor methods over the SQP methods in columns 3 to 6. The improvement is measured by $(Tot_s - Tot_t)/Tot_s$, where Tot_s is the total amount of work (number of iterations or number of line searches, respectively) required by the

TABLE 1
Test results for small problems.

Problem class	Number of problems	DT vs SQP		PT vs SQP	
		Iteration improvement	Line search improvement	Iteration improvement	Line search improvement
NS	97	17.92%	9.49%	15.44%	13.80%
SS1	31	50.61%	43.63%	55.02%	43.77%
SLK1	18	45.35%	25.49%	52.89%	48.06%
SLNK1	13	30.99%	12.87%	33.66%	21.74%
SS2	19	44.35%	37.56%	44.90%	36.29%
SLK2	8	38.30%	36.97%	31.21%	26.03%

SQP method for solving all problems solved by both methods and Tot_t is the total amount of work required by the tensor method for solving all problems solved by both methods. The line search improvement is calculated in an analogous fashion.

The line search improvement statistics are not as important as the iteration improvement statistics. For each line search performed, only an extra function evaluation of the objective function and the constraint function is required, whereas each iteration in addition requires the evaluation of both the first and the second derivatives of the objective and constraint functions. We include the line search statistics here because we think they may give some indication of the possible behavior of the analogous methods that use approximations to second derivatives. For these methods, the number of gradient and function evaluations becomes more important, particularly when the function evaluations are expensive and similar in cost to the gradient evaluations.

Our test results on small nonsingular problems show that the tensor methods are equally or more efficient than the standard SQP method for most of the problems. The improvement in the total number of iterations over all problems in the test set is 17.92% for the direct tensor method and 15.44% for the projected tensor method. We should point out that many of these problems require just a few iterations for the standard method to solve; hence for these problems there is not much room for improvement. The consistency of the efficiency gains indicate that the tensor methods may be slightly preferable to the standard method for solving these sorts of nonlinear equality constrained optimization problems.

The tensor methods have a large improvement over the SQP method on the small singular test problems. The improvement in the number of iterations is generally 30–50%. The tensor methods almost always take considerably fewer iterations than the SQP method. The improvement seems slightly larger on singular problems with rank deficiency one than on singular problems with rank deficiency two (SS1 versus SS2 and SLK1 versus SLK2).

There were very few failures by the tensor methods or the SQP method for any of the small test problems sets and no appreciable difference in success rates among the three methods.

Large problems. The large test problems used in our tests include orthogonal matrix problems, orthogonal distance regression problems, optimal control problems, and eigenvalue problems. Most of these problems have about 50 to 250 variables. Large singular problems were created by applying the two construction approaches previously discussed to large nonsingular problems. In addition, a naturally rank deficient orthogonal regression problem from [18] was used. The main purpose of

testing on these larger problems was to see whether the improvements of the tensor methods over the standard method for small problems carry over to larger problems. It is natural to wonder about this since the tensor methods being used only add information in a one-dimensional subspace at each iteration.

On this test set of large problems, the tensor methods appear to have an even greater advantage over the SQP method than for small problems. On nonsingular problems the average efficiency advantage in the number of iterations summed over all the problems is 19.38% for the direct tensor method and 20.99% for the projected tensor method. The tensor methods have a consistent improvement over the SQP method on large singular problems. The improvement ranges from 30% to 90%, with an average of more than 70%. As an example, the test results for the singular orthogonal distance regression problem from [18] are given in Table 2.

TABLE 2
Version of orthogonal regression D with rank deficiency one.

n	m	SQP	DT	PT
		#iter/#ls	#iter/#ls	#iter/#ls
23	10	24/23	14/18	17/16
53	25	42/63	24/37	11/6
203	100	65/119	10/4	12/6

A few comments. The tests use the exact derivative values $\nabla^2 c(x^k)\bar{s}\bar{s}$ (note that $\bar{s}$ has different meanings in the direct method and the projected method) instead of their interpolation values to form the tensor terms, but a limited number of tests with the interpolation values indicate that using them instead makes at most a negligible difference in the performance of the tensor methods.

One may notice that the SQP method, with which we compared, implemented a simple global strategy. From our experience, usually as long as both methods use the same global strategy, the comparison is meaningful. More importantly, one should notice that this simple global strategy has a major difference from most global strategies implemented in SQP methods. It is that the penalty parameter in the merit function is updated in a multiplicative fashion instead of an additive fashion (see §4.3). This strategy is a key to the convergence of both the SQP method and the tensor methods on singular problems. For example, our versions of the SQP method and tensor methods can be driven without difficulty to reach a tight tolerance on a singular version of the orthogonal distance regression problem, while an SQP method that has a more sophisticated global strategy could not achieve this accuracy [18]. In addition, in tests using our methods but the standard additive penalty parameter update strategy, both the SQP and tensor methods were very inefficient for singular problems.

6. Summary and topics for future research. This paper has introduced tensor methods for nonlinear equality constrained optimization problems. These tensor methods have the same leading terms for storage and arithmetic cost per iteration as the standardly used SQP method. Implementations of full equality constrained optimization algorithms using slightly idealized tensor methods have been shown to be more efficient on both nonsingular and singular problems than the analogous implementation of the SQP method. The efficiency advantage of tensor methods is particularly large on singular problems and appears to be greater on large-scale problems than on small problems.

Based on these results, it would appear worthwhile to continue research on tensor methods for constrained optimization. The two main topics for future research would appear to be the practical implementation of tensor methods for constrained optimization and new tensor methods for constrained optimization. We discuss each of these briefly.

As seen in §5, our implementation of tensor methods is still at an early stage. Several directions can be pursued immediately to improve the current implementation or to make it more efficient: (1) the formation and solution of the tensor model using more than one past iterate can be put into the implementation; (2) the exact directional derivative term $\nabla^2 c(x^k)s_i s_i$ can be replaced by the interpolation value $2(c(x^{-i}) - c(x^k) - A_k^T s_i)$; (3) more efficient algorithms can be used for modifying the reduced Hessian matrix when it is not safely positive definite; (4) more efficient line search algorithms can be used for the global strategy. There are also several less straightforward directions that can be pursued in the future including the following: (1) the implementation can take advantage of the sparsity structure of the constraint gradient matrix for large problems; (2) other merit functions such as the l_2 merit function can be considered for the overall tensor methods; (3) implementations that deal with simple bound constraints or general inequality constraints can be considered; (4) stable ways of dealing with the Lagrange multiplier and the penalty parameter can be considered since they approach infinity for singular problems, which is a threat to numerical stability.

Also, new tensor methods can be developed. First, our current tensor methods can be improved. In our experience, our current tensor methods do not work well for problems whose singular vectors corresponding to the vanishing singular values of the constraint gradient matrix are discontinuous near the solution. This is because for these problems there is no stable way for our current tensor methods to incorporate the ideal directional derivative information into the tensor model. It seems, however, that one can make use of the gradient of the objective function to help in this situation. We are currently investigating this possibility. Second, the basic ideas of our tensor methods can be applied to the SQP model that uses a quasi-Newton Hessian approximation instead of the exact Hessian. We believe that with the techniques developed in this paper, this can be carried out in a straightforward fashion. The advantage of using the Hessian approximation is the low computational cost per iteration and relative fast convergence. This is a very attractive direction for our next step in the research on tensor methods for constrained optimization.

REFERENCES

[1] P. T. Boggs, R. H. Byrd, and R. B. Schnabel, *A stable and efficient algorithm for nonlinear orthogonal distance regression*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 1052–1078.

[2] A. Bouaricha, *Solving Large Sparse Systems of Nonlinear Equations and Nonlinear Least Squares Problems Using Tensor Methods on Sequential and Parallel Computers*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1992.

[3] R. H. Byrd, D. Feng, and R. B. Schnabel, *On Optimality Conditions for Singular Constrained Optimization*, Tech. report 95.03, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffett Field, CA, 1995.

[4] R. H. Byrd and J. Nocedal, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1991), pp. 285–323.

[5] R. M. Chamberlain, C. Lemarechal, H. C. Pedersen, and M. J. D. Powell, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Programming, 16 (1982), pp. 1–17.

[6] I. D. COOPE AND R. FLETCHER, *Some numerical experience with globally convergent algorithm for nonlinearly constrained optimization*, J. Optim. Theory Appl., 32 (1980), pp. 1–16.
[7] R. S. DEMBO, *A set of geometric programming test problems and their solutions*, Math. Programming, 10 (1976), pp. 192–213.
[8] D. FENG, *Tensor Methods for Constrained Optimization*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1993.
[9] D. FENG, P. D. FRANK, AND R. B. SCHNABEL, *Local convergence analysis of tensor methods for nonlinear equations*, Math. Programming, 62 (1993), pp. 427–459.
[10] D. FENG AND R. B. SCHNABEL, *Local Convergence Analysis of Tensor and SQP Methods for Singular Constrained Optimization*, Tech. report 95.10, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffett Field, CA, 1995.
[11] ———, *Tensor Methods for Equality Constrained Optimization*, Tech. report CU-CS-729-94, Computer Science Department, University of Colorado, Boulder, CO, 1994.
[12] R. FLETCHER, *Practical Methods of Optimization, Second Edition*, John Wiley, New York, 1989.
[13] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, London, New York, 1981.
[14] S. P. HAN, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11 (1976), pp. 263–282.
[15] W. HOCK AND K. SCHITTKOWSKI, *Test examples for nonlinear programming codes*, in Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, New York, 1981.
[16] W. P. HUFFMAN, R. G. MELVIN, D. P. YOUNG, F. T. JOHNSON, J. E. BUSSOLETTI, M. B. BIETERMAN, AND C. L. HILMES, *Practical Design and Optimization in Computational Fluid Dynamics*, AIAA paper 93-3111, American Institute of Aeronautics and Astronautics, 1993.
[17] U. KIRSCH, *Singular and local optima in structural optimization*, in 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, FL, September 1994, pp. 150–160.
[18] M. LALEE, J. NOCEDAL, AND T. PLANTENGA, *On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization*, Tech. report NAM 08, Department of EECS, Northwestern University, Evanston, IL, 1993.
[19] A. C. LORENC, *Analysis methods for numerical weather prediction*, Quart. J. R. Met. Soc., 122 (1986), pp. 1177–1194.
[20] B. A. MURTAGH, *A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints*, Math. Programming Studies, 16 (1982), pp. 84–117.
[21] E. O. OMOJOKUN, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1989.
[22] D. A. PIERRE AND M. J. LOWE, *Mathematical Programming via Augmented Lagrangians*, Addison–Wesley, Reading, MA, 1975.
[23] M. J. D. POWELL, *Variable metric methods for constrained optimization*, in Mathematical Programming The State of Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1983, pp. 288–311.
[24] R. B. SCHNABEL AND E. ESKOW, *A new modified Cholesky factorization*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 1136–1158.
[25] R. B. SCHNABEL AND P. D. FRANK, *Tensor methods for nonlinear equations*, SIAM J. Numer. Anal., 21 (1984), pp. 815–843.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 674–703, August 1996
© 1996 Society for Industrial and Applied Mathematics
009

CONVERGENCE PROPERTIES OF AN AUGMENTED LAGRANGIAN ALGORITHM FOR OPTIMIZATION WITH A COMBINATION OF GENERAL EQUALITY AND LINEAR CONSTRAINTS*

A. R. CONN†, N. GOULD‡, A. SARTENAER§, AND PH. L. TOINT§

Abstract. We consider the global and local convergence properties of a class of augmented Lagrangian methods for solving nonlinear programming problems. In these methods, linear and more general constraints are handled in different ways. The general constraints are combined with the objective function in an augmented Lagrangian. The iteration consists of solving a sequence of subproblems; in each subproblem the augmented Lagrangian is approximately minimized in the region defined by the linear constraints. A subproblem is terminated as soon as a stopping condition is satisfied. The stopping rules that we consider here encompass practical tests used in several existing packages for linearly constrained optimization. Our algorithm also allows different penalty parameters to be associated with disjoint subsets of the general constraints. In this paper, we analyze the convergence of the sequence of iterates generated by such an algorithm and prove global and fast linear convergence as well as show that potentially troublesome penalty parameters remain bounded away from zero.

Key words. constrained optimization, augmented Lagrangian methods, linear constraints, convergence theory

AMS subject classifications. 90C30, 65K05

1. Introduction.

In this paper, we consider the problem of calculating a local minimizer of the smooth function

$$f(x), \tag{1.1}$$

where x is required to satisfy the general equality constraints

$$c_i(x) = 0, \quad 1 \leq i \leq m \tag{1.2}$$

and the linear inequality constraints

$$Ax - b \geq 0. \tag{1.3}$$

Here f and c_i map $\Re^n$ into $\Re$, A is a p-by-n matrix, and $b \in \Re^p$.

* Received by the editors June 24, 1993; accepted for publication (in revised form) March 17, 1995. This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by Air Force Office of Scientific Research contract F49620-91-C-0079. The research of the third author was supported by the Belgian National Fund for Scientific Research. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

† IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598 (arconn @watson.ibm.com).

‡ Rutherford Appleton Laboratory, Chilton, Oxfordshire, England (n.gould@petterbox.rl.ac.uk). Current reports available by anonymous ftp from the directory "pub/reports" on joyous-gard.cc.rl.ac.uk (internet 130.246.9.91).

§ Department of Mathematics, Facultés Universitaires Notre-Dame de la Paix, 61, rue de Bruxelles, B-5000 Namur, Belgium (as@math.fundp.ac.be and pht@math.fundp.ac.be). Current reports available by anonymous ftp from the directory "pub/reports" on thales.math.fundp.ac.be (internet 138.48.20.102).

A classic technique for solving problem (1.1)–(1.3) is to minimize a suitable sequence of *augmented Lagrangian functions.* If we only consider the problem (1.1)–(1.2), these functions are defined by

$$\Phi(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{1}{2\mu}\sum_{i=1}^{m} c_i(x)^2, \tag{1.4}$$

where the components λ_i of the vector λ are known as *Lagrange multiplier estimates* and μ is known as the *penalty parameter* (see, for instance, Hestenes [18], Powell [23], and Bertsekas [3]). The question then arises of how to deal with the additional linear inequality constraints (1.3). The case where A is the identity matrix (that is, when (1.3) specifies bounds on the variables) has been considered by Conn, Gould, and Toint in [5] and [7]. They propose keeping these constraints explicitly outside the augmented Lagrangian formulation, handling them directly at the level of the augmented Lagrangian minimization. That is, a sequence of optimization problems, in which (1.4) is approximately minimized within the region defined by the simple bounds, is attempted. In this approach all linear inequalities other than bound constraints are converted to equations by introducing slack variables and incorporated in the augmented Lagrangian function. This strategy has been implemented and successfully applied within the LANCELOT package for large-scale nonlinear optimization (see Conn, Gould, and Toint [6]). However, such a method may be inefficient when linear constraints are present because there are a number of effective techniques specifically designed to handle such constraints directly (see Arioli et al. [1], Forsgren and Murray [14], Toint and Tuyttens [24], or Vanderbei and Carpenter [25], for instance). This is especially important for large-scale problems. The purpose of this paper is therefore to define and analyze an algorithm where the constraints (1.3) are kept outside the augmented Lagrangian and handled at the level of the subproblem minimization, thus allowing the use of specialized packages to solve the subproblem.

Our proposal extends the method of Conn, Gould, and Toint [5] in that not only bounds but general linear inequalities are treated separately. Fletcher [13, p. 295] remarks on the potential advantages of this strategy.

Furthermore, it is often worthwhile from the practical point of view to associate different penalty parameters with subsets of the general constraints (1.2) to reflect different degrees of nonlinearity. This possibility has been considered by many authors, including Fletcher [13, p. 292], Powell [23], and Bertsekas [3, p. 124]. In this case, the formulation of the augmented Lagrangian (1.4) can be refined. We partition the set of constraints (1.2) into q disjoint *subsets* $\{\mathcal{Q}_j\}_{j=1}^{q}$ and redefine the augmented Lagrangian as

$$\Phi(x,\lambda,\mu) = f(x) + \sum_{j=1}^{q}\sum_{i\in\mathcal{Q}_j}\left[\lambda_i c_i(x) + \frac{1}{2\mu_j}c_i(x)^2\right], \tag{1.5}$$

where μ is now a q-dimensional vector whose jth component is $\mu_j > 0$, the penalty parameter associated with subset $\mathcal{Q}_j$. Because of its potential usefulness, and because its analysis is difficult to infer directly from the single penalty parameter case, this refined formulation will be adopted in this paper.

The theory presented below handles the linear inequality constraints in a purely geometric way. Hence the same theory applies without modifications if linear equality constraints are also imposed and all the iterates are assumed to stay feasible with respect to these new constraints. It is indeed enough to apply the theory in the affine

subspace corresponding to this feasible set. As a consequence, linear constraints need not be included in the augmented Lagrangian and thus have the desirable property that they have no impact on the structure of its Hessian matrix.

The paper is organized as follows. In §2, we introduce our basic assumptions on the problem and the necessary terminology. Section 3 presents the proposed algorithm and the definition of a suitable stopping criterion for the subproblem. The global convergence analysis is developed in §4, while the rate of convergence is analyzed in §5. Second-order conditions are investigated in §6. Section 7 considers some possible extensions of the theory. Finally, some conclusions and perspectives are outlined in §8.

2. The problem and related terminology. We consider the problem stated in (1.1)–(1.3) and make the following assumptions.

AS1. The region $\mathcal{B} = \{x \mid Ax - b \geq 0\}$ is nonempty.

AS2. The functions $f(x)$ and $c_i(x)$ $(i = 1, \ldots, m)$ are twice continuously differentiable for all $x \in \mathcal{B}$.

Assumption AS1 is clearly necessary for the problem to make sense. We note that it does not prevent $\mathcal{B}$ from being unbounded.

We now introduce the notation that will be used throughout the paper.

Let $g(x)$ denote the gradient $\nabla_x f(x)$ of $f(x)$ and $H(x)$ denote its Hessian matrix $\nabla_{xx} f(x)$. We also define $J(x)$ to be the m-by-n Jacobian of $c(x)$, where

$$c(x) = [c_1(x), \ldots, c_m(x)]^T.$$

Hence

$$J(x)^T = [\nabla_x c_1(x), \ldots, \nabla_x c_m(x)].$$

Let $H_i(x)$ denote the Hessian matrix $\nabla_{xx} c_i(x)$ of $c_i(x)$. Finally, let $g^\ell(x, \lambda)$ and $H^\ell(x, \lambda)$, respectively, denote the gradient, $\nabla_x \ell(x, \lambda)$, and the Hessian matrix, $\nabla_{xx} \ell(x, \lambda)$, of the Lagrangian function

$$\ell(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i c_i(x).$$

We note that $\ell(x, \lambda)$ is the Lagrangian solely with respect to the c_i constraints. If we define *first-order Lagrange multiplier estimates* componentwise as

$$\bar{\lambda}(x, \lambda_{[\mathcal{Q}_j]}, \mu_j)_{[\mathcal{Q}_j]} = \lambda_{[\mathcal{Q}_j]} + c(x)_{[\mathcal{Q}_j]}/\mu_j \quad (j = 1, \ldots, q), \tag{2.1}$$

where $w_{[\mathcal{S}]}$ denotes the $|\mathcal{S}|$-dimensional subvector of w whose entries are indexed by the set $\mathcal{S}$, we shall use the identity

$$\begin{aligned} \nabla_x \Phi(x, \lambda, \mu) = &\ \nabla_x f(x) + \textstyle\sum_{j=1}^{q} \sum_{i \in \mathcal{Q}_j} \left[\lambda_i \nabla_x c_i(x) + \frac{1}{\mu_j} c_i(x) \nabla_x c_i(x) \right] \\ = &\ g^\ell(x, \bar{\lambda}(x, \lambda, \mu)). \end{aligned} \tag{2.2}$$

Now suppose that $\{x_k \in \mathcal{B}\}$, $\{\lambda_k\}$, and $\{\mu_k\}$ are infinite sequences of n-vectors, m-vectors, and positive q-vectors, respectively. For any function F, we shall use the notation that F_k denotes F evaluated with arguments x_k, λ_k, and/or μ_k as appropriate. So, for instance, using the identity (2.2), we have that

$$\nabla_x \Phi_k = \nabla_x \Phi(x_k, \lambda_k, \mu_k) = g^\ell(x_k, \bar{\lambda}_k), \tag{2.3}$$

where we have written (2.1) in the compact form

$$\bar{\lambda}_k = \bar{\lambda}(x_k, \lambda_k, \mu_k). \tag{2.4}$$

We denote the vector w at iteration k by w_k and its ith component by $w_{k,i}$. We also use $w_{k,[\mathcal{S}]}$ to denote the $|\mathcal{S}|$-dimensional subvector of w_k whose entries are indexed by $\mathcal{S}$.

Now let $\{x_k\}$, $k \in \mathcal{K}$, for some subset $\mathcal{K}$ of the natural numbers $\mathbf{N}$ be a convergent subsequence with limit point x_*. Then we denote the matrix whose rows are those of A corresponding to active constraints at x_*—that is, the constraints that are satisfied as equalities at x_*—by A_*. Furthermore, we choose Z_* to be a matrix whose columns form an orthonormal basis of the null space of A_*, that is,

$$A_* Z_* = 0 \quad \text{and} \quad Z_*^T Z_* = I.$$

We define the *least-squares Lagrange multiplier estimates* (corresponding to A_*) as

$$\lambda(x) \stackrel{\text{def}}{=} -((J(x)Z_*)^+)^T Z_*^T g(x) \tag{2.5}$$

at all points where the right generalized inverse

$$(J(x)Z_*)^+ \stackrel{\text{def}}{=} Z_*^T J(x)^T (J(x)Z_* Z_*^T J(x)^T)^{-1}$$

of $J(x)Z_*$ is well defined. We note that whenever $J(x)Z_*$ has full rank, $\lambda(x)$ is differentiable and its derivative is given in the following lemma.

LEMMA 2.1. *Suppose that* AS2 *holds. If $J(x)Z_*Z_*^T J(x)^T$ is nonsingular, $\lambda(x)$ is differentiable and its derivative is given by*

$$\nabla_x \lambda(x) = -((J(x)Z_*)^+)^T Z_*^T H^\ell(x, \lambda(x)) - (J(x)Z_* Z_*^T J(x)^T)^{-1} R(x), \tag{2.6}$$

where the ith row of $R(x)$ is $(Z_*^T g(x) + Z_*^T J(x)^T \lambda(x))^T Z_*^T H_i(x)$.

Proof. The result follows by observing that (2.5) may be rewritten as

$$r(x) - Z_*^T J(x)^T \lambda(x) = Z_*^T g(x) \quad \text{and} \quad J(x) Z_* r(x) = 0 \tag{2.7}$$

for some vector $r(x)$. Differentiating (2.7) and eliminating the derivative of $r(x)$ from the resulting equations give the required result. □

We stress that, as stated, the Lagrange multiplier estimate (2.5) is not directly calculable since it requires a priori knowledge of x_*. It is merely introduced as an analytical device.

Finally, the symbol $\|\cdot\|$ will denote the ℓ_2-norm or the induced matrix norm. We are now in position to describe more precisely the algorithm that we propose to use.

3. Statement of the algorithm. We consider the algorithmic model we wish to use to solve the problem (1.1)–(1.3). This model proceeds at iteration k by computing an iterate x_k that satisfies (1.3) and approximately solves the subproblem

$$\min_{x \in \mathcal{B}} \Phi(x, \lambda_k, \mu_k), \tag{3.1}$$

where the values of the Lagrange multipliers λ_k and penalty parameters μ_k are fixed for the subproblem. Subsequently we update the Lagrange multipliers and/or decrease the penalty parameters, depending on how much the constraint violation for (1.2) has

been reduced within each subset of the constraints. The motivation is simply to ensure global convergence by driving, in the worst case, the penalty parameters to zero, in which case the algorithms essentially reduce to the quadratic penalty function method (see, for example, Gould [15]). The tests on the size of the general constraint violation are designed to allow the multiplier updates to take over in the neighborhood of a stationary point.

The approximate minimization for problem (3.1) is performed in an *inner iteration* that is stopped as soon as its current iterate is "sufficiently critical." We propose to base this decision on the identification of the linear constraints that are "dominant" at x (even though they might not be active) and on a measure of criticality for the part of the problem where those constraints are irrelevant. Given $\omega \geq 0$, a criticality tolerance for the subproblem, we define, for a vector $x \in \mathcal{B}$, the set of *dominant constraints at* x as the constraints whose indices are in the set

$$D(x,\omega) \stackrel{\text{def}}{=} \{i \in \{1,\ldots,p\} \mid a_i^T x - b_i \leq \kappa_0 \omega\} \tag{3.2}$$

for some $\kappa_0 > 0$. Here $a_i^T \in \Re^n$ is the ith row of the matrix A and b_i is the corresponding component of the right-hand side vector b. Denoting by $A_{D(x,\omega)}$ the submatrix of A consisting of the row(s) whose index is in $D(x,\omega)$, we also define

$$N(x,\omega) = \{A_{D(x,\omega)}^T \xi \mid \xi \in \Re^{|D(x,\omega)|} \text{ and } \xi_i \leq 0 \quad (i = 1,\ldots,|D(x,\omega)|)\},$$

the cone spanned by the outward normals of the dominant constraints. The associated polar cone is then

$$T(x,\omega) = N(x,\omega)^0 = \mathrm{cl}\{d \mid d^T v \leq 0 \text{ for all } v \in N(x,\omega)\},$$

where $\mathrm{cl}(V)$ denotes the closure of the set V. The cone $T(x,\omega)$ is the tangent cone with respect to the dominant constraints at x for the tolerance ω. Note that $D(x,\omega)$ might be empty, in which case $A_{D(x,\omega)}$ is assumed to be zero, $N(x,\omega)$ reduces to the origin, and $T(x,\omega)$ is the full space.

We then formulate our "sufficient criticality" criterion for the subproblem as follows. We require that

$$\|P_{T(x_k,\omega_k)}(-\nabla_x \Phi_k)\| \leq \omega_k, \tag{3.3}$$

where $P_V(\cdot)$ is the projection onto the convex set V and ω_k is a suitable tolerance at iteration k. Once x_k satisfying (3.3) has been determined by the inner iteration, we denote

$$D_k = D(x_k,\omega_k), \quad N_k = N(x_k,\omega_k), \quad T_k = T(x_k,\omega_k). \tag{3.4}$$

For future reference, we define Z_k to be a matrix whose columns form an orthonormal basis of $\mathcal{V}_k$, the null space of A_{D_k}, and Y_k to be a matrix whose columns form an orthonormal basis of $\mathcal{W}_k = \mathcal{V}_k^{\perp}$. As above, we have that T_k is the full space and N_k reduces to the origin when D_k is empty. We note that, in this case, $Z_k = P_{T_k} = I$, the identity operator, and $Y_k = P_{N_k} = 0$. We also note that $\mathcal{V}_k \subseteq T_k$ and, hence, that

$$\|Z_k^T \nabla_x \Phi_k\| = \|Z_k Z_k^T \nabla_x \Phi_k\| \leq \|P_{T_k}(-\nabla_x \Phi_k)\|, \tag{3.5}$$

since $Z_k Z_k^T$ is the orthogonal projection onto $\mathcal{V}_k$.

It is important to note that the stopping rule (3.3) covers a number of more specific choices, including the rule used in much existing software for linearly constrained

optimization (such as MINOS [21], LSNNO [24], or VE14 and VE19 from the Harwell Subroutine Library [17]). The reader is referred to §7.2 for further details.

We are now in position to describe our algorithmic model more precisely. In this model, we define α_k to be the maximum penalty parameter at iteration k (see (3.10)). At this iteration, the parameters ω_k and η_k represent criticality and feasibility levels, respectively.

ALGORITHM 3.1.

Step 0 [Initialization]. A partition of the set $\{1,\ldots,m\}$ into q disjoint subsets $\{\mathcal{Q}_j\}_{j=1}^q$ is given, as well as initial vectors of Lagrange multiplier estimates λ_0 and positive penalty parameters μ_0 such that

$$\mu_{0,j} < 1 \quad (j = 1,\ldots,q). \tag{3.6}$$

The strictly positive constants κ_0, $\omega_* \ll 1$, $\eta_* \ll 1$, $\tau < 1$, $\alpha_\eta < 1$, and $\beta_\eta < 1$ are specified. Set $\alpha_0 = \max_{j=1,\ldots,q} \mu_{0,j}$, $\omega_0 = \alpha_0$, $\eta_0 = \alpha_0^{\alpha_\eta}$, and $k = 0$.

Step 1 [Inner iteration]. Find $x_k \in \mathcal{B}$ that approximately solves (3.1), i.e., such that (3.3) holds.

Step 2 [Test for convergence]. If $\|P_{T_k}(-\nabla_x \Phi_k)\| \leq \omega_*$ and $\|c(x_k)\| \leq \eta_*$, stop.

Step 3 [Disaggregated updates]. For $j = 1,\ldots,q$, execute Step 3a if

$$\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \eta_k \tag{3.7}$$

or Step 3b otherwise.

Step 3a [Update Lagrange multiplier estimates]. Set

$$\begin{aligned} \lambda_{k+1,[\mathcal{Q}_j]} &= \bar{\lambda}(x_k, \lambda_{k,[\mathcal{Q}_j]}, \mu_{k,j})_{[\mathcal{Q}_j]}, \\ \mu_{k+1,j} &= \mu_{k,j}. \end{aligned}$$

Step 3b [Reduce the penalty parameter]. Set

$$\begin{aligned} \lambda_{k+1,[\mathcal{Q}_j]} &= \lambda_{k,[\mathcal{Q}_j]}, \\ \mu_{k+1,j} &= \tau_{k,j}\mu_{k,j}, \end{aligned} \tag{3.8}$$

where

$$\tau_{k,j} = \begin{cases} \tau & \text{if } \mu_{k,j} = \alpha_k, \\ \min(\tau, \alpha_k) & \text{otherwise.} \end{cases} \tag{3.9}$$

Step 4 [Aggregated updates]. Define

$$\alpha_{k+1} = \max_{j=1,\ldots,q} \mu_{k+1,j}. \tag{3.10}$$

If

$$\alpha_{k+1} < \alpha_k, \tag{3.11}$$

then set

$$\begin{aligned} \omega_{k+1} &= \alpha_{k+1}, \\ \eta_{k+1} &= \alpha_{k+1}^{\alpha_\eta}; \end{aligned} \tag{3.12}$$

otherwise set

$$\begin{aligned} \omega_{k+1} &= \omega_k \alpha_{k+1}, \\ \eta_{k+1} &= \eta_k \alpha_{k+1}^{\beta_\eta}. \end{aligned} \tag{3.13}$$

Increment k by one and go to Step 1.

Algorithm 3.1 is specifically designed for the first-order estimate (2.1), a formula with potential advantages for large-scale computations. We refer the reader to §7.1 for a further discussion of a more flexible choice of the multipliers, covering, among others, the choice of the least-squares estimates $\lambda(x)$ as defined in (2.5).

We immediately verify that our algorithm is coherent, in that

$$\lim_{k\to\infty} \omega_k = \lim_{k\to\infty} \eta_k = 0. \tag{3.14}$$

Indeed, we obtain from (3.6) that $\alpha_k < 1$ for all k, and (3.14) then follows from (3.12) and (3.13) if α_k tends to zero or from (3.13) alone if α_k is bounded away from zero.

The restriction (3.6) is imposed to simplify the exposition. In a more practical setting, it may be ignored provided that the definition of α_0 and (3.10) are replaced by

$$\alpha_0 = \min\left(\gamma_s, \max_{j=1,\ldots,q} \mu_{0,j}\right) \quad \text{and} \quad \alpha_{k+1} = \min\left(\gamma_s, \max_{j=1,\ldots,q} \mu_{k+1,j}\right),$$

respectively, for some constant $\gamma_s \in (0,1)$, and that (3.11) is replaced by

$$\max_{j=1,\ldots,q} \mu_{k+1,j} < \max_{j=1,\ldots,q} \mu_{k,j}.$$

Algorithm 3.1 may be extended in other ways. For instance, one may respectively replace the definition of ω_0, the first equation in (3.12), and the first equation of (3.13) by

$$\omega_0 = \omega_s \alpha_0^{\alpha_\omega}, \quad \omega_{k+1} = \omega_s \alpha_{k+1}^{\alpha_\omega}, \qquad \omega_{k+1} = \omega_k \alpha_{k+1}^{\beta_\omega}$$

for some $\omega_s > 0$, $\alpha_\omega > \alpha_\eta$, and $\beta_\omega > \beta_\eta$. The definition of η_0 and the second equation in (3.12) may then be replaced by

$$\eta_0 = \eta_s \alpha_0^{\beta_\eta} \qquad \text{and} \qquad \eta_{k+1} = \eta_s \alpha_{k+1}^{\alpha_\eta} \tag{3.15}$$

for some $\eta_s > 0$. None of these extensions alter the results of the convergence theory developed below. The values used in the LANCELOT package in a similar context are $\alpha_\eta = \tau = \gamma_s = 0.1$ and $\beta_\eta = 0.9$ (relation (3.15) is also used with $\eta_s = 0.12589$, ensuring that $\eta_0 = 0.01$). The values $\omega_s = \alpha_\omega = \beta_\omega = 1$ and $\mu_{0,j} = 0.1$ $(j = 1, \ldots, q)$ also seem suitable. The parameters ω_* and η_* specify the final accuracy requested by the user.

Finally, the purpose of the update (3.9) is to put more emphasis on the feasibility of the constraints whose violation is proportionally higher in order to achieve a "balance" among all constraint violations. This balance then allows the true asymptotic regime of the algorithm to be reached. The advantage of (3.9) is that this balancing effect is obtained gradually and not enforced at every major iteration, as is the case in Powell [23]. Furthermore, Powell's approach increases the penalties corresponding to the constraints that are becoming too slowly feasible, based on the ℓ_∞-norm. Thus it is only when they have changed sufficiently so that they are all within the constraint violation tolerance that the Lagrange multiplier update is performed. By contrast, we update the multipliers of the well-behaved constraints (assuming they correspond to a particular partition—which is likely since that is, partly at least, why the partitions exist) independently of more badly behaved ones. In addition, by virtue of using the ℓ_2-norm, we do not give quite the same emphasis to the most violated constraint.

4. Global convergence analysis. We now proceed to show that Algorithm 3.1 is globally convergent under the following assumptions.

AS3. The iterates $\{x_k\}$ considered lie within a closed, bounded domain Ω.

AS4. The matrix $J(x_*)Z_*$ has column rank no smaller than m at any limit point x_* of the sequence $\{x_k\}$ considered in this paper.

We notice that AS3 implies that there exists at least a convergent subsequence of iterates but does not, of course, guarantee that this subsequence converges to a stationary point, i.e., that "the algorithm works." We also note that it is always satisfied in practice because the linear constraints (1.3) includes lower and upper bounds on the variables, either actual or implied by the finite precision of computer arithmetic.

Assumption AS4 guarantees that the dimension of the null space of A_* is large enough to provide the number of degrees of freedom that are necessary to satisfy the nonlinear constraints, and we require that the gradients of these constraints (projected onto this null space) are linearly independent at every limit point of the sequence of iterates. This assumption is the direct generalization of AS3 used by Conn, Gould, and Toint [5].

We shall analyze the convergence of our algorithm in the case where the convergence tolerances ω_* and η_* are both zero. We first need the following lemma, proving that (3.3) prevents both the reduced gradient of the augmented Lagrangian and its orthogonal complement from being arbitrarily large when ω_k is small.

LEMMA 4.1. *Let $\{x_k\} \subset \mathcal{B}, k \in \mathcal{K}$, be a sequence that converges to the point x_* and suppose that*

$$\|P_{T_k}(-\nabla_x \Phi_k)\| \leq \omega_k,$$

where the ω_k are positive scalar parameters that converge to zero as $k \in \mathcal{K}$ increases. Then

$$\|Z_*^T \nabla_x \Phi_k\| \leq \|Z_k^T \nabla_x \Phi_k\| \leq \omega_k \quad \text{and} \quad \|Y_k^T (x_k - x_*)\| \leq \kappa_1 \omega_k \tag{4.1}$$

for some $\kappa_1 > 0$ and for all $k \in \mathcal{K}$ sufficiently large.

Proof. Observe that, for $k \in \mathcal{K}$ sufficiently large, ω_k is sufficiently small and x_k sufficiently close to x_* to ensure that all the constraints in D_k are active at x_*. This implies that the subspace orthogonal to the normals of the dominant constraints at x_k, $\mathcal{V}_k$ contains the subspace orthogonal to the normals of the constraints active at x_*. Hence, we deduce that

$$\|Z_*^T \nabla_x \Phi_k\| \leq \|Z_k^T \nabla_x \Phi_k\| \leq \|P_{T_k}(-\nabla_x \Phi_k)\| \leq \omega_k,$$

where we have used (3.5) to obtain the second inequality and (3.3) to deduce the third. This proves the first part of (4.1).

We now turn to the second. If D_k is empty, then Y_k is the zero matrix and the second part of (4.1) immediately follows. Assume therefore that $D_k \neq \emptyset$. We first select a submatrix $\hat{A}_{D_k}$ of A_{D_k} that is of maximal full row-rank and note that the orthogonal projection onto the subspace spanned by the $\{a_i\}_{i \in D_k}$ is nothing but

$$Y_k Y_k^T = \hat{A}_{D_k}^T [\hat{A}_{D_k} \hat{A}_{D_k}^T]^{-1} \hat{A}_{D_k}.$$

Hence we obtain from the orthogonality of Y_k, the bound $|D_k| \leq p$, (3.2) and (3.4), and the fact that all constraints in D_k are active at x_* for k sufficiently large, that

$$\begin{aligned} \|Y_k^T (x_k - x_*)\| &\leq \|\hat{A}_{D_k}^T [\hat{A}_{D_k} \hat{A}_{D_k}^T]^{-1}\| . \|\hat{A}_{D_k}(x_k - x_*)\| \\ &\leq \|\hat{A}_{D_k}^T [\hat{A}_{D_k} \hat{A}_{D_k}^T]^{-1}\| \, p \, \kappa_0 \omega_k. \end{aligned} \tag{4.2}$$

But there are only a finite number of nonempty sets D_k for all possible choices of x_k and ω_k, and we may thus deduce the second part of (4.1) from (4.2) by defining

$$\kappa_1 = p\kappa_0 \min \|\hat{A}_{D_k}^T [\hat{A}_{D_k} \hat{A}_{D_k}^T]^{-1}\|,$$

where the minimum is taken on all possible choices of D_k and $\hat{A}_{D_k}$. □

We now examine the behavior of the sequence $\{\nabla_x \Phi_k\}$. We first recall a result extracted from the classical perturbation theory of convex optimization problems. This result is well known and can be found, for instance, in [12, pp. 14–17].

LEMMA 4.2. *Assume that U is a continuous point-to-set mapping from $S \subseteq \Re^\ell$ into the power set of $\Re^n$ such that the set $U(\theta)$ is convex and nonempty for each $\theta \in S$. Assume that the real-valued function $F(y, \theta)$ is defined and continuous on the space $\Re^n \times S$ and convex in y for each fixed θ. Then the real-valued function F_* defined by*

$$F_*(\theta) \stackrel{\text{def}}{=} \inf_{y \in U(\theta)} F(y, \theta)$$

is continuous on S.

We now show that, if it converges, the sequence $\{\nabla_x \Phi_k\}$ tends to a vector that is a linear combination of the rows of A_* with nonnegative coefficients.

LEMMA 4.3. *Let $\{x_k\} \subset \mathcal{B}$, $k \in \mathcal{K}$, be a sequence that converges to the point x_* and suppose that the gradients $\nabla_x \Phi_k$, $k \in \mathcal{K}$, converge to some limit $\nabla_x \Phi_*$. Assume furthermore that* (3.3) *holds for $k \in \mathcal{K}$ and that ω_k tends to zero as $k \in \mathcal{K}$ increases. Then*

$$\nabla_x \Phi_* = A_*^T \pi_*$$

for some vector $\pi_* \geq 0$, where A_* is the matrix whose rows are those of A corresponding to active constraints at x_*.

Proof. We first define

$$\sigma_k \stackrel{\text{def}}{=} \max_{\substack{A(x_k+d)-b \geq 0 \\ \|d\| \leq 1}} \left(-\nabla_x \Phi_k{}^T d\right) \tag{4.3}$$

with the aim of showing that this quantity tends to zero when $k \in \mathcal{K}$ increases. We obtain from (4.3), the Moreau decomposition [20] of $\nabla_x \Phi_k$, and the Cauchy–Schwarz inequality that

$$\begin{array}{rcl} \sigma_k & \leq & \max_{\substack{A(x_k+d)-b \geq 0 \\ \|d\| \leq 1}} P_{T_k}(-\nabla_x \Phi_k)^T d + \max_{\substack{A(x_k+d)-b \geq 0 \\ \|d\| \leq 1}} P_{N_k}(-\nabla_x \Phi_k)^T d \\ & \leq & \|P_{T_k}(-\nabla_x \Phi_k)\| + \max_{d \in B_k} P_{N_k}(-\nabla_x \Phi_k)^T d, \end{array} \tag{4.4}$$

where $B_k \stackrel{\text{def}}{=} \{d \in \Re^n \mid a_i^T(x_k + d) - b_i \geq 0 \quad (i \in D_k) \quad \text{and} \quad \|d\| \leq 1\}$. Since, for x_k sufficiently close to x_* and ω_k sufficiently small, all the constraints in D_k must be active at x_*, we have that N_k is included in the normal cone $N(x_*, 0)$ and therefore the vector $P_{N_k}(-\nabla_x \Phi_k)$ belongs to this normal cone. Moreover, since the maximization problem of the last right-hand side of (4.4) is a concave program, since x_* is feasible for (1.3) and since $\|x_* - x_k\| \leq 1$ for $k \in \mathcal{K}$ large enough, we thus deduce that $d = x_* - x_k$ is a global solution of this problem. Observing that

$$P_{N_k}(-\nabla_x \Phi_k)^T d = [Y_k Y_k^T P_{N_k}(-\nabla_x \Phi_k)]^T d = P_{N_k}(-\nabla_x \Phi_k)^T Y_k Y_k^T d,$$

we obtain

$$\max_{d\in B_k} P_{N_k}(-\nabla_x\Phi_k)^T d = \max_{d\in B_k} P_{N_k}(-\nabla_x\Phi_k)^T Y_k Y_k^T d \leq \|P_{N_k}(-\nabla_x\Phi_k)\|.\|Y_k^T(x_k - x_*)\|, \tag{4.5}$$

where we used the Cauchy–Schwarz inequality to deduce the last inequality. We may now apply Lemma 4.1 and deduce from the second part of (4.1), (4.5), and the contractive character of the projection onto a convex set containing the origin that

$$\max_{d\in B_k} P_{N_k}(-\nabla_x\Phi_k)^T d \leq \kappa_1\omega_k\|\nabla_x\Phi_k\|,$$

and thus, from (4.4) and our assumptions, that

$$\sigma_k \leq \omega_k + \kappa_1\omega_k\|\nabla_x\Phi_k\|.$$

Our assumption on the ω_k sequence then implies that σ_k converges to zero as k increases in $\mathcal{K}$.

Consider now the minimization problem

$$\begin{array}{ll} \min_{d\in\Re^n} & \nabla_x\Phi_*^T d \\ \text{subject to} & A(x_* + d) - b \geq 0, \\ & \|d\| \leq 1. \end{array} \tag{4.6}$$

Since the sequences $\{\nabla_x\Phi_k\}$ and $\{x_k\}$ converge to $\nabla_x\Phi_*$ and x_*, respectively, we deduce from Lemma 4.2 applied to the optimization problem (4.3) (with the choices $\theta^T = (\nabla_x\Phi^T, x^T)$, $U(\theta) = \{d \mid A(x+d) - b \geq 0,\ \|d\| \leq 1\}$, $y = d$, $F(y,\theta) = \nabla_x\Phi^T d$), and the convergence of the sequence σ_k to zero that the optimal value for problem (4.6) is zero. The vector $d = 0$ is thus a solution for problem (4.6) and satisfies

$$\nabla_x\Phi_* = A_*^T\pi_* - 2\zeta d = A_*^T\pi_*$$

for some vector $\pi_* \geq 0$, which ends the proof. □

The important part of our convergence analysis is the next lemma.

LEMMA 4.4. *Suppose that* AS1 *and* AS2 *hold. Let* $\{x_k\} \subset \mathcal{B}, k \in \mathcal{K}$, *be a sequence satisfying* AS3 *that converges to the point* x_* *for which* AS4 *holds and let* $\lambda_* = \lambda(x_*)$, *where* λ *satisfies* (2.5). *Assume that* $\{\lambda_k\}$, $k \in \mathcal{K}$, *is any sequence of vectors and that* $\{\mu_k\}$, $k \in \mathcal{K}$, *form a nonincreasing sequence of* q*-dimensional vectors. Suppose further that* (3.3) *holds where the* ω_k *are positive scalar parameters that converge to zero as* $k \in \mathcal{K}$ *increases. Then*

(i) *there are positive constants* κ_2 *and* κ_3 *such that*

$$\|\bar{\lambda}(x_k, \lambda_k, \mu_k) - \lambda_*\| \leq \kappa_2\omega_k + \kappa_3\|x_k - x_*\|, \tag{4.7}$$

$$\|\lambda(x_k) - \lambda_*\| \leq \kappa_3\|x_k - x_*\|, \tag{4.8}$$

and

$$\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \kappa_2\omega_k\mu_{k,j} + \mu_{k,j}\|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\| + \kappa_3\mu_{k,j}\|x_k - x_*\| \tag{4.9}$$

for all $j = 1, \ldots, q$ *and all* $k \in \mathcal{K}$ *sufficiently large.*

Suppose, in addition, that $c(x_*) = 0$. *Then*

(ii) x_* *is a Kuhn–Tucker point (first-order stationary point) for the problem* (1.1)–(1.3), λ_* *is the corresponding vector of Lagrange multipliers, and the sequences* $\{\bar{\lambda}(x_k, \lambda_k, \mu_k)\}$ *and* $\{\lambda(x_k)\}$ *converge to* λ_* *for* $k \in \mathcal{K}$;

(iii) *the gradients* $\nabla_x \Phi_k$ *converge to* $g^\ell(x_*, \lambda_*)$ *for* $k \in \mathcal{K}$.

Proof. As a consequence of AS2–AS4, we have that for $k \in \mathcal{K}$ sufficiently large, $(J_k Z_*)^+$ exists, is bounded, and converges to $(J(x_*)Z_*)^+$. Thus, we may write

$$\|((J_k Z_*)^+)^T\| \leq \kappa_2 \tag{4.10}$$

for some constant $\kappa_2 > 0$. Equations (2.3) and (2.4), the inner iteration termination criterion (3.3), and Lemma 4.1 give that

$$\|Z_*^T(g_k + J_k^T \bar{\lambda}_k)\| \leq \omega_k \tag{4.11}$$

for all $k \in \mathcal{K}$ large enough. By assumptions AS2, AS3, AS4, and (2.5), $\lambda(x)$ is bounded for all x in a neighborhood of x_*. Thus we may deduce from (2.5), (4.10), and (4.11) that

$$\begin{array}{rcl} \|\bar{\lambda}_k - \lambda(x_k)\| & = & \|((J_k Z_*)^+)^T Z_*^T g_k + \bar{\lambda}_k\| \\ & = & \|((J_k Z_*)^+)^T (Z_*^T g_k + (J_k Z_*)^T \bar{\lambda}_k)\| \\ & \leq & \|((J_k Z_*)^+)^T\| \omega_k \\ & \leq & \kappa_2 \omega_k. \end{array} \tag{4.12}$$

Moreover, from the integral mean value theorem and Lemma 2.1 we have that

$$\lambda(x_k) - \lambda(x_*) = \int_0^1 \nabla_x \lambda(x(s)) ds \cdot (x_k - x_*), \tag{4.13}$$

where $\nabla_x \lambda(x)$ is given by equation (2.6) and $x(s) = x_k + s(x_* - x_k)$. Now the terms within the integral sign are bounded for all x sufficiently close to x_* and hence (4.13) gives

$$\|\lambda(x_k) - \lambda_*\| \leq \kappa_3 \|x_k - x_*\| \tag{4.14}$$

for all $k \in \mathcal{K}$ sufficiently large and for some constant $\kappa_3 > 0$, which implies inequality (4.8). We then have that $\lambda(x_k)$ converges to λ_*. Combining (4.12) and (4.14) we obtain

$$\|\bar{\lambda}_k - \lambda_*\| \leq \|\bar{\lambda}_k - \lambda(x_k)\| + \|\lambda(x_k) - \lambda_*\| \leq \kappa_2 \omega_k + \kappa_3 \|x_k - x_*\|, \tag{4.15}$$

which gives the required inequality (4.7). Then, since by assumption ω_k tends to zero as k increases, (4.15) implies that $\bar{\lambda}_k$ converges to λ_* and therefore, from the identity (2.3), $\nabla_x \Phi_k$ converges to $g^\ell(x_*, \lambda_*)$. Furthermore, multiplying (2.1) by $\mu_{k,j}$ we obtain

$$c(x_k)_{[\mathcal{Q}_j]} = \mu_{k,j}((\bar{\lambda}_k - \lambda_*)_{[\mathcal{Q}_j]} + (\lambda_* - \lambda_k)_{[\mathcal{Q}_j]}). \tag{4.16}$$

Taking norms of (4.16) and using (4.15), we derive (4.9).

Now suppose that

$$c(x_*) = 0. \tag{4.17}$$

Lemma 4.3 and the convergence of $\nabla_x \Phi_k$ to $g^\ell(x_*, \lambda_*)$ give that

$$g(x_*) + J(x_*)^T \lambda_* = A_*^T \pi_*$$

for some vector $\pi_* \geq 0$. This last equation and (4.17) show that x_* is a Kuhn–Tucker point and λ_* is the corresponding set of Lagrange multipliers. Moreover, (4.7) and (4.8) ensure the convergence of the sequences $\{\bar{\lambda}(x_k, \lambda_k, \mu_k)\}$ and $\{\lambda(x_k)\}$ to λ_* for $k \in \mathcal{K}$. Hence the lemma is proved. □

We finally require the following lemma in the proof of global convergence, which shows that the Lagrange multiplier estimates cannot behave too badly.

LEMMA 4.5. *Suppose that, for some j $(1 \leq j \leq q)$, $\mu_{k,j}$ converges to zero as k increases when Algorithm* 3.1 *is executed. Then the product $\mu_{k,j}\|\lambda_{k,[\mathcal{Q}_j]}\|$ converges to zero.*

Proof. As $\mu_{k,j}$ converges to zero, Step 3b must be executed infinitely often for the jth subset. Let $\mathcal{K}_j = \{k_0, k_1, k_2, \ldots\}$ be the set of indices of the iterations in which Step 3b is executed.

We consider how the jth subset of Lagrange multiplier estimates changes between two successive iterations indexed in the set $\mathcal{K}_j$. First note that $\lambda_{k_v+1,[\mathcal{Q}_j]} = \lambda_{k_v,[\mathcal{Q}_j]}$. At iteration $k_v + t$, for $k_v < k_v + t \leq k_{v+1}$, we have

$$\lambda_{k_v+t,[\mathcal{Q}_j]} = \lambda_{k_v,[\mathcal{Q}_j]} + \sum_{l=1}^{t-1} \frac{c(x_{k_v+l})_{[\mathcal{Q}_j]}}{\mu_{k_v+l,j}}, \tag{4.18}$$

where the summation is null if $t = 1$ and

$$\mu_{k_{v+1},j} = \mu_{k_v+t,j} = \mu_{k_v+1,j} = \tau_{k_v,j}\,\mu_{k_v,j}. \tag{4.19}$$

Substituting (4.19) into (4.18), multiplying both sides by $\mu_{k_v+t,j}$, taking norms, and using (3.9) yield

$$\mu_{k_v+t,j}\|\lambda_{k_v+t,[\mathcal{Q}_j]}\| \leq \tau\mu_{k_v,j}\|\lambda_{k_v,[\mathcal{Q}_j]}\| + \sum_{l=1}^{t-1} \|c(x_{k_v+l})_{[\mathcal{Q}_j]}\|$$

and hence

$$\mu_{k_v+t,j}\|\lambda_{k_v+t,[\mathcal{Q}_j]}\| \leq \tau\mu_{k_v,j}\|\lambda_{k_v,[\mathcal{Q}_j]}\| + \sum_{l=1}^{k_{v+1}-k_v-1} \|c(x_{k_v+l})_{[\mathcal{Q}_j]}\|.$$

Using the fact that (3.7) holds for $k_v + 1 \leq k_v + l \leq k_{v+1} - 1$, we deduce that

$$\begin{aligned} \mu_{k_v+t,j}\|\lambda_{k_v+t,[\mathcal{Q}_j]}\| &\leq \tau\mu_{k_v,j}\|\lambda_{k_v,[\mathcal{Q}_j]}\| + \sum_{l=1}^{k_{v+1}-k_v-1} \eta_{k_v+l} \\ &\leq \tau\mu_{k_v,j}\|\lambda_{k_v,[\mathcal{Q}_j]}\| + \sum_{l=1}^{\infty} \eta_{k_v+l}. \end{aligned}$$

Now defining

$$\delta_v \overset{\text{def}}{=} \mu_{k_v,j}\|\lambda_{k_v,[\mathcal{Q}_j]}\| \quad \text{and} \quad \rho_v \overset{\text{def}}{=} \sum_{l=1}^{\infty} \eta_{k_v+l}, \tag{4.20}$$

we obtain that

$$\mu_{k_v+t,j}\|\lambda_{k_v+t,[\mathcal{Q}_j]}\| \leq \tau\delta_v + \rho_v \tag{4.21}$$

for all t such that $k_v < k_v + t \leq k_{v+1}$ and, in particular,

$$\delta_{v+1} \leq \tau \delta_v + \rho_v. \tag{4.22}$$

Thus, from (4.22) and the inequality $\tau < 1$, if ρ_v converges to zero, then δ_v and hence, from (4.21), $\mu_{k_v+t,j} \|\lambda_{k_v+t,[\mathcal{Q}_j]}\|$ both converge to zero. To complete the proof it therefore suffices to show that ρ_v converges to zero as v tends to infinity.

Suppose first that α_k is bounded away from zero. Then we must have that (3.13) is used for all k sufficiently large, with $\alpha_{k+1} = \alpha_{\min}$ for some $\alpha_{\min} \in (0, 1)$. This and the definition of ρ_v in (4.20) imply that

$$\rho_v = \frac{\eta_{k_v+1}}{1 - \alpha_{\min}^{\beta_\eta}}$$

for sufficiently large v. Since (3.13) also guarantees that η_k tends to zero, we deduce that ρ_v converges to zero. This completes the proof for the first case.

Suppose now that α_k converges to zero. This implies that each of the q independent penalty parameters is reduced an infinite number of times. Consider the progress of α_k over the course of q successive decreases (3.11). Since (3.11) happens only when the currently largest penalty parameter, $\mu_{k,j}$ say, is reduced, as (3.9) requires that this penalty parameter is reduced by τ and because there can only possibly be at most $q - 1$ other penalty parameters in the interval $(\tau\mu_{k,j}, \mu_{k,j}]$, it follows that α_k must be reduced by at least τ over q successive decreases (3.11). Thus, considering the possible outcomes (3.12) and (3.13), each η_{k_v+l} must be bounded by a quantity of the form $(\tau^i \alpha_{k_v})^{\alpha_\eta + t\beta_\eta}$ for some indices i and t. Furthermore, at most q such terms can involve any particular i and t. Therefore, since $\tau\alpha_{k_v} < 1$, we obtain that

$$\begin{aligned}
\rho_v &\leq q \sum_{i=0}^{\infty} \sum_{t=0}^{\infty} (\tau^i \alpha_{k_v})^{\alpha_\eta + t\beta_\eta} \\
&= q \sum_{i=0}^{\infty} \frac{(\tau^i \alpha_{k_v})^{\alpha_\eta}}{1 - (\tau^i \alpha_{k_v})^{\beta_\eta}} \\
&\leq q \sum_{i=0}^{\infty} \frac{(\tau^i \alpha_{k_v})^{\alpha_\eta}}{1 - (\tau \alpha_{k_v})^{\beta_\eta}} \\
&= q \frac{\alpha_{k_v}^{\alpha_\eta}}{(1 - \tau^{\alpha_\eta})(1 - (\tau \alpha_{k_v})^{\beta_\eta})}.
\end{aligned}$$

Thus we see that, as α_{k_v} converges to zero, so does ρ_v, completing the proof for the second case. □

We can now derive the desired global convergence property of Algorithm 3.1, which is analogous to Theorem 4.4 in Conn, Gould, and Toint [5].

THEOREM 4.6. *Assume that* AS1 *and* AS2 *hold. Let x_* be any limit point of the sequence $\{x_k\}$ generated by Algorithm* 3.1 *of* §3 *for which* AS3 *and* AS4 *hold, and let $\mathcal{K}$ be the set of indices of an infinite subsequence of the x_k whose limit is x_*. Finally, let $\lambda_* = \lambda(x_*)$. Then conclusions* (i), (ii) *and* (iii) *of Lemma* 4.4 *hold.*

Proof. Our assumptions are sufficient to reach the conclusions of part (i) of Lemma 4.4. We now show that $c(x_*)_{[\mathcal{Q}_j]} = 0$ for $j = 1, \ldots, q$ and therefore that $c(x_*) = 0$. To see this, we consider a j $(1 \leq j \leq q)$ and analyze two separate cases.

The first case is when $\mu_{k,j}$ is bounded away from zero. Hence Step 3a must be executed at every iteration for k sufficiently large, implying that (3.7) is always

satisfied for k large enough. We then deduce from (3.14) that $c(x_k)_{[\mathcal{Q}_j]}$ converge to zero.

The second case is when $\mu_{k,j}$ converges to zero. Then Lemma 4.5 shows that $\mu_{k,j}\|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\|$ tends to zero. Using this limit and (3.14) in (4.9), we obtain that $c(x_k)_{[\mathcal{Q}_j]}$ tends to zero, as desired.

As a consequence, conclusions (ii) and (iii) of Lemma 4.4 hold. □

We finally note that global convergence of Algorithm 3.1 can be proved under much weaker assumptions on $\tau_{k,j}$ and ω_k. The reader is again referred to Conn et al. [9] for further details.

5. Asymptotic convergence analysis. The distinction between dominant and nondominant (*floating*) linear inequality constraints has some implications in terms of the identification of those constraints that are active at a limit point of the sequence of iterates generated by the algorithm. Given such a point x_* we know from Theorem 4.6 that it is critical, that is, that $-g^\ell(x_*, \lambda_*) \in N_* = N(x_*, 0)$ for the corresponding Lagrange multipliers λ_*. If we now consider a linear constraint with index $i \in \{1, \ldots, p\}$ that is active at x_*, we may define the normal cone $N_*^{[i]}$ to be the cone spanned by the outwards normals to all linear inequality constraints active at x_*, except the ith one. We then say that the ith linear inequality constraint is *strongly active* at x_* if $-g^\ell(x_*, \lambda_*) \notin N_*^{[i]}$. In other words, the ith constraint is strongly active at a critical point if this point ceases to be critical when this constraint is ignored. Let us denote by $S(x_*)$ the set of strongly active constraints at x_*. All nonstrongly active constraints at x_* are called *weakly active* at x_*. We now prove the reasonable result that all strongly active constraints at a limit point x_* are dominant for k large enough.

THEOREM 5.1. *Assume that* AS1–AS3 *hold. Let* $\{x_k\}$, $k \in \mathcal{K}$, *be a convergent subsequence of iterates produced by Algorithm* 3.1, *whose limit point is* x_* *with corresponding Lagrange multipliers* λ_*. *Assume furthermore that* AS4 *holds at* x_*. *Then*

$$S(x_*) \subseteq D_k$$

for all k *sufficiently large.*

Proof. Consider a linear inequality constraint $i \in S(x_*)$. Then, by the definition of this latter set, we have that $-g^\ell(x_*, \lambda_*) \notin N_*^{[i]}$. Since Theorem 4.6 guarantees that $\nabla_x \Phi_k$ converges to $g^\ell(x_*, \lambda_*)$ and since $N_*^{[i]}$ is closed, we have that $-\nabla_x \Phi_k \notin N_*^{[i]}$ for $k \in \mathcal{K}$ large enough. Therefore, one obtains from the Moreau decomposition [20] of $-\nabla_x \Phi_k$ that

$$\|P_{T_*^{[i]}}(-\nabla_x \Phi_k)\| \geq \epsilon \tag{5.1}$$

for some $\epsilon > 0$ and for all sufficiently large $k \in \mathcal{K}$, where $T_*^{[i]} = [N_*^{[i]}]^0$. We have also from (3.3) that $\|P_{T_k}(-\nabla_x \Phi_k)\|$ is arbitrarily small because ω_k tends to zero (see (3.14)). Assume now that, for some arbitrarily large $k \in \mathcal{K}$, we have $i \notin D_k$. This implies that $T_*^{[i]} \subseteq T_k$ and, hence, that (5.1) is impossible. We therefore deduce that i must belong to D_k, which proves the theorem. □

This result is important and is the generalization of Theorem 5.4 by Conn, Gould, and Toint [5]. It can also be interpreted as a means of active constraint identification, as is clear from the following easy corollary.

COROLLARY 5.2. *Suppose that the conditions of Theorem* 5.1 *hold. Assume furthermore that all linear inequality constraints active at* x_* *have linearly independent normals and are nondegenerate in the sense that*

$$-g^\ell(x_*, \lambda_*) \in \mathrm{ri}[N_*], \tag{5.2}$$

where $\mathrm{ri}[V]$ *denotes the relative interior of a convex set* V. *Then* D_k *is identical to the set of active linear inequality constraints at* x_* *for all* $k \in \mathcal{K}$ *sufficiently large.*

Proof. The nondegeneracy assumption and the linear independence of the active constraints normals imply that λ_* is unique and only has strictly negative components. Therefore each of the active linear inequality constraints at x_* is strongly active at x_*, and the desired conclusion follows from Theorem 5.1. □

We note here that the nondegeneracy assumption corresponds to strict complementarity slackness in our context (see, for instance, Dunn [11] or Burke, Moré, and Toraldo [4]).

We now make some additional assumptions before pursuing our local convergence analysis. We intend to show that all penalty parameters are bounded away from zero.

AS5. The second derivatives of the functions $f(x)$ and $c_i(x)$ $(1 \leq i \leq m)$ are Lipschitz continuous at any limit point x_* of the sequence of iterates $\{x_k\}$.

AS6. Suppose that (x_*, λ_*) is a Kuhn–Tucker point for problem (1.1)–(1.3), and let $\mathcal{I}$ be any subset of the linear inequality constraints that are active at x_* that contains all strongly active constraints ($S(x_*) \subseteq \mathcal{I}$) plus an arbitrary subset of weakly active constraints at x_*. Then, if the columns of the matrix Z form an orthonormal basis of the subspace orthogonal to the normals of the constraints in $\mathcal{I}$, we assume that the matrix

$$\begin{pmatrix} Z^T H^\ell(x_*, \lambda_*) Z & Z^T J(x_*)^T \\ J(x_*) Z & 0 \end{pmatrix}$$

is nonsingular for all possible choices of the weakly active constraints in the set $\mathcal{I}$.

We note that AS6 implies AS4 and seems reasonable in that the definition of strongly and weakly active constraints may vary with small perturbations in the problem, for instance when $g^\ell(x_*, \lambda_*)$ lies in one of the extreme faces of the cone N_*. Our assumption might be seen as a safeguard against the possible effect of all such perturbations.

We now make the distinction between the subsets for which the penalty parameter converges to zero and those for which it stays bounded away from zero. We define

$$\mathcal{Z} \stackrel{\text{def}}{=} \{j \in \{1, \ldots, q\} \mid \lim_{k \to \infty} \mu_{k,j} = 0\} \quad \text{and} \quad \mathcal{P} \stackrel{\text{def}}{=} \{1, \ldots, q\} \setminus \mathcal{Z}.$$

We also denote

$$\mu_{k,\mathcal{Z}} \stackrel{\text{def}}{=} \max_{j \in \mathcal{Z}} \mu_{k,j} \tag{5.3}$$

and

$$\rho_k \stackrel{\text{def}}{=} \sum_{j \in \mathcal{Z}} \mu_{k,j} \|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\|.$$

We now prove an analog to Lemma 5.1 by Conn, Gould, and Toint [5] that is suitable for our more general framework.

LEMMA 5.3. *Assume that* AS1–AS3 *hold. Let* $\{x_k\}$, $k \in \mathcal{K}$, *be a convergent subsequence of iterates produced by Algorithm* 3.1, *whose limit point is* x_* *with corresponding Lagrange multipliers* λ_*. *Assume that* AS5 *and* AS6 *hold at* x_*. *Assume furthermore that* $\mathcal{Z} \neq \emptyset$.

(i) *If* $\mathcal{P} = \emptyset$, *there are positive constants* $\bar{\alpha} < 1$, κ_4, κ_5, κ_6, κ_7 *and an integer* k_1 *such that, if* $\alpha_{k_1} \leq \bar{\alpha}$, *then*

$$\|x_k - x_*\| \leq \kappa_4 \omega_k + \kappa_5 \alpha_k \|\lambda_k - \lambda_*\|, \tag{5.4}$$

$$\|\bar{\lambda}(x_k, \lambda_k, \mu_k) - \lambda_*\| \leq \kappa_6 \omega_k + \kappa_7 \alpha_k \|\lambda_k - \lambda_*\|, \tag{5.5}$$

and

$$\|c(x_k)\| \leq q\kappa_6 \omega_k \alpha_k + q\alpha_k(1 + \kappa_7 \alpha_k)\|\lambda_k - \lambda_*\| \tag{5.6}$$

for all $k \geq k_1$, $k \in \mathcal{K}$.

(ii) *If, on the other hand,* $\mathcal{P} \neq \emptyset$, *there are positive constants* $\bar{\alpha} < 1$, κ_4, κ_5, κ_6, κ_7 *and an integer* k_1 *such that, if* $\mu_{k_1,\mathcal{Z}} \leq \bar{\alpha}$, *then*

$$\|(\bar{\lambda}(x_k, \lambda_k, \mu_k) - \lambda_*)_{[\mathcal{Q}_j]}\| \leq \kappa_6 \eta_k + \kappa_7 \rho_k \tag{5.7}$$

and

$$\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \kappa_6 \eta_k \mu_{k,\mathcal{Z}} + (1 + \kappa_7 \mu_{k,\mathcal{Z}})\rho_k \tag{5.8}$$

for all $k \geq k_1$, $k \in \mathcal{K}$, *and all* $j \in \mathcal{Z}$.

Proof. We will denote the gradient and Hessian of the Lagrangian function, taken with respect to x, at the limit point (x_*, λ_*) by g_*^ℓ and H_*^ℓ, respectively. Similarly, J_* will denote $J(x_*)$. We also define $\delta_k = x_k - x_*$. We observe that the assumptions of the lemma guarantee that Theorem 4.6 can be used.

We first note that there is only a finite number of possible D_k, and we may thus consider subsequences of $\mathcal{K}$ such that D_k is constant in each subsequence. We also note that each $k \in \mathcal{K}$ belongs to a unique such subsequence. To prove our result, it is thus sufficient to consider an arbitrary infinite subsequence $\bar{\mathcal{K}}$ such that, for $k \in \bar{\mathcal{K}}$, D_k is independent of k. This "constant" index set will be denoted by D. As a consequence, the cones N_k and T_k, the subspaces $\mathcal{V}_k$ and $\mathcal{W}_k$, and the orthogonal matrices Z_k and Y_k are also independent of k; they are denoted by N, T, $\mathcal{V}$, $\mathcal{W}$, Z, and Y, respectively.

Using (2.3) and Taylor's expansion around x_*, we obtain that

$$\begin{aligned} \nabla_x \Phi_k &= g_k + J_k^T \bar{\lambda}_k \\ &= g(x_*) + H(x_*)\delta_k + J_*^T \bar{\lambda}_k + \textstyle\sum_{i=1}^m \bar{\lambda}_{k,i} H_i(x_*)\delta_k + r_1(x_k, x_*, \bar{\lambda}_k) \\ &= g_*^\ell + H_*^\ell \delta_k + J_*^T(\bar{\lambda}_k - \lambda_*) + r_1(x_k, x_*, \bar{\lambda}_k) + r_2(x_k, x_*, \bar{\lambda}_k, \lambda_*), \end{aligned} \tag{5.9}$$

where

$$r_1(x_k, x_*, \bar{\lambda}_k) \stackrel{\text{def}}{=} \int_0^1 [H^\ell(x_* + s\delta_k, \bar{\lambda}_k) - H^\ell(x_*, \bar{\lambda}_k)]\delta_k \, ds$$

and

$$r_2(x_k, x_*, \bar{\lambda}_k, \lambda_*) \stackrel{\text{def}}{=} \sum_{i=1}^m (\bar{\lambda}_{k,i} - \lambda_{*,i}) H_i(x_*)\delta_k.$$

The boundedness and Lipschitz continuity of the Hessian matrices of f and c_i in a neighborhood of x_* together with the convergence of $\bar{\lambda}_k$ to λ_* then imply that

$$\|r_1(x_k, x_*, \bar{\lambda}_k)\| \leq \kappa_8 \|\delta_k\|^2 \tag{5.10}$$

and

$$\|r_2(x_k, x_*, \bar{\lambda}_k, \lambda_*)\| \leq \kappa_9 \|\delta_k\| \, \|\bar{\lambda}_k - \lambda_*\| \tag{5.11}$$

for some positive constants κ_8 and κ_9. Moreover, using Taylor's expansion again, along with the fact that Theorem 4.6 ensures the equality $c(x_*) = 0$, we obtain that

$$c(x_k) = J_* \delta_k + r_3(x_k, x_*), \tag{5.12}$$

where

$$[r_3(x_k, x_*)]_i = \int_0^1 s \int_0^1 \delta_k^T H_i(x_* + ts\delta_k)\delta_k \, dt \, ds$$

(see Gruver and Sachs [16, p. 11]). The boundedness of the Hessian matrices of the c_i in a neighborhood of x_* then gives that

$$\|r_3(x_k, x_*)\| \leq \kappa_{10} \|\delta_k\|^2 \tag{5.13}$$

for some positive constant κ_{10}. Combining (5.9) and (5.12), we obtain

$$\begin{pmatrix} H_*^\ell & J_*^T \\ J_* & 0 \end{pmatrix} \begin{pmatrix} \delta_k \\ \bar{\lambda}_k - \lambda_* \end{pmatrix} = \begin{pmatrix} \nabla_x \Phi_k - g_*^\ell \\ c(x_k) \end{pmatrix} - \begin{pmatrix} r_1 + r_2 \\ r_3 \end{pmatrix}, \tag{5.14}$$

where we have suppressed the arguments of the residuals r_1, r_2, and r_3 for brevity. Using the orthogonal decomposition of $\Re^n$ into $\mathcal{V} \oplus \mathcal{W}$ and defining

$$Q = \begin{pmatrix} Z & Y & 0 \\ 0 & 0 & I \end{pmatrix},$$

we may rewrite (5.14) as

$$Q^T \begin{pmatrix} H_*^\ell & J_*^T \\ J_* & 0 \end{pmatrix} Q Q^T \begin{pmatrix} \delta_k \\ \bar{\lambda}_k - \lambda_* \end{pmatrix} = Q^T \begin{pmatrix} \nabla_x \Phi_k - g_*^\ell \\ c(x_k) \end{pmatrix} - Q^T \begin{pmatrix} r_4 \\ r_3 \end{pmatrix},$$

where $r_4 \stackrel{\text{def}}{=} r_1 + r_2$. Expanding this last equation gives that

$$\begin{aligned} &\begin{pmatrix} Z^T H_*^\ell Z & Z^T H_*^\ell Y & Z^T J_*^T \\ Y^T H_*^\ell Z & Y^T H_*^\ell Y & Y^T J_*^T \\ J_* Z & J_* Y & 0 \end{pmatrix} \begin{pmatrix} Z^T \delta_k \\ Y^T \delta_k \\ \bar{\lambda}_k - \lambda_* \end{pmatrix} \\ &= \begin{pmatrix} Z^T (\nabla_x \Phi_k - g_*^\ell) \\ Y^T (\nabla_x \Phi_k - g_*^\ell) \\ c(x_k) \end{pmatrix} - \begin{pmatrix} Z^T r_4 \\ Y^T r_4 \\ r_3 \end{pmatrix}. \end{aligned} \tag{5.15}$$

We now observe that (3.3), the inclusion $\mathcal{V} \subseteq T$, and the fact that ω_k tends to zero imply that

$$Z^T g_*^\ell = 0. \tag{5.16}$$

Substituting (5.16) in (5.15), removing the middle horizontal block, and rearranging the terms of this latter equation then yield that

$$\begin{pmatrix} Z^T H_*^\ell Z & Z^T J_*^T \\ J_* Z & 0 \end{pmatrix} \begin{pmatrix} Z^T \delta_k \\ \bar{\lambda}_k - \lambda_* \end{pmatrix} = \begin{pmatrix} Z^T (\nabla_x \Phi_k - H_*^\ell Y Y^T \delta_k) \\ c(x_k) - J_* Y Y^T \delta_k \end{pmatrix} - \begin{pmatrix} Z^T r_4 \\ r_3 \end{pmatrix}. \tag{5.17}$$

Roughly speaking, we now proceed by showing that the right-hand side of this relation is of the order of $\theta_k + \rho_k$, where

$$\theta_k \stackrel{\text{def}}{=} \begin{cases} \omega_k & \text{if } \mathcal{P} = \emptyset, \\ \eta_k & \text{if } \mathcal{P} \neq \emptyset. \end{cases} \tag{5.18}$$

We will then ensure that the vector on the left-hand side is of the same size, which is essentially the result we aim to prove. We first observe that

$$\|\delta_k\| = \|Z Z^T \delta_k + Y Y^T \delta_k\| \leq \|Z^T \delta_k\| + \kappa_1 \omega_k \tag{5.19}$$

from (4.1). We then obtain from (4.7) and (5.19) that

$$\|\bar{\lambda}_k - \lambda_*\| \leq \kappa_{11} \omega_k + \kappa_3 \|Z^T \delta_k\|, \tag{5.20}$$

where $\kappa_{11} = \kappa_2 + \kappa_3 \kappa_1$. Furthermore, from (5.10), (5.11), (5.13), (5.19), and (5.20),

$$\left\| \begin{pmatrix} Z^T r_4 \\ r_3 \end{pmatrix} \right\| \leq \kappa_{12} \|Z^T \delta_k\|^2 + \kappa_{13} \|Z^T \delta_k\| \omega_k + \kappa_{14} {\omega_k}^2, \tag{5.21}$$

where $\kappa_{12} = \kappa_8 + \kappa_3 \kappa_9 + \kappa_{10}$, $\kappa_{13} = 2\kappa_1(\kappa_8 + \kappa_{10}) + \kappa_9(\kappa_{11} + \kappa_3 \kappa_1)$, and $\kappa_{14} = \kappa_1^2(\kappa_8 + \kappa_{10}) + \kappa_1 \kappa_9 \kappa_{11}$. We now bound $c(x_k)$ by distinguishing components from $\mathcal{Z}$ and $\mathcal{P}$. We first note that, since the penalty parameters for each subset in $\mathcal{P}$ are bounded away from zero, the test (3.7) is satisfied for all k sufficiently large. Moreover, the remaining components of $c(x_k)$ satisfy the bound

$$\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \kappa_2 \omega_k \mu_{k,j} + \mu_{k,j} \|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\| + \kappa_3 \mu_{k,j} \|x_k - x_*\| \tag{5.22}$$

for all $j \in \mathcal{Z}$ and all k sufficiently large, using (4.9). Hence, using (5.3), (3.7), and (5.22), we deduce that

$$\begin{aligned} \|c(x_k)\| &\leq \sum_{j \in \mathcal{P}} \|c(x_k)_{[\mathcal{Q}_j]}\| + \sum_{j \in \mathcal{Z}} \|c(x_k)_{[\mathcal{Q}_j]}\| \\ &\leq q\eta_k + q\kappa_2 \omega_k \mu_{k,\mathcal{Z}} + \rho_k + q\kappa_3 \mu_{k,\mathcal{Z}} \|\delta_k\|. \end{aligned} \tag{5.23}$$

Note that the first term of the last right-hand side appears only if $\mathcal{P}$ is not empty. Since the algorithm ensures that

$$\omega_k \leq \eta_k \tag{5.24}$$

because $\alpha_\eta < 1$ and $\beta_\eta < 1$, we may obtain from (4.1), (5.23), and (5.19) that

$$\begin{aligned} \left\| \begin{pmatrix} Z^T (\nabla_x \Phi_k - H_*^\ell Y Y^T \delta_k) \\ c(x_k) - J_* Y Y^T \delta_k \end{pmatrix} \right\| &\leq \kappa_{15} \theta_k + \rho_k + q\kappa_{11} \mu_{k,\mathcal{Z}} \theta_k \\ &\quad + q\kappa_3 \mu_{k,\mathcal{Z}} \|Z^T \delta_k\|, \end{aligned} \tag{5.25}$$

where $\kappa_{15} = q + 1 + \kappa_1(\|Z^T H_*^\ell Y\| + \|J_* Y\|)$. By assumption AS6, the coefficient matrix on the left-hand side of (5.17) is nonsingular. Let M be the norm of its inverse. Multiplying both sides of the equation by this inverse and taking norms, we obtain from (5.18), (5.21), (5.24), and (5.25) that

$$\left\| \begin{pmatrix} Z^T\delta_k \\ \bar{\lambda}_k - \lambda_* \end{pmatrix} \right\| \leq M[\kappa_{12}\|Z^T\delta_k\|^2 + \kappa_{13}\|Z^T\delta_k\|\theta_k + \kappa_{14}\theta_k^2 + \kappa_{15}\theta_k + \rho_k + q\kappa_{11}\mu_{k,\mathcal{Z}}\theta_k + q\kappa_3\mu_{k,\mathcal{Z}}\|Z^T\delta_k\|]. \tag{5.26}$$

Suppose now that k is sufficiently large to ensure that

$$\theta_k \leq \frac{1}{4M\kappa_{13}} \tag{5.27}$$

and let

$$\bar{\alpha} \stackrel{\text{def}}{=} \min\left[\alpha_0, \frac{1}{4Mq\kappa_3}\right]. \tag{5.28}$$

Recall that α_0 and hence $\bar{\alpha} < 1$. Then, if $\mu_{k,\mathcal{Z}} \leq \bar{\alpha}$, the relations (5.26)–(5.28) give

$$\|Z^T\delta_k\| \leq \frac{1}{2}\|Z^T\delta_k\| + M[\kappa_{16}\theta_k + \rho_k + \kappa_{12}\|Z^T\delta_k\|^2], \tag{5.29}$$

where $\kappa_{16} = q\kappa_{11} + \kappa_{14} + \kappa_{15}$. Since δ_k, and hence $\|Z^T\delta_k\|$, converge to zero, we have that

$$\|Z^T\delta_k\| \leq \frac{1}{4M\kappa_{12}} \tag{5.30}$$

for k large enough. Hence inequalities (5.29) and (5.30) yield that

$$\|Z^T\delta_k\| \leq 4M(\kappa_{16}\theta_k + \rho_k). \tag{5.31}$$

If $\mathcal{P}$ is empty, we use (5.19), (5.31), and (5.18); the fact that $\mu_{k,\mathcal{Z}} = \alpha_k$; and the inequality

$$\rho_k \leq q\alpha_k\|\lambda_k - \lambda_*\|$$

to deduce (5.4), where $\kappa_4 \stackrel{\text{def}}{=} 4M\kappa_{16} + \kappa_1$ and $\kappa_5 = 4Mq$. Defining $\kappa_6 \stackrel{\text{def}}{=} \kappa_2 + \kappa_3\kappa_4$ and $\kappa_7 \stackrel{\text{def}}{=} \kappa_3\kappa_5$, we deduce (5.5) from (4.7) and (5.4). Now, using (2.1),

$$\|c(x_k)\| \leq \sum_{j=1}^{q} \|c(x_k)_{[\mathcal{Q}_j]}\| = \sum_{j=1}^{q} \mu_{k,j}\|(\bar{\lambda}_k - \lambda_k)_{[\mathcal{Q}_j]}\| \leq q\alpha_k(\|\bar{\lambda}_k - \lambda_*\| + \|\lambda_k - \lambda_*\|) \tag{5.32}$$

and (5.6) then follows from (5.32) and (5.5).

If, on the other hand, $\mathcal{P}$ is not empty, (5.7) results from (4.7), (5.19), (5.31) with $\theta_k = \eta_k$, and (5.24) with $\kappa_6 \stackrel{\text{def}}{=} 4M\kappa_3\kappa_{16} + \kappa_2 + \kappa_3\kappa_1$ and $\kappa_7 \stackrel{\text{def}}{=} 4M\kappa_3$. Finally, (5.8) results from (2.1) and (5.7). □

For the remainder of this section, we will restrict our attention to the case where the sequence of iterates converges to a single limit point. Obviously, this makes AS3

unnecessary. We briefly comment at the end of the section on why this additional assumption cannot be relaxed.

We now show that, if the maximum penalty parameter α_k converges to zero, then the Lagrange multiplier estimates λ_k converge to their true values λ_*.

LEMMA 5.4. *Assume* AS1 *and* AS2 *hold. Assume that* $\{x_k\}$, *the sequence of iterates generated by Algorithm* 3.1, *converges to the single limit point* x_* *at which* AS6 *holds, and with corresponding Lagrange multipliers* λ_*. *Then, if* α_k *tends to zero, the sequence* λ_k *converges to* λ_*.

Proof. Recall that AS6 implies AS4 and therefore that our assumptions are sufficient to apply Theorem 4.6.

We observe that the desired convergence holds if $\lambda_{k,[\mathcal{Q}_j]}$ converges to $\lambda_{*,[\mathcal{Q}_j]}$ for all $j = 1, \ldots, q$. It is thus sufficient to show this latter result for an arbitrary j between 1 and q. The result is obvious if Step 3a is executed infinitely often for the jth subset. Indeed, each time this step is executed, $\lambda_{k+1,[\mathcal{Q}_j]} = \bar{\lambda}_{k,[\mathcal{Q}_j]}$, and inequality (4.7) guarantees that $\bar{\lambda}_{k,[\mathcal{Q}_j]}$ converges to $\lambda_{*,[\mathcal{Q}_j]}$. Suppose, therefore, that Step 3a is not executed infinitely often for this subset. Then $\|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\|$ will remain fixed for all $k \geq k_2$, for some $k_2 > 0$, as Step 3b is executed for each remaining iteration. But (4.9) then implies that $\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \kappa_{17}\mu_{k,j}$ for some constant $\kappa_{17} > 0$ and for all $k \geq k_3 \geq k_2$. As α_k tends to zero and $\alpha_\eta < 1$, $\kappa_{17}\mu_{k,j} \leq \kappa_{17}\alpha_k \leq \alpha_k^{\alpha_\eta} = \eta_k$ for all k sufficiently large for which α_k strictly decreases. But then inequality (3.7) must be satisfied for some $k \geq k_3$, which is impossible, since this would imply that Step 3a is again executed for the jth subset. Hence Step 3a must be executed infinitely often. □

We now consider the behavior of the maximum penalty parameter α_k and show the important result that, under stated assumptions, it is bounded away from zero. The proof of this result is inspired by the technique developed by Conn, Gould, and Toint [5]. When the single penalty parameter definition of the augmented Lagrangian (1.4) is used (or, equivalently, when $q = 1$), one then avoids a steadily increasing ill-conditioning of the Hessian of the augmented Lagrangian. Note that this ill-conditioning is also avoided when $q > 1$, as we show in Theorem 5.6.

THEOREM 5.5. *Assume* AS1 *and* AS2 *hold, and suppose that the sequence of iterates* $\{x_k\}$ *of Algorithm* 3.1 *converges to a single limit point* x_* *with corresponding Lagrange multipliers* λ_*, *at which* AS5 *and* AS6 *hold. Then there is a constant* $\alpha_{\min} \in (0, 1)$ *such that* $\alpha_k = \alpha_{\min}$ *for all* k.

Proof. Suppose otherwise that α_k tends to zero (that is, $\mathcal{P} = \emptyset$) and hence that $\mu_{k,j}$ tends to zero for each j between 1 and q. Then Step 3b must be executed infinitely often for each subset. We aim to obtain a contradiction to this statement by showing that Step 3a is always executed for each subset for sufficiently large k. We note that our assumptions are sufficient to apply Theorem 4.6. Furthermore, we may apply Lemma 5.3 to the complete sequence of iterates.

First observe that

$$\alpha_k \leq \bar{\alpha} < 1 \tag{5.33}$$

for all $k \geq k_1$, where $\bar{\alpha}$ and k_1 are those of Lemma 5.3. Note that

$$\omega_k \leq \alpha_k$$

for all $k \geq k_1$. This follows by definition if (3.12) is executed. Otherwise it is a consequence of the fact that α_k is unchanged while ω_k is reduced when (3.13) occurs.

Let k_4 be the smallest integer k such that

$$\alpha_k^{1-\alpha_\eta} \leq \frac{1}{q(2+\kappa_6)} \tag{5.34}$$

and

$$\alpha_k^{1-\beta_\eta} \leq \min\left[\frac{1}{\kappa_{18}}, \frac{1}{q(2\kappa_{18}+\kappa_6)}\right], \tag{5.35}$$

where $\kappa_{18} = \max(1, \kappa_6 + \kappa_7)$. Note that (5.33) and (5.35) imply that

$$\alpha_k \leq \alpha_k^{1-\beta_\eta} \leq \frac{1}{\kappa_{18}} \leq \frac{1}{\kappa_7} \tag{5.36}$$

for all $k \geq \max(k_1, k_4)$. Furthermore, let k_5 be such that

$$\|\lambda_k - \lambda_*\| \leq 1 \tag{5.37}$$

for all $k \geq k_5$, which is possible because of Lemma 5.4. Now define $k_6 = \max(k_1, k_4, k_5)$, let Γ be the set $\{k \mid (3.12) \text{ is executed at iteration } k-1 \text{ and } k \geq k_6\}$, and let k_0 be the smallest element of Γ. By the assumption that α_k tends to zero, Γ has an infinite number of elements.

By the definition of Γ, for iteration k_0, $\omega_{k_0} = \alpha_{k_0}$ and $\eta_{k_0} = \alpha_{k_0}^{\alpha_\eta}$. Then inequality (5.6) gives that, for each j,

$$\begin{aligned}
\|c(x_{k_0})_{[\mathcal{Q}_j]}\| &\leq \|c(x_{k_0})\| \\
&\leq q(\alpha_{k_0} + \kappa_7\alpha_{k_0}^2)\|\lambda_{k_0} - \lambda_*\| + q\kappa_6\omega_{k_0}\alpha_{k_0} \\
&\leq 2q\alpha_{k_0}\|\lambda_{k_0} - \lambda_*\| + q\kappa_6\omega_{k_0}\alpha_{k_0} && \text{(from (5.36))} \\
&\leq q\alpha_{k_0}(2 + \kappa_6\alpha_{k_0}) && \text{(from (5.37))} \\
&\leq q(2+\kappa_6)\alpha_{k_0} && \text{(from (5.33))} \\
&\leq \alpha_{k_0}^{\alpha_\eta} = \eta_{k_0} && \text{(from (5.34)).}
\end{aligned} \tag{5.38}$$

As a consequence of this inequality, Step 3a will be executed for each j with $\lambda_{k_0+1,[\mathcal{Q}_j]} = \bar{\lambda}(x_{k_0}, \lambda_{k_0,[\mathcal{Q}_j]}, \mu_{k_0,j})_{[\mathcal{Q}_j]}$. Inequality (5.5) together with (5.37) guarantee that

$$\|\lambda_{k_0+1} - \lambda_*\| \leq \kappa_6\omega_{k_0} + \kappa_7\alpha_{k_0}\|\lambda_{k_0} - \lambda_*\| \leq \kappa_{18}\alpha_{k_0}. \tag{5.39}$$

We shall now make use of an inductive proof. Assume that, for each j, Step 3a is executed for iterations $k_0 + i$ $(0 \leq i \leq t)$ and that

$$\|\lambda_{k_0+i+1} - \lambda_*\| \leq \kappa_{18}\alpha_{k_0}^{1+\beta_\eta i}. \tag{5.40}$$

Inequalities (5.38) and (5.39) show that this is true for $t = 0$. We aim to show that the same is true for $i = t+1$. Our assumption that Step 3a is executed gives that, for iteration $k_0 + t + 1$, $\alpha_{k_0+t+1} = \alpha_{k_0}$, $\omega_{k_0+t+1} = \alpha_{k_0}^{t+2}$, and $\eta_{k_0+t+1} = \alpha_{k_0}^{\beta_\eta(t+1)+\alpha_\eta}$. Then inequality (5.6) yields that, for each j,

$$\begin{aligned}
\|c(x_{k_0+t+1})_{[\mathcal{Q}_j]}\| &\leq \|c(x_{k_0+t+1})\| \\
&\leq q(\alpha_{k_0+t+1} + \kappa_7\alpha_{k_0+t+1}^2)\|\lambda_{k_0+t+1} - \lambda_*\| \\
&\quad + q\kappa_6\omega_{k_0+t+1}\alpha_{k_0+t+1} \\
&\leq 2q\alpha_{k_0+t+1}\|\lambda_{k_0+t+1} - \lambda_*\| + q\kappa_6\omega_{k_0+t+1}\alpha_{k_0+t+1} && \text{(from (5.36))} \\
&\leq 2q\kappa_{18}\alpha_{k_0}\alpha_{k_0}^{1+\beta_\eta t} + q\kappa_6\alpha_{k_0}^{t+3} && \text{(from (5.40))} \\
&\leq 2q\kappa_{18}\alpha_{k_0}\alpha_{k_0}^{\alpha_\eta+\beta_\eta t} + q\kappa_6\alpha_{k_0}^{\alpha_\eta+\beta_\eta(t+1)+1} \\
&\leq q(2\kappa_{18}+\kappa_6)\alpha_{k_0}^{1-\beta_\eta}\alpha_{k_0}^{\beta_\eta(t+1)+\alpha_\eta} && \text{(from (5.36))} \\
&\leq \alpha_{k_0}^{\beta_\eta(t+1)+\alpha_\eta} = \eta_{k_0+t+1} && \text{(from (5.35)).}
\end{aligned}$$

Hence Step 3a will again be executed for each j with

$$\lambda_{k_0+t+2,[\mathcal{Q}_j]} = \bar{\lambda}(x_{k_0+t+1}, \lambda_{k_0+t+1,[\mathcal{Q}_j]}, \mu_{k_0+t+1,j})_{[\mathcal{Q}_j]}.$$

Inequality (5.5) then implies that

$$\begin{aligned}
\|\lambda_{k_0+t+2} - \lambda_*\| &\leq \kappa_6\omega_{k_0+t+1} + \kappa_7\alpha_{k_0+t+1}\|\lambda_{k_0+t+1} - \lambda_*\| \\
&\leq \kappa_6\alpha_{k_0}^{t+2} + \kappa_7\kappa_{18}\alpha_{k_0}\alpha_{k_0}^{1+\beta_\eta t} && \text{(from (5.40))} \\
&\leq \kappa_6\alpha_{k_0}^{1+\beta_\eta(t+1)} + \kappa_7\kappa_{18}\alpha_{k_0}\alpha_{k_0}^{1+\beta_\eta t} \\
&= (\kappa_6 + \kappa_7\kappa_{18}\alpha_{k_0}^{1-\beta_\eta})\alpha_{k_0}^{1+\beta_\eta(t+1)} \\
&\leq (\kappa_6 + \kappa_7)\alpha_{k_0}^{1+\beta_\eta(t+1)} && \text{(from (5.35))} \\
&\leq \kappa_{18}\alpha_{k_0}^{1+\beta_\eta(t+1)},
\end{aligned}$$

which establishes (5.40) for $i = t + 1$. Thus Step 3a is executed for each $j = 1, \ldots, q$ for all iterations $k \geq k_0$. But this implies that Γ is finite, which contradicts the assumption that Step 3b is executed infinitely often for each subset. Hence the theorem is proved. □

This theorem was all that was needed in Conn, Gould, and Toint [5]. However, the situation is more complex here because q may be larger than one. If the ill-conditioning of the Hessian is to be avoided, we must now prove the stronger result that *all* penalty parameters stay bounded away from zero.

THEOREM 5.6. *Assume* AS1 *and* AS2 *hold, and suppose that the sequence of iterates* $\{x_k\}$ *of Algorithm* 3.1 *converges to a single limit point* x_* *with corresponding Lagrange multipliers* λ_*, *at which* AS5 *and* AS6 *hold. Then there is a constant* $\mu > 0$ *such that* $\mu_{k,j} \geq \mu$ *for all* k *and all* $j = 1, \ldots, q$.

Proof. Assume otherwise that $\mathcal{Z}$ is not empty and hence that $\mu_{k,\mathcal{Z}}$ converges to zero. Then Step 3b must be executed infinitely often for $j \in \mathcal{Z}$. We aim to obtain a contradiction to this statement by showing that, for any $j \in \mathcal{Z}$, Step 3a is always executed for sufficiently large k. We may deduce from Theorem 5.5 that α_k attains its minimum value $\alpha_{\min} \in (0, 1)$ at iteration $k_{\max}$, say. Hence, $\mathcal{P} \neq \emptyset$. Furthermore, we may apply Lemma 5.3 to the complete sequence of iterates. Let $k_7 \geq k_{\max}$ be the smallest integer for which

$$\mu_{k,\mathcal{Z}} \leq \min\left[\bar{\alpha}, \frac{1}{2\kappa_6 + \kappa_7}, \frac{\alpha_{\min}^{\beta_\eta+\epsilon} - \alpha_{\min}}{q(2\kappa_6 + \kappa_7)}\right] \tag{5.41}$$

for all $k \geq k_7 \geq k_1$, where $\bar{\alpha}$ and k_1 are those of Lemma 5.3 and where $\epsilon = \frac{1}{2}(1 - \beta_\eta)$. Note that $\alpha_{\min}^{\beta_\eta+\epsilon} > \alpha_{\min}$ as $\beta_\eta < 1$.

Consider the jth subset for some $j \in \mathcal{Z}$. At iteration $k \geq k_7$, the algorithm ensures that

$$\mu_{k+1,j}\|(\lambda_{k+1} - \lambda_*)_{[\mathcal{Q}_j]}\| \leq \alpha_{\min}\mu_{k,j}\|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\|$$

if Step 3b is executed for the jth subset, while (5.7) ensures that

$$\mu_{k+1,j}\|(\lambda_{k+1} - \lambda_*)_{[\mathcal{Q}_j]}\| \leq \mu_{k,j}(\kappa_6\eta_k + \kappa_7\rho_k)$$

if Step 3a is executed for the same subset. Summing on all $j \in \mathcal{Z}$ and defining

$$\begin{aligned}
\mathcal{Z}_{k,a} &= \{j \in \mathcal{Z} \mid \text{Step 3a is executed for the } j\text{th subset at iteration } k\}, \\
\mathcal{Z}_{k,b} &= \{j \in \mathcal{Z} \mid \text{Step 3b is executed for the } j\text{th subset at iteration } k\},
\end{aligned}$$

we obtain that

$$\begin{aligned} \rho_{k+1} &\leq \alpha_{\min} \sum_{j \in \mathcal{Z}_{k,b}} \mu_{k,j} \|(\lambda_k - \lambda_*)_{[\mathcal{Q}_j]}\| + \sum_{j \in \mathcal{Z}_{k,a}} \mu_{k,j}(\kappa_6 \eta_k + \kappa_7 \rho_k) \\ &\leq (\alpha_{\min} + \kappa_7 q \mu_{k,\mathcal{Z}})\rho_k + \kappa_6 q \mu_{k,\mathcal{Z}} \eta_k. \end{aligned} \tag{5.42}$$

For the purpose of obtaining a contradiction, assume now that

$$\rho_k \geq \frac{1}{2}\eta_k \tag{5.43}$$

for all $k \geq k_7$. Then (5.42) gives that, for all $k \geq k_7$,

$$\frac{\rho_{k+1}}{\rho_k} \leq \alpha_{\min} + \kappa_7 q \mu_{k,\mathcal{Z}} + 2\kappa_6 q \mu_{k,\mathcal{Z}} \leq \alpha_{\min}^{\beta_\eta + \epsilon} < 1 \tag{5.44}$$

because of (5.41). Hence we obtain from (5.44) that

$$\rho_{k+1} \leq \rho_{k_7} \alpha_{\min}^{(k-k_7+1)(\beta_\eta+\epsilon)}.$$

Therefore, since $\rho_{k_7}\alpha_{\min}^{(k-k_7+1)\epsilon}$ tends to zero, we obtain that

$$\rho_{k+1} < \frac{1}{2}\alpha_{\min}^{\alpha_\eta + (k_7 - k_{\max})\beta_\eta} \alpha_{\min}^{(k-k_7+1)\beta_\eta} = \frac{1}{2}\alpha_{\min}^{\alpha_\eta + (k-k_{\max}+1)\beta_\eta} = \frac{1}{2}\eta_{k+1}$$

for all sufficiently large k, where the last equality results from the definition of $k_{\max}$ and (3.13). But this contradicts (5.43), which implies that (5.43) does not hold for all k sufficiently large. As a consequence, there exists a subsequence $\mathcal{K}$ such that

$$\rho_k < \frac{1}{2}\eta_k \tag{5.45}$$

for all $k \in \mathcal{K}$. Consider such a k. Then, using (5.42) and (5.45), we deduce that

$$\rho_{k+1} < \frac{1}{2}\eta_k(\alpha_{\min} + q\kappa_7 \mu_{k,\mathcal{Z}} + 2q\kappa_6 \mu_{k,\mathcal{Z}}) \leq \frac{1}{2}\alpha_{\min}^{\beta_\eta+\epsilon}\eta_k \leq \frac{1}{2}\eta_{k+1},$$

where we have used (5.41) to obtain the second inequality. As a consequence, $k+1 \in \mathcal{K}$ and (5.45) holds for all k sufficiently large. Returning to the subset $j \in \mathcal{Z}$, we now obtain from (5.8) and (5.45) that

$$\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \eta_k \left(\kappa_6 \mu_{k,\mathcal{Z}} + \frac{1}{2}(1 + \kappa_7 \mu_{k,\mathcal{Z}})\right) \leq \eta_k$$

for all k sufficiently large because of (5.41). Hence Step 3a is executed for the subset j and for all sufficiently large k, which implies that j does not belong to $\mathcal{Z}$. Therefore $\mathcal{Z}$ is empty and the proof of the theorem is completed. □

As in Conn, Gould, and Toint [5], we examine the rate of convergence of our algorithms.

THEOREM 5.7. *Under the assumptions of Theorem* 5.6, *the iterates* x_k *and the Lagrange multipliers* $\bar{\lambda}_k$ *of Algorithm* 3.1 *are at least R-linearly convergent with R-factor at most* $\alpha_{\min}^{\beta_\eta}$, *where* $\alpha_{\min}$ *is the smallest value of the maximum penalty parameter generated by the algorithm.*

Proof. The proof parallels that of Lemma 5.3. First, Theorem 5.5 shows that the maximum penalty parameter α_k stays bounded away from zero and thus remains fixed at some value $\alpha_{\min} > 0$, for $k \geq k_{\max}$. For all subsequent iterations,

$$\omega_{k+1} = \alpha_{\min}\omega_k \quad \text{and} \quad \eta_{k+1} = \alpha_{\min}^{\beta_\eta}\eta_k \tag{5.46}$$

hold. Moreover, Theorem 5.6 implies that, for all $j = 1, \ldots, q$, (3.7) holds for all $k \geq k_{\max}$ sufficiently large. Hence because of (4.1), the bound on the right-hand side of (5.25) may be replaced by $\kappa_{15}\omega_k + q\eta_k$, and thus

$$\|Z^T\delta_k\| \leq M[\kappa_{15}\omega_k + q\eta_k + \kappa_{12}\|Z^T\delta_k\|^2 + \kappa_{13}\|Z^T\delta_k\|\omega_k + \kappa_{14}\omega_k^2]. \tag{5.47}$$

Therefore, if k is sufficiently large so that

$$\omega_k \leq \frac{1}{2M\kappa_{13}} \tag{5.48}$$

and

$$\|Z^T\delta_k\| \leq \frac{1}{4M\kappa_{12}}, \tag{5.49}$$

inequalities (5.47)–(5.49) can be rearranged to yield

$$\|Z^T\delta_k\| \leq 4M((\kappa_{14} + \kappa_{15})\omega_k + q\eta_k).$$

But then (5.19) gives that

$$\|\delta_k\| \leq \kappa_{19}\omega_k + \kappa_{20}\eta_k, \tag{5.50}$$

where $\kappa_{19} = \kappa_1 + 4M(\kappa_{14} + \kappa_{15})$ and $\kappa_{20} = 4Mq$. Since $\beta_\eta < 1$ and $\alpha_{\min} < 1$, (5.46) and (5.50) show that x_k converges to x_* at least R-linearly, with R-factor $\alpha_{\min}^{\beta_\eta}$. Inequalities (4.7) and (5.50) then guarantee the same property for $\bar{\lambda}_k$. □

To conclude this section, we note that the conclusions of Theorems 5.5, 5.6, and 5.7 require that the complete sequence of iterates converges to a unique limit point. As indicated above, this assumption cannot be relaxed. The counterexample presented by Conn, Gould, and Toint [5] (where the linear inequality constraints are simple bound constraints on the problem's variables) shows that the sequence of penalty parameters may indeed converge to zero if there is more than a single limit point.

6. Second-order conditions. If we further strengthen the stopping test for the inner iteration beyond (3.3) to include second-order conditions, we can then guarantee that our algorithms converge to an isolated local solution. More specifically, we require the following additional assumption.

AS7. Suppose that x_k satisfies (3.3), converges to x_* for $k \in \mathcal{K}$, such that Z_* has a rank strictly greater than m. Then, if Z is defined as in AS6, we assume that $Z^T\nabla_{xx}\Phi_k Z$ is uniformly positive definite (that is, its smallest eigenvalue is uniformly bounded away from zero) for all $k \in \mathcal{K}$ sufficiently large.

We can then prove the following result.

THEOREM 6.1. *Under assumptions* AS1–AS3, AS5–AS7, *the iterates* x_k, $k \in \mathcal{K}$, *generated by Algorithm* 3.1 *converge to an isolated local solution of* (1.1)–(1.3).

Proof. By the definition of Φ,

$$\nabla_{xx}\Phi_k = H^\ell(x_k, \bar{\lambda}_k) + \sum_{j=1}^{q} \frac{1}{\mu_{k,j}} J_{\mathcal{Q}_j}(x_k)^T J_{\mathcal{Q}_j}(x_k), \tag{6.1}$$

where $J_{\mathcal{Q}_j}(x)$ is the Jacobian of $c(x)_{[\mathcal{Q}_j]}$. Note that the rank of Z is at least that of Z_*. AS7 then implies that there exists a nonzero vector s such that

$$J(x_k)Zs = 0$$

and hence

$$J_{\mathcal{Q}_j}(x_k)Zs = 0 \tag{6.2}$$

for each j. For any such vector, AS7 further implies that

$$s^T Z^T \nabla_{xx}\Phi_k Zs \geq \kappa_{21}\|s\|^2$$

for some $\kappa_{21} > 0$, which in turn gives that

$$s^T Z^T H^\ell(x_k, \bar{\lambda}_k)Zs \geq \kappa_{21}\|s\|^2$$

because of (6.1) and (6.2). By continuity of H^ℓ as x_k and $\bar{\lambda}_k$ approach their limits, this ensures that

$$s^T Z^T H^\ell(x_*, \lambda_*)Zs \geq \kappa_{21}\|s\|^2$$

for all nonzero s satisfying

$$J(x_*)Zs = 0,$$

which implies that x_* is an isolated local solution of (1.1)–(1.3) (see, for instance, Avriel [2, Thm. 3.11]). □

If we assume that the inner iteration stopping test is tightened so that $\nabla_{xx}\Phi_k$ is required to be uniformly positive definite in the null space of the dominant constraints, and if we assume that the nondegeneracy condition (5.2) holds, then Corollary 5.2 ensures that $Z_k = Z = Z_*$ for sufficiently large k and Theorem 6.1 holds. A weaker version of this result also holds, where only positive semidefiniteness of the augmented Lagrangian's Hessian is required, yielding then that x_* is a (possibly not isolated) minimizer of the problem.

7. Extensions.

7.1. Flexible Lagrange multiplier updates. The formula (2.1) has definite advantages for large-scale computations but may otherwise appear unduly restrictive. The purpose of the first extension we consider is to introduce more freedom in our algorithmic framework by replacing this formula by a more general condition, allowing a much larger class of Lagrange multiplier updates to be used. More specifically, we consider modifying Algorithm 3.1 as follows.

ALGORITHM 7.1 This algorithm is identical to Algorithm 3.1, except that Step 3 is replaced by the following, where γ is a constant in $(0, 1)$.

Step 3 [Disaggregated updates]. Compute a new vector of Lagrange multiplier estimates $\hat{\lambda}_{k+1}$. For $j = 1, \ldots, q$, execute Step 3a if

$$\|c(x_k)_{[\mathcal{Q}_j]}\| \leq \eta_k$$

or Step 3b otherwise.

Step 3a [Update Lagrange multiplier estimates]. Set

$$\begin{aligned} \lambda_{k+1,[\mathcal{Q}_j]} &= \begin{cases} \hat{\lambda}_{k+1,[\mathcal{Q}_j]} & \text{if } \|\hat{\lambda}_{k+1,[\mathcal{Q}_j]}\| \leq \mu_{k+1,j}^{-\gamma}, \\ \lambda_{k,[\mathcal{Q}_j]} & \text{otherwise,} \end{cases} \\ \mu_{k+1,j} &= \mu_{k,j}. \end{aligned}$$

Step 3b [Reduce the penalty parameter]. Set

$$\begin{aligned} \lambda_{k+1,[\mathcal{Q}_j]} &= \begin{cases} \hat{\lambda}_{k+1,[\mathcal{Q}_j]} & \text{if } \|\hat{\lambda}_{k+1,[\mathcal{Q}_j]}\| \leq \mu_{k+1,j}^{-\gamma}, \\ \lambda_{k,[\mathcal{Q}_j]} & \text{otherwise,} \end{cases} \\ \mu_{k+1,j} &= \tau_{k,j}\mu_{k,j}, \end{aligned}$$

where $\tau_{k,j}$ is defined by (3.9) in Algorithm 3.1.

Algorithm 7.1 allows a more flexible choice of the multipliers than Algorithm 3.1 but requires that some control is enforced to prevent their growth at an unacceptably fast rate. It covers, among others, the choice of the least-squares estimates $\lambda(x)$ as defined in (2.5).

The global convergence theory presented in §4 for Algorithm 3.1 can be extended to cover Algorithm 7.1. This extension is detailed in Conn et al. [9]. Conn et al. [10] extend the local convergence analysis of §5 to Algorithm 7.1, under the additional condition that

$$\|\hat{\lambda}_{k+1} - \lambda_*\| \leq \kappa_{22}\|x_k - x_*\| + \kappa_{23}\omega_k$$

holds for some positive constants κ_{22} and κ_{23} and all $k \in \mathcal{K}$ sufficiently large, where $\mathcal{K}$ is the index set of a subsequence of iterates (generated by Algorithm 7.1) converging to the critical point x_* with corresponding Lagrange multipliers λ_*. Both (2.1) and (2.5) satisfy this condition because of Theorem 4.6.

We also note that Corollary 5.2 ensures that the least-squares multiplier estimates (2.5) are implementable when the nondegeneracy condition (5.2) holds. By this we mean that the estimates

$$\hat{\lambda}_k = -((J_k Z_k)^+)^T Z_k^T g_k$$

are identical to those defined in (2.5) for all k sufficiently large and, unlike (2.5), are well defined when x_* is unknown.

7.2. Alternative criticality measures. In Algorithms 3.1 and 7.1 we used the criticality measure $\|P_{T_k}(-\nabla_x \Phi_k)\|$ to define the stopping criterion of the inner iteration (see (3.3)) because it is general. However, this quantity might not be easily computed in the course of the numerical method used to calculate x_k, especially when the dimension of the problem is high. It is therefore of interest to examine other criticality measures that might be easier to calculate. It is the purpose of this section to analyze such alternative proposals.

Given D_k, N_k, and A_{D_k} as above, we first claim that (3.3) can be replaced by the requirement that there exists a set of nonpositive "dominant multipliers" $\{\xi_{ik}\}_{i \in M_k}$ ($M_k \subseteq D_k$, $\xi_{ik} \leq 0$) such that

$$\|\nabla_x \Phi_k + A_{D_k}^T \xi_k\| \leq \omega_k, \tag{7.1}$$

where ξ_k is the $|D_k|$-dimensional vector whose ith component is ξ_{ik} if $i \in M_k$ or zero otherwise. We prove this claim.

LEMMA 7.1. *Assume that there exists a nonpositive* ξ_k *such that* (7.1) *holds at* x_k. *Then* (3.3) *also holds at* x_k.

Proof. Since the vector $A_{D_k}^T \xi_k$ belongs, by construction, to the cone N_k defined in (3.4), we can immediately deduce from the definition of the othogonal projection and (7.1) that

$$\|P_{T_k}(-\nabla_x \Phi_k)\| = \| - \nabla_x \Phi_k - P_{N_k}(-\nabla_x \Phi_k)\| \leq \| - \nabla_x \Phi_k - A_{D_k}^T \xi_k\| \leq \omega_k,$$

which is the desired inequality. □

Condition (7.1) is appealing for two reasons. First, a set of (possibly approximate) multipliers is available in many numerical procedures that might be used to perform the inner iteration and to compute a suitable x_k; one can then select those multipliers that correspond to the dominant constraints, further restrict this choice to the nonpositive ones, and finally check (7.1). Such a scheme is implicitly used by both the Harwell [17] barrier-function quadratic programming codes `VE14` and `VE19` and the IMSL [19] general linearly constrained minimization package `LCONG`.

Alternatively, suitable multipliers can be computed, for instance, by (approximately) solving the least-squares problem

$$\min_{\xi} \|\nabla_x \Phi_k + A_{D_k}^T \xi\|$$

and selecting the nonpositive components of the resulting vector ξ or by (approximately) solving the constrained least-squares problem

$$\min_{\xi \leq 0} \|\nabla_x \Phi_k + A_{D_k}^T \xi\|. \tag{7.2}$$

Condition (7.1) is also appealing because it provides, in a single condition, both a stopping condition on the inner iteration and a measure of the tolerated "inexactness" in solving the associated least-squares problem, if this is the procedure chosen to obtain the dominant multipliers.

We may therefore deduce from Lemma 7.1 that the convergence theory holds for Algorithms 3.1 and 7.1 whenever (7.1) is used instead of (3.3).

Condition (7.1) can be further specialized. For instance, one might choose to impose the familiar "reduced gradient" criterion

$$\|\hat{Z}(x_k)^T \nabla_x \Phi_k\| \leq \omega_k,$$

where $\hat{Z}(x_k)$ is an orthogonal matrix whose columns span the null space of the constraints active at x_k, provided that the multipliers associated with these linear constraints are all nonpositive. In this case, we have that

$$\|P_{T_k}(-\nabla_x \Phi_k)\| \leq \|P_{T(x_k,0)}(-\nabla_x \Phi_k)\| = \|\hat{Z}(x_k)^T \nabla_x \Phi_k\| \leq \omega_k \tag{7.3}$$

because $T(x_k, 0)$, the tangent cone to the set determined by the linear inequality constraints active at x_k, contains T_k. As a consequence, the convergence theory still holds when this criterion, which has been implemented by several subroutines for minimizing a general objective function subject to linear constraints (for example, the NAG [22], quadratic programming code `E04NFF` and the more general package `E04UCF`), is used as an inner-iteration stopping rule within Algorithms 3.1 and 7.1. This is also true for reduced gradient methods (e.g., MINOS [21] or LSNNO [24]),

which compute a full column rank matrix $\check{Z}(x_k)$ whose columns are generally non-orthonormal but depend upon a subset of the (finite number) of coefficients for the linear constraints. Indeed, the norm of $\check{Z}(x_k)$ is then bounded above and away from zero, and a relationship that is a weighted form of (7.3) thus also holds in these cases.

To preserve coherence with the framework presented in Conn et al. [8], we finally note that σ_k as defined in (4.3) may also be viewed as a criticality measure. Hence we might decide to stop the inner iteration when

$$\sigma_k \leq \omega_k. \tag{7.4}$$

The reader is referred to Conn et al. [9] for a proof that global convergence is still obtained for this modification of Algorithms 3.1 and 7.1. However, the authors have not been able to prove the desired local convergence properties with only (7.4). Instead, the local convergence theory is covered for Algorithms 3.1 and 7.1 for the stronger condition

$$\sigma_k \leq \omega_k^2 \tag{7.5}$$

(see Conn et al. [10] for details). This condition is theoretically interesting but might be practically too strong. Note, as we now show, that it implies a variant of (3.3).

THEOREM 7.2. *Assume that* $\{x_k\}$, $k \in \mathcal{K}$, *is a convergent subsequence of vectors of* $\mathcal{B}$ *such that* (7.5) *holds for each* $k \in \mathcal{K}$, *where the* ω_k *converge to zero as* k *increases in* $\mathcal{K}$. *Then the inequality*

$$\|P_{T_k}(-\nabla_x \Phi_k)\| \leq \kappa_{24}\omega_k \tag{7.6}$$

also holds for each $k \in \mathcal{K}$ *sufficiently large and for some* $\kappa_{24} \geq 1$.

Proof. We first consider the simple case where $p = 0$, that is, when no linear inequality is present. In this case, it is easy to check from (4.3) that $\sigma_k = \|\nabla_x \Phi_k\|$. But we must have that $D_k = \emptyset$. Thus $\sigma_k = \|P_{T_k}(-\nabla_x \Phi_k)\|$. We therefore obtain that (7.6) holds with $\kappa_{24} = 1$ and k large enough to ensure that $\omega_k \leq 1$.

Assume now that $p > 0$. The Moreau decomposition of $-\nabla_x \Phi_k$ [20] is given by

$$-\nabla_x \Phi_k = P_{T_k}(-\nabla_x \Phi_k) + P_{N_k}(-\nabla_x \Phi_k).$$

If $P_{T_k}(-\nabla_x \Phi_k)$ is zero, then (3.3) obviously holds for any choice of κ_{24}. Assume therefore that $P_{T_k}(-\nabla_x \Phi_k)$ is nonzero. We now show that $x_k + d_k \in \mathcal{B}$, where we define

$$d_k \stackrel{\text{def}}{=} \epsilon_k \frac{P_{T_k}(-\nabla_x \Phi_k)}{\|P_{T_k}(-\nabla_x \Phi_k)\|}, \quad \text{with } \epsilon_k \stackrel{\text{def}}{=} \min\left[1, \frac{\kappa_0 \omega_k}{\|A\|_\infty}\right]. \tag{7.7}$$

Assume first that $i \in D_k$. Then $-a_i \in N_k$ and $a_i^T d_k \geq 0$ because of the polarity of N_k and T_k. Since $x_k \in \mathcal{B}$, we obtain that

$$a_i^T(x_k + d_k) - b_i = (a_i^T x_k - b_i) + a_i^T d_k \geq 0. \tag{7.8}$$

On the other hand, if $i \notin D_k$, we have that $a_i^T x_k - b_i > \kappa_0 \omega_k$ and hence

$$(a_i^T x_k - b_i) + a_i^T d_k > \kappa_0 \omega_k - \|a_i\|.\|d_k\| = \kappa_0 \omega_k - \epsilon_k \|a_i\| \geq \kappa_0 \omega_k - \kappa_0 \omega_k = 0. \tag{7.9}$$

Gathering (7.8) and (7.9), we obtain that $x_k + d_k \in \mathcal{B}$, as desired. Furthermore, since $\|d_k\| \leq 1$ by definition, we have verified that d_k is feasible for the minimization

problem (4.3) associated with the definition of σ_k. Hence,

$$\begin{aligned} \sigma_k &\geq -\nabla_x \Phi_k{}^T d_k \\ &= P_{T_k}(-\nabla_x \Phi_k)^T d_k + P_{N_k}(-\nabla_x \Phi_k)^T d_k \\ &= \|P_{T_k}(-\nabla_x \Phi_k)\| \, \|d_k\| \\ &= \epsilon_k \|P_{T_k}(-\nabla_x \Phi_k)\|, \end{aligned} \tag{7.10}$$

where we have used successively the Moreau decomposition of $-\nabla_x \Phi_k$, the definition of d_k, and the orthogonality of the terms in the Moreau decomposition. If $\epsilon_k = 1$, then (7.5) and (7.10) imply that

$$\|P_{T_k}(-\nabla_x \Phi_k)\| \leq \omega_k{}^2 \leq \omega_k \tag{7.11}$$

for $k \in \mathcal{K}$ sufficiently large. Otherwise, we deduce from (7.10), (7.5), and (7.7) that

$$\|P_{T_k}(-\nabla_x \Phi_k)\| \leq \frac{\|A\|_\infty}{\kappa_0} \omega_k. \tag{7.12}$$

As a consequence of (7.11) and (7.12), we therefore obtain that (7.6) holds with

$$\kappa_{24} = \max\left[1, \frac{\|A\|_\infty}{\kappa_0}\right].$$

Combining all cases, we conclude that (7.6) holds with this last value of κ_{24}. □

We finally note that Lemma 7.1 and Theorem 7.2 do not depend on the actual form of the augmented Lagrangian (1.5) but are valid independently of the function minimized in the inner iteration. This observation could be useful if alternative techniques for augmenting the Lagrangian are considered for a merit function.

8. Conclusion. We have considered a class of augmented Lagrangian algorithms for constrained nonlinear optimization, where the linear constraints present in the problem are handled directly and where multiple penalty parameters are allowed. The algorithms in this class have the advantage that efficient techniques for handling linear constraints may be used at the inner iteration level and also that the sparsity pattern of the Hessian of the augmented Lagrangian is independent of that of the linear constraints. The global and local convergence results available for the specific case where linear constraints reduce to simple bounds have been extended to the more general and useful context where any form of linear constraint is permitted.

We finally note that the theory presented is directly relevant to practical computation since the inner iteration stopping rule (3.3) covers the type of optimality tests used in available packages for linearly constrained problems. This means that these packages can be applied to obtain an (approximate) solution of the subproblem and constitutes a realistic and attractive algorithmic development.

It is now the authors' intention to perform extensive numerical experiments on large-scale problems. This development requires considerable care and sophistication if an efficient solver for the subproblem is to be integrated with the class of algorithms described here.

Acknowledgments. The authors wish to acknowledge funding provided by a NATO travel grant. They are also grateful to J. Nocedal and the anonymous referees for their constructive comments.

REFERENCES

[1] M. ARIOLI, T. F. CHAN, I. S. DUFF, N. I. M. GOULD, AND J. K. REID, *Computing a search direction for large-scale linearly constrained nonlinear optimization calculations*, Tech. report TR/PA/93/34, CERFACS, Toulouse, France, 1993.

[2] M. AVRIEL, *Nonlinear Programming: Analysis and Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1976.

[3] D. P. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, London, 1982.

[4] J. V. BURKE, J. J. MORÉ, AND G. TORALDO, *Convergence properties of trust region methods for linear and convex constraints*, Math. Programming, 47 (1990), pp. 305–336.

[5] A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[6] ———, LANCELOT: *A Fortran package for large-scale nonlinear optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer-Verlag, Heidelberg, Berlin, and New York, 1992.

[7] ———, *On the number of inner iterations per outer iteration of a globally convergent algorithm for optimization with general nonlinear equality constraints and simple bounds*, in Proc. 14th Biannual Numerical Analysis Conference Dundee, 1991, D. F. Griffiths and G. A. Watson, eds., Longman Sci. Tech., Harlow, 1992, pp. 49–68.

[8] A. R. CONN, NICK GOULD, A. SARTENAER, AND PH. L. TOINT, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[9] ———, *Global Convergence of Two Augmented Lagrangian Algorithms for Optimization with a Combination of General Equality and Linear Constraints*, Tech. report 93/01, Department of Mathematics, FUNDP, Namur, Belgium, 1993.

[10] ———, *Local Convergence Properties of Two Augmented Lagrangian Algorithms for Optimization with a Combination of General Equality and Linear Constraints*, Tech. report 93/20, Department of Mathematics, FUNDP, Namur, Belgium, 1993.

[11] J. C. DUNN, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.

[12] A. V. FIACCO, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, New York, 1983.

[13] R. FLETCHER, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, 1987.

[14] A. L. FORSGREN AND W. MURRAY, *Newton methods for large-scale linear equality-constrained minimization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 560–587.

[15] N. I. M. GOULD, *On the convergence of a sequential penalty function method for constrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 107–128.

[16] W. A. GRUVER AND E. SACHS, *Algorithmic Methods in Optimal Control*, Pitman, Boston, MA, 1980.

[17] HARWELL SUBROUTINE LIBRARY, *A catalogue of subroutines (release* 11), Advanced Computing Department, Harwell Laboratory, Harwell, UK, 1993.

[18] M. R. HESTENES, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.

[19] IMSL, *MATH/LIBRARY User's Manual, Vols.* 1–3, IMSL Inc., Houston, TX, 1987.

[20] J. J. MOREAU, *Décomposition orthogonale d'un espace hilbertien selon deux cônes mutuellement polaires*, C.R. Acad. Sci. Paris, 255 (1962), pp. 238–240.

[21] B. A. MURTAGH AND M. A. SAUNDERS, *Large-scale linearly constrained optimization*, Math. Programming, 14 (1978), pp. 41–72.

[22] NAG, *Fortran Manual, Mark* 16, NAG Ltd., Oxford, UK, 1993.

[23] M. J. D. POWELL, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, London and New York, 1969,

[24] PH. L. TOINT AND D. TUYTTENS, *LSNNO: A Fortran subroutine for solving large scale nonlinear network optimization problems*, ACM Trans. Math. Software, 18 (1992), pp. 308–328.

[25] R. J. VANDERBEI AND T. J. CARPENTER, *Indefinite systems for interior point methods*, Math. Programming, 58 (1993), pp. 1–32.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 704–713, August 1996
© 1996 Society for Industrial and Applied Mathematics
010

A NEW MERIT FUNCTION AND A SUCCESSIVE QUADRATIC PROGRAMMING ALGORITHM FOR VARIATIONAL INEQUALITY PROBLEMS*

KOUICHI TAJI† AND MASAO FUKUSHIMA‡

Abstract. Recently, various merit functions for variational inequality problems have been proposed and their properties have been studied. Unfortunately, these functions may not be easy to evaluate unless the constraints of the problem have a relatively simple structure. In this paper, a new merit function for variational inequality problems with general convex constraints is proposed. At each point, the proposed function is defined as an optimal value of a quadratic programming problem whose constraints consist of a linear approximation of the given nonlinear constraints. It is shown that the set of constrained minima of the proposed merit function coincides with the set of solutions to the original variational inequality problem. It is also shown that this function is directionally differentiable in all directions and, under suitable assumptions, any stationary point of the function over the constraint set actually solves the original variational inequality problem. Finally, a descent method for solving the variational inequality problem is proposed and its convergence is proved. The method is closely related to a successive quadratic programming method for solving nonlinear programming problems.

Key words. variational inequality problem, merit function, successive quadratic programming, global convergence

AMS subject classifications. 49J40, 49M40, 90C30

1. Introduction.

Let S be a nonempty closed convex subset of R^n and let F be a continuous mapping from R^n into R^n. The variational inequality (VI) problem is to find a vector $x^* \in S$ such that

$$\langle F(x^*), x - x^* \rangle \geq 0 \quad \text{for all} \quad x \in S, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in R^n. This problem has been used to formulate and study various equilibrium models arising in economics, operations research, transportation, and regional science [13].

To solve the VI problem (1), many iterative methods, such as the Newton method, projection methods, the linearized Jacobi method, and the successive overrelaxation methods, have been proposed and their convergence properties have been examined [7].

On the other hand, various merit functions for the VI problem have been proposed and their properties have been studied [1, 2, 5, 10, 14]. Among them, the gap function

$$g(x) = \max_y \{\langle F(x), x - y \rangle \mid y \in S\}, \tag{2}$$

first introduced by Auslender [2], has the property that its minimum on S coincides with a solution to the VI problem, and hence problem (1) can be reformulated as the optimization problem

$$\text{minimize } g(x) \quad \text{subject to } x \in S. \tag{3}$$

* Received by the editors July 11, 1994; accepted for publication (in revised form) January 17, 1995.

† Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-01, Japan (taji@is.aist-nara.ac.jp).

‡ Department of Applied Mathematics and Physics, Graduate School of Engineering, Kyoto University, Kyoto 606-01, Japan (fuku@kuamp.kyoto-u.ac.jp).

Based on this formulation, Marcotte [11] proposed a descent algorithm for monotone variational inequalities, and Marcotte and Dussault [12] presented a globally convergent modification of the Newton method.

Fukushima [5] has proposed another optimization formulation of the VI problem, using the regularized gap function

$$\hat{f}(x) = \max_y \left\{ -\langle F(x), y - x\rangle - \frac{1}{2}\langle y - x, G(y - x)\rangle \;\middle|\; y \in S \right\}, \tag{4}$$

where G is an $n \times n$ symmetric positive definite matrix. The function is shown to be differentiable whenever F is, while the gap function g defined by (2) is generally nondifferentiable. Fukushima [5] has proposed a descent method for solving the VI problem using the merit function $\hat{f}$. The merit function $\hat{f}$ has also been used in a globally convergent modification of the Newton method by Taji, Fukushima, and Ibaraki [15]. Independently, Auchmuty [1] has proposed a class of merit functions which includes the gap function g and Fukushima's merit function. Larsson and Patriksson [10] have developed and generalized Auchmuty's class of merit functions. Wu, Florian, and Marcotte [16] have proposed a general descent framework for the VI problem by using a class of gap functions.

In this paper, we propose a new merit function, which is defined by (4) with the set S replaced by its polyhedral outer approximation. This function has the advantage over the function $\hat{f}$, particularly when S is a general convex set specified by nonlinear convex inequalities so that $\hat{f}$ is difficult to evaluate. We show that the proposed merit function has the property that its minimum on S is a solution to (1). We also show that this function is directionally differentiable in all directions and, under suitable assumptions, any stationary point of the function over the constraint set actually solves the original VI problem. Finally, we propose a descent method for solving the VI problem and establish its convergence. We note that the method is closely related to a successive quadratic programming method for solving nonlinear programming problems.

2. A new merit function. In this section, we introduce a new merit function for the VI problem which is a relaxed version of the merit function introduced by Fukushima [5]. In the remainder of this paper, we suppose that the set S consists of points $x \in R^n$ satisfying

$$c_i(x) \le 0, \quad i = 1, 2, \ldots, m, \tag{5}$$

where $c_i : R^n \to R$ are twice continuously differentiable convex functions, and that Slater's constraint qualification holds, i.e., there exists an $\hat{x} \in R^n$ such that

$$c_i(\hat{x}) < 0 \;\text{ for all }\; i = 1, 2, \ldots, m. \tag{6}$$

We also assume that the mapping F is continuously differentiable. Under these assumptions, it is known that x^* is a solution to (1) if and only if there exist Lagrange multipliers $\lambda_i^*, i = 1, \ldots, m$, such that the following conditions are satisfied [7, Prop. 2.2]:

$$\begin{aligned} &F(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0, \\ &c_i(x^*) \le 0,\ \lambda_i^* \ge 0,\ \lambda_i^* c_i(x^*) = 0,\ i = 1, 2, \ldots, m. \end{aligned} \tag{7}$$

For each $x \in R^n$, we define the set $T(x)$ as

$$T(x) = \{y \in R^n \mid c_i(x) + \langle \nabla c_i(x), y - x \rangle \leq 0, \quad i = 1, 2, \ldots, m\}. \tag{8}$$

Note that, for all $x \in R^n$, the set $T(x)$ is a polyhedral convex set that always contains the set S.

Using the set $T(x)$, we define the function f by

$$f(x) = \max \left\{ -\langle F(x), y - x \rangle - \frac{1}{2} \langle y - x, G(y - x) \rangle \;\middle|\; y \in T(x) \right\}, \tag{9}$$

where G is an $n \times n$ symmetric positive definite matrix. Note that the positive definiteness of G and the convexity of $T(x)$ guarantee that the maximum in (9) is always attained uniquely. Thus we can rewrite (9) as

$$f(x) = -\langle F(x), H(x) - x \rangle - \frac{1}{2} \langle H(x) - x, G(H(x) - x) \rangle, \tag{10}$$

where $H(x)$ is the unique solution y to the quadratic programming problem

$$\text{QP}(x): \quad \begin{array}{ll} \text{minimize}_y & \frac{1}{2}\langle y - x, G(y - x) \rangle + \langle F(x), y - x \rangle \\ \text{subject to} & y \in T(x). \end{array} \tag{11}$$

Note that by the optimality condition for (11), $H(x)$ satisfies the variational inequality

$$\langle F(x) + G(H(x) - x), y - H(x) \rangle \geq 0 \quad \text{for all } y \in T(x). \tag{12}$$

The next lemma characterizes a solution of problem (1) as a fixed point of the mapping H.

LEMMA 2.1. *x is a solution to* (1) *if and only if $H(x) = x$.*

Proof. Let x^* solve (1). It is known [3, p. 143] that if Slater's constraint qualification (6) holds, then x^* also satisfies the inequality

$$\langle F(x^*), y - x^* \rangle \geq 0 \quad \text{for all} \quad y \in T(x^*). \tag{13}$$

Since $H(x^*)$ solves $\text{QP}(x^*)$, $H(x^*)$ satisfies

$$\langle F(x^*) + G(H(x^*) - x^*), y - H(x^*) \rangle \geq 0 \quad \text{for all } y \in T(x^*). \tag{14}$$

Since $H(x^*) \in T(x^*)$, it follows from (13) that

$$\langle F(x^*), H(x^*) - x^* \rangle \geq 0,$$

while since $x^* \in S \subset T(x^*)$, (14) implies

$$\langle F(x^*) + G(H(x^*) - x^*), x^* - H(x^*) \rangle \geq 0.$$

Adding the previous two inequalities, we have

$$\langle G(H(x^*) - x^*), H(x^*) - x^* \rangle \leq 0.$$

Hence, we must have $x^* = H(x^*)$ because of the positive definiteness of G.

On the other hand, suppose $x^* = H(x^*)$. Then we have from (12)

$$\langle F(x^*), y - x^* \rangle \geq 0 \quad \text{for all} \quad y \in T(x^*),$$

which implies

$$\langle F(x^*), y - x^* \rangle \geq 0 \quad \text{for all} \quad y \in S$$

because $S \subset T(x^*)$. Therefore, to prove that x^* solves (1), we only need to show that $x^* \in S$. Since $H(x^*) \in T(x^*)$ and $x^* = H(x^*)$, it follows from the definition (8) of $T(x)$ that $c_i(x^*) \leq 0$ for all $i = 1, 2, \ldots, m$. This completes the proof. □

Now let us consider the optimization problem

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in S, \tag{15}$$

where the function f is defined by (9). The next theorem establishes the equivalence between the VI problem (1) and the optimization problem (15).

THEOREM 2.2. *Let the function $f: R^n \to R$ be defined by* (9). *Then $f(x) \geq 0$ for all $x \in S$. Moreover, $x \in S$ and $f(x) = 0$ if and only if x solves the VI problem* (1). *Hence x solves* (1) *if and only if it solves the optimization problem* (15) *and $f(x) = 0$.*

Proof. Recall that $x \in T(x)$ whenever $x \in S$. So it follows from (9) that

$$\begin{aligned} f(x) &\geq -\langle F(x), x - x \rangle - \frac{1}{2} \langle x - x, G(x - x) \rangle \\ &= 0 \end{aligned}$$

for any $x \in S$. We shall show that $x \in S$ and $f(x) = 0$ if and only if $H(x) = x$. This along with Lemma 2.1 proves the rest of the theorem. First suppose that $x \in S$ and $f(x) = 0$. Since $x \in T(x)$, (12) implies

$$\langle F(x) + G(H(x) - x), x - H(x) \rangle \geq 0,$$

namely,

$$-\langle F(x), H(x) - x \rangle \geq \langle H(x) - x, G(H(x) - x) \rangle. \tag{16}$$

Hence, it follows from (10) and (16) that

$$f(x) \geq \frac{1}{2} \langle H(x) - x, G(H(x) - x) \rangle.$$

But since G is positive definite and $f(x) = 0$, we have $H(x) = x$. Next suppose that $H(x) = x$. Then it follows from (10) that $f(x) = 0$. Moreover, by the feasibility of $H(x)$ to problem (11), we have

$$c_i(x) + \langle \nabla c_i(x), H(x) - x \rangle \leq 0, \ i = 1, \ldots, m.$$

Thus $H(x) = x$ implies $x \in S$. This completes the proof. □

Remark 2.3. This theorem also says that, if the optimization problem (15) has a global minimizer that does not zero the function f, then problem (1) has no solution. On the other hand, when problem (1) does not have a solution, problem (15) may have a global minimum x with $f(x) > 0$. For example, consider the case where $F : R \to R$, $F(x) = -1$, and $S = \{x \in R \mid x \geq 0\}$. Clearly, the corresponding VI problem has no solution. The function f associated with $G = 1$ is $f(x) = \frac{1}{2}$ for all $x \geq 0$, so any $x \geq 0$ is a global minimizer of problem (15). The same remark also holds for the function $\hat{f}$ defined by (4). We note, however, that the condition $f(x) = 0$ is inadvertently missing in the last statement of [5, Thm. 3.1].

Next we consider the continuity and differentiability of f.

DEFINITION 2.4 (see [9]). *Let Φ be a point-to-set-mapping from X into 2^X, where X is a subset of R^n.*

1. Φ *is* open *at a point* $\bar{x}$ *if, for any sequence* $\{x^k\}$ *such that* $x^k \to \bar{x}$, $\bar{y} \in \Phi(\bar{x})$ *implies that there exists a sequence* $\{y^k\}$ *such that* $y^k \in \Phi(x^k)$ *and* $y^k \to \bar{y}$.

2. Φ *is* closed *at a point* $\bar{x}$ *if, for any sequence* $\{x^k\}$ *such that* $x^k \to \bar{x}$, $y^k \in \Phi(x^k)$ *and* $y^k \to \bar{y}$ *imply* $\bar{y} \in \Phi(\bar{x})$.

3. Φ *is* continuous *at a point* $\bar{x}$ *if it is both open and closed at* $\bar{x}$. Φ *is continuous on* X *if it is continuous at any point in* X.

PROPOSITION 2.5. *Let* c_i, $i = 1, 2, \ldots, m$, *be continuously differentiable and satisfy Slater's constraint qualification* (6). *Then the point-to-set mapping* T *defined by* (8) *is continuous on* R^n.

Proof. Since $c_i(x) + \langle \nabla c_i(x), y - x \rangle$ is continuous with respect to (x, y) for each i, it follows from [9, Thm. 10] that T is closed at each x. By assumption, there is an $\hat{x}$ such that $c_i(\hat{x}) < 0$ for all $i = 1, 2, \ldots, m$. Moreover, from the convexity of c_i, we have $c_i(x) + \langle \nabla c_i(x), \hat{x} - x \rangle \leq c_i(\hat{x}) < 0$. Hence, it follows from [9, Thm. 12] that T is open at each x. Therefore, T is continuous on R^n. □

LEMMA 2.6. *Let the mapping* F *be continuous. If Slater's constraint qualification* (6) *holds, then the mapping* H *is bounded on any bounded set.*

Proof. Suppose that H is not bounded on some bounded set B. Then there is a sequence $\{x^k\}$ in B such that $\| H(x^k) \| \to \infty$. Since $\{F(x^k)\}$ is bounded on B by the continuity of F and since G is positive definite, it follows from (10) that $f(x^k) \to -\infty$. On the other hand, since $\hat{x} \in T(x^k)$ for all k, we have

$$\begin{aligned} f(x^k) &= \max \left\{ -\langle F(x^k), y - x^k \rangle - \frac{1}{2} \langle y - x^k, G(y - x^k) \rangle \;\middle|\; y \in T(x^k) \right\} \\ &\geq -\langle F(x^k), \hat{x} - x^k \rangle - \frac{1}{2} \langle \hat{x} - x^k, G(\hat{x} - x^k) \rangle, \end{aligned}$$

which is bounded below by the boundedness of $\{x^k\}$ and $\{F(x^k)\}$. This is a contradiction. □

DEFINITION 2.7. *Let* ϕ *be a function from* R^n *into* $[-\infty, +\infty]$, *and let* x *be a point where* ϕ *is finite. We say that* ϕ *is* directionally differentiable *at* x *in the direction* d *if the limit*

$$\lim_{t \to 0, t > 0} \frac{\phi(x + td) - \phi(x)}{t} \tag{17}$$

exists, and we call the limit the directional derivative *and denote it by* $\phi'(x; d)$.

THEOREM 2.8. *Suppose that the mapping* F *is continuous and* c_i, $i = 1, 2, \ldots, m$, *are continuously differentiable. Suppose also that Slater's constraint qualification* (6) *holds. Then the function* f *defined by* (10) *is continuous on* R^n. *Moreover, if* F *is continuously differentiable and* c_i, $i = 1, \ldots, m$, *are twice continuously differentiable, then* f *is directionally differentiable in any direction* $d \in R^n$ *and its directional derivative* $f'(x; d)$ *is given by*

$$f'(x; d) = \min_{\lambda \in \Lambda(x)} \left\langle F(x) - [\nabla F(x) - G](H(x) - x) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)(H(x) - x),\ d \right\rangle, \tag{18}$$

where $\Lambda(x)$ *is defined by*

$$\begin{aligned} \Lambda(x) = \Bigg\{ \lambda \in R^m \;\Bigg|\; & F(x) + G(H(x) - x) + \sum_{i=1}^{m} \lambda_i \nabla c_i(x) = 0, \lambda_i \geq 0, \\ & \lambda_i [c_i(x) + \langle \nabla c_i(x), H(x) - x \rangle] = 0, i = 1, \ldots, m \Bigg\}. \end{aligned} \tag{19}$$

Proof. To prove the first half, it is sufficient to show that the mapping $H(x)$ is continuous. Under the given assumptions, the point-to-set mapping T is continuous by Proposition 2.5 and the objective function of (11) is continuous with respect to (x, y). Therefore, by [9, Thm. 8], the mapping H is closed. However, since H is everywhere defined and single valued, H must be continuous for any x. The second half follows from [8, Thm. 2]. □

Remark 2.9. If the set $\Lambda(x)$ is a singleton, f is differentiable at x and the gradient is given by

$$\nabla f(x) = F(x) - [\nabla F(x) - G](H(x) - x) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)(H(x) - x).$$

A sufficient condition [4, Thm. 6] for $\Lambda(x)$ to be a singleton is that the vectors $\nabla c_i(x), i \in I(x)$, are linearly independent, where $I(x) = \{i \mid c_i(x) + \langle \nabla c_i(x), H(x) - x\rangle = 0\}$, and the strict complementarity condition is satisfied i.e., $\lambda_i = 0$ implies $c_i(x) + \langle \nabla c_i(x), H(x) - x\rangle < 0$.

Remark 2.10. The merit function $\hat{f}$ defined by (4) also has the property that the set of zeros of $\hat{f}$ on S coincides with the set of solutions to problem (1). Moreover, $\hat{f}$ is continuously differentiable when the mapping F is. It is easy to see that $f(x) \geq \hat{f}(x)$ because $T(x)$ contains S for all x. In particular, when c_i are all linear, (4) coincides with (9). Thus, the function f is regarded as a relaxed version of $\hat{f}$.

To obtain a solution to (1) by solving problem (15), we must find a global minimizer of f on S. It is therefore desirable to know conditions under which a stationary point of (15) is a global optimal solution, because most optimization algorithms only find a stationary point of the problem. The next theorem answers the question.

THEOREM 2.11. *Suppose that the mapping F is continuously differentiable, ∇F is positive definite, the convex functions c_i, $i = 1, \ldots, m$, are twice continuously differentiable, and Slater's constraint qualification* (6) *is satisfied. If $x \in S$ and*

$$f'(x; y - x) \geq 0 \quad \textit{for all } y \in S, \tag{20}$$

then x is a solution to (1).

Before proving the theorem, we show the next lemma. The lemma will also be used to derive a descent condition in the next section.

LEMMA 2.12. *Let the mapping F be continuously differentiable and the convex functions c_i, $i = 1, \ldots, m$, be twice continuously differentiable. If $d = H(x) - x$, then we have*

$$f'(x; d) \leq -\langle d, \nabla F(x) d\rangle + \min_{\lambda \in \Lambda(x)} \left(\sum_{i \in I_+} \lambda_i c_i(x) \right), \tag{21}$$

where $I_+ = \{i | c_i(x) > 0\}$.

Proof. Since $d = H(x) - x$, it follows from the Karush–Kuhn–Tucker conditions for problem (11) that d together with a Lagrange multiplier vector λ satisfies

$$F(x) + Gd + \sum_{i=1}^{m} \lambda_i \nabla c_i(x) = 0, \tag{22a}$$

$$c_i(x) + \langle \nabla c_i(x), d\rangle \leq 0, \quad \lambda_i \geq 0, \tag{22b}$$

$$\lambda_i [c_i(x) + \langle \nabla c_i(x), d\rangle] = 0, \quad i = 1, \ldots, m. \tag{22c}$$

From (18), (22a), and (22c), we have

$$
\begin{aligned}
f'(x;d) &= \min_{\lambda\in\Lambda(x)} \left\langle F(x) - [\nabla F(x) - G]d - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)d,\ d \right\rangle \\
&= \min_{\lambda\in\Lambda(x)} \left\{ -\langle d, \nabla F(x)d\rangle - \left\langle d, \sum_{i=1}^{m} \lambda_i \nabla c_i(x) \right\rangle - \left\langle d, \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)d \right\rangle \right\} \qquad (23) \\
&= \min_{\lambda\in\Lambda(x)} \left\{ -\langle d, \nabla F(x)d\rangle + \sum_{i=1}^{m} \lambda_i c_i(x) - \sum_{i=1}^{m} \lambda_i \langle d, \nabla^2 c_i(x)d\rangle \right\}.
\end{aligned}
$$

Since the convexity of c_i ensures that $\nabla^2 c_i(x)$ is positive semidefinite, i.e., $\langle d, \nabla c_i^2(x)d\rangle \geq 0$, and since $\sum_{i\notin I_+} \lambda_i c_i(x) \leq 0$, we have from (23)

$$
f'(x;d) \leq -\langle d, \nabla F(x)d\rangle + \min_{\lambda\in\Lambda(x)} \left(\sum_{i\in I_+} \lambda_i c_i(x) \right).
$$

This completes the proof. □

Proof of Theorem 2.11. Suppose that $x \in S$ satisfies (20). First note that, under Slater's constraint qualification, we have

$$
T(x) \subset x + \mathrm{cl}\{\alpha(y-x) \mid y \in S, \alpha > 0\}, \qquad (24)
$$

where cl denotes closure of a set [3, p. 143]. Thus, from (20), (24), and the positive homogeneity of $f'(x;\cdot)$, we can deduce that

$$
f'(x; y-x) \geq 0 \quad \text{for all } y \in T(x). \qquad (25)
$$

Since $x \in S$ implies $I_+ = \phi$, it follows from Lemma 2.12 that

$$
f'(x; H(x)-x) \leq -\langle H(x)-x, \nabla F(x)(H(x)-x)\rangle. \qquad (26)
$$

Since $H(x) \in T(x)$, we have from (25) and (26) that

$$
\langle H(x)-x, \nabla F(x)(H(x)-x)\rangle \leq 0.
$$

But since $\nabla F(x)$ is positive definite, we have $x = H(x)$. Therefore, it follows from Lemma 2.1 that x is a solution to (1). □

3. Successive quadratic programming algorithm. In this section, we present a successive quadratic programming algorithm for solving the VI problem (1) and prove its convergence.

First, we show that the vector

$$
d = H(x) - x
$$

is a descent direction of the penalty function $\theta_r : R^n \to R$ defined by

$$
\theta_r(x) = f(x) + r \sum_{i=1}^{m} \max(0, c_i(x)), \qquad (27)
$$

where r is a sufficiently large positive parameter.

THEOREM 3.1. *Let the mapping F be continuously differentiable and the convex functions c_i, $i = 1, \ldots, m$, be twice continuously differentiable. If $\nabla F(x)$ is positive definite and*

$$\| \lambda \|_\infty \leq r \quad \text{for all } \lambda \in \Lambda(x),$$

then the vector $d = H(x) - x$ satisfies the descent condition

$$\theta_r'(x; d) < 0$$

whenever $d \neq 0$.

Proof. Let $I_+ = \{i \mid c_i(x) > 0\}$ and $I_0 = \{i \mid c_i(x) = 0\}$. By Theorem 2.8 and [6, Lem. 3.1], θ_r is directionally differentiable and the directional derivative is given by

$$\theta_r'(x; d) = f'(x; d) + r \sum_{i \in I_+} \langle \nabla c_i(x), d \rangle + r \sum_{i \in I_0} \max(0, \langle \nabla c_i(x), d \rangle). \tag{28}$$

First note that $d = H(x) - x$ together with a Lagrange multiplier vector λ satisfies

$$F(x) + Gd + \sum_{i=1}^{m} \lambda_i \nabla c_i(x) = 0, \tag{29a}$$

$$c_i(x) + \langle \nabla c_i(x), d \rangle \leq 0, \quad \lambda_i \geq 0, \tag{29b}$$

$$\lambda_i [c_i(x) + \langle \nabla c_i(x), d \rangle] = 0, \quad i = 1, \ldots, m. \tag{29c}$$

Then (29b) yields

$$\sum_{i \in I_0} \max(0, \langle \nabla c_i(x), d \rangle) = 0. \tag{30}$$

By Lemma 2.12, we have

$$f'(x; d) \leq -\langle d, \nabla F(x) d \rangle + \min_{\lambda \in \Lambda(x)} \left(\sum_{i \in I_+} \lambda_i c_i(x) \right). \tag{31}$$

Hence, it follows from (28), (29b), (30), and (31) that

$$\begin{aligned} \theta_r'(x; d) &\leq -\langle d, \nabla F(x) d \rangle + \min_{\lambda \in \Lambda(x)} \left(\sum_{i \in I_+} \lambda_i c_i(x) \right) + r \sum_{i \in I_+} \langle \nabla c_i(x), d \rangle \\ &\leq -\langle d, \nabla F(x) d \rangle + \min_{\lambda \in \Lambda(x)} \left(\sum_{i \in I_+} (\lambda_i - r) c_i(x) \right) \\ &< 0 \end{aligned}$$

because, by assumption, $\nabla F(x)$ is positive definite and $\| \lambda \|_\infty \leq r$ for all $\lambda \in \Lambda(x)$. The proof is complete. □

ALGORITHM.

Step 0. Choose x^0, $r > 0$ and a symmetric positive definite matrix G. Let $k := 0$.

Step 1. Find the unique solution d^k of the following quadratic programming problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\langle d, Gd\rangle + \langle F(x^k), d\rangle \\ \text{subject to} & c_i(x^k) + \langle \nabla c_i(x^k), d\rangle \leq 0, \quad i = 1, 2, \ldots, m. \end{array}$$

Step 2. Set $x^{k+1} := x^k + t_k d^k$, where

$$t_k := \arg \min_{0 \leq t \leq 1} \theta_r(x^k + td^k).$$

Let $k := k + 1$. Go to Step 1.

The next theorem establishes the global convergence of this algorithm.

THEOREM 3.2. *Suppose that $\nabla F(x)$ is positive definite on R^n. Suppose also that the parameter r is chosen sufficiently large. If the sequence $\{x^k\}$ generated by the algorithm is bounded, then $\{x^k\}$ converges to the unique solution to the VI problem* (1).

Proof. To prove the global convergence, we shall use Zangwill's global convergence Theorem A [17, p. 91]. Since $\{x^k\}$ is bounded, it follows from [6, Lem. 3.3] that there exists a positive number $\bar{r}$ such that $\| \lambda^k \|_\infty \leq \bar{r}$ for all k, where λ^k is an arbitrary optimal Lagrange multiplier vector of the quadratic program solved in Step 1 at iteration k. Assuming that $r \geq \bar{r}$, we have from Theorem 3.1 that d^k satisfies the descent condition $\theta_r'(x^k; d^k) < 0$ whenever x^k is not a solution to (1). Moreover, since the map $d = H(x) - x$ and the function θ_r are continuous with respect to x and since the exact line search strategy on a bounded interval is closed in the sense of [17], the overall algorithmic map is closed. Therefore, the previously mentioned theorem [17] guarantees that any accumulation point of $\{x^k\}$ is a solution to (1). Since problem (1) has at most one solution by the positive definiteness of $\nabla F(x)$, we conclude that the entire sequence converges to the solution to (1). □

Remark 3.3. When F is a gradient mapping of some differentiable convex function φ, problem (1) corresponds to a necessary and sufficient optimality condition for the convex programming problem

$$\begin{array}{ll} \text{minimize} & \varphi(x) \\ \text{subject to} & c_i(x) \leq 0, \quad i = 1, 2, \ldots, m. \end{array} \tag{32}$$

We may therefore apply the algorithm to (32) with the identification $F = \nabla\varphi$. Then the subproblem solved in Step 1 is the same as that of a standard successive quadratic programming problem [6], but the merit function used in the line search is quite different.

Acknowledgments. The authors thank a referee for a helpful comment that led to Remark 2.3.

REFERENCES

[1] G. AUCHMUTY, *Variational principles for variational inequalities*, Numer. Funct. Anal. Optim., 10 (1989), pp. 863–874.

[2] A. AUSLENDER, *Optimisation: Méthodes Numériques*, Masson, Paris, 1976.

[3] M. S. BAZARAA AND C. M. SHETTY, *Foundations of Optimization*, Springer-Verlag, Berlin, Heidelberg, 1976.

[4] A. V. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[5] M. FUKUSHIMA, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.
[6] S. P. HAN, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.
[7] P. T. HARKER AND J. S. PANG, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.
[8] W. HOGAN, *Directional derivatives for extremal-value functions with applications to the completely convex case*, Oper. Res., 21 (1973), pp. 187–209.
[9] ———, *Point-to-set maps in mathematical programming*, SIAM Rev., 15 (1973), pp. 591–603.
[10] T. LARSSON AND M. PATRIKSSON, *A class of gap functions for variational inequalities*, Math. Programming, 64 (1994), pp. 53–79.
[11] P. MARCOTTE, *A new algorithm for solving variational inequalities with application to the traffic assignment problem*, Math. Programming, 33 (1985), pp. 339–351.
[12] P. MARCOTTE AND J. P. DUSSAULT, *A note on a globally convergent Newton method for solving variational inequalities*, Oper. Res. Lett., 6 (1987), pp. 35–42.
[13] A. NAGURNEY, *Network Economics: A Variational Inequality Approach*, Kluwer Academic, Norwell, MA, 1993.
[14] J. S. PANG, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.
[15] K. TAJI, M. FUKUSHIMA, AND T. IBARAKI, *A globally convergent Newton method for solving strongly monotone variational inequalities*, Math. Programming, 58 (1993), pp. 369–383.
[16] J. H. WU, M. FLORIAN, AND P. MARCOTTE, *A general descent framework for the monotone variational inequality problem*, Math. Programming, 61 (1993), pp. 281–300.
[17] W. I. ZANGWILL, *Nonlinear Programming: A Unified Approach*, Prentice–Hall, Englewood Cliffs, NJ, 1969.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 714–726, August 1996
© 1996 Society for Industrial and Applied Mathematics
011

CO-COERCIVITY AND ITS ROLE IN THE CONVERGENCE OF ITERATIVE SCHEMES FOR SOLVING VARIATIONAL INEQUALITIES *

D. L. ZHU† AND P. MARCOTTE‡

Abstract. This paper stresses the role of co-coercivity and related notions in the convergence of iterative schemes for solving monotone but not necessarily strongly or even strictly monotone variational inequalities. The analysis will be conducted within the conceptual framework of the "auxiliary problem principle."

Key words. mathematical programming, variational inequalities, monotonicity, auxiliary problem, co-coercivity

AMS subject classifications. 47H05, 49J40, 49M27, 90C25

1. Introduction.

Let C denote a nonempty, closed, convex subset of R^n, F a continuous mapping from C into R^n, and f a continuous convex function from C into R. We say that a point x^* is a solution of the variational inequality VIP if it satisfies

$$\text{VIP}: \qquad \langle F(x^*), x - x^* \rangle + f(x) - f(x^*) \geq 0 \quad \forall x \in C, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard dot product in R^n. Throughout the paper, we assume that the mapping F is monotone over C and that the function f is convex over C.

Various iterative schemes have been proposed for solving variational inequalities. The interested reader may consult the survey of Harker and Pang [6] for a good introduction to the subject. For several iterative methods, the convergence proof requires the mapping F to be strongly monotone over C. Recently, Tseng [12] and Marcotte and Wu [8] studied the convergence of iterative processes when the mapping F is affine and co-coercive. It is the main purpose of this paper to investigate iterative schemes for solving nonlinear variational inequalities under the co-coercivity assumption.

The paper will proceed as follows. Properties of co-coercive mappings are introduced in §2; in §3 we address the global convergence of a very general iterative scheme; in §4 we study a variant of the iterative scheme introduced in §3 and consider various realizations of this method. In particular, we will stress the relationships with asymmetric projection methods and consider its application to the traffic equilibrium problem. Next we introduce the notion of partial co-coercivity that will prove useful in the analysis of algorithms for solving systems of variational inequalities. Finally we will propose a primal-dual realization of our general algorithmic framework.

2. Co-coercive mappings and their properties.

In this paper we will denote by $\|x\|_B$ the norm induced by the positive definite matrix B, i.e.,

$$\|x\|_B = \langle Bx, x \rangle^{1/2}.$$

* Received by the editors June 1, 1994; accepted for publication (in revised form) February 20, 1995.

† Centre de recherche sur les transports, Université de Montréal, Montréal, Québec, Canada H3C 3J7 (daoli@crt.umontreal.ca).

‡ Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, Succursale A, Montréal, Québec, Canada H3C 3J7 (marcotte@iro.umontreal.ca).

Also, $\|x\|_2$ will denote the standard norm in R^n. For any vector norm $\|\cdot\|$, the corresponding matrix norm is

$$\|A\| = \max_{\|x\|=1} \|Ax\|.$$

DEFINITION 2.1. *The mapping F is co-coercive on C if there exists a positive constant α such that*

$$\langle F(y) - F(x), y - x\rangle \geq \alpha \|F(y) - F(x)\|^2 \tag{2}$$

for all x, y in C.

It is clear that co-coercive mappings are monotone but not necessarily strongly monotone (consider a constant mapping). Conversely, strongly monotone and Lipschitz continuous mappings are co-coercive, and it follows that co-coercivity is an intermediate concept that lies between simple and strong monotonicity.

The sum of co-coercive mappings is trivially co-coercive. As shown below, co-coercivity is also preserved under affine transformations, which include scalings and translations as particular cases.

PROPOSITION 2.2. *Let A be an $m \times n$ matrix. If the mapping F from R^m into R^m is co-coercive with modulus α, then the mapping $y \to A^t F(Ay) + d$ is co-coercive with modulus $\alpha/\|A\|^2$.*

Proof. Assume that A is nonzero (otherwise the result holds trivially). We have

$$\begin{aligned}
\langle [A^t F(Ay) + d] - [A^t F(Ax) + d], y - x\rangle &= \langle F(Ay) - F(Ax), Ay - Ax\rangle \\
&\geq \alpha \|F(Ay) - F(Ax)\|^2 \\
&\geq \frac{\alpha}{\|A\|^2} \|A^t F(Ay) - A^t F(Ax)\|^2 \\
&= \frac{\alpha}{\|A\|^2} \|[A^t F(Ay) + d] - [A^t F(Ax) + d]\|^2,
\end{aligned}$$

and $A^t F(Ay) + d$ is co-coercive, as claimed. □

Co-coercive mappings are monotone and Lipschitz continuous over their domain of definition. Although the converse does not hold in general, it does hold in R^1, where a function is co-coercive on an interval if and only if it is monotone (increasing) and Lipschitz continuous on that interval. The next properties of co-coercive mappings will be required in subsequent sections of this paper.

PROPOSITION 2.3. *Let F be a mapping from a closed convex subset C of R^n into R^n. If the mapping $F - \alpha I$ is Lipschitz continuous with Lipschitz constant β ($\beta \leq \alpha$) over C, then F is co-coercive on C. Conversely, if F is co-coercive with modulus β ($\beta \geq \frac{1}{2\alpha}$) on C, then $F - \alpha I$ is Lipschitz continuous on C, with Lipschitz constant α.*

Proof. Denote $G = F - \alpha I$. For any x, y in C we can write

$$\begin{aligned}
\|F(y) - F(x)\|^2 &= \|G(y) - G(x) + \alpha y - \alpha x\|^2 \\
&= \|G(y) - G(x)\|^2 + 2\alpha\langle G(y) - G(x), y - x\rangle + \alpha^2 \|y - x\|^2 \\
&\leq (\beta^2 + \alpha^2)\|y - x\|^2 + 2\alpha\langle G(y) - G(x), y - x\rangle \\
&\leq 2\alpha\langle [G(y) + \alpha y] - [G(x) + \alpha x], y - x\rangle,
\end{aligned}$$

and this implies

$$\langle F(y) - F(x), y - x\rangle \geq \frac{1}{2\alpha} \|F(y) - F(x)\|^2,$$

i.e., that F is co-coercive on C with modulus $\frac{1}{2\alpha}$.

As for the second claim, observe that

$$\begin{aligned}\|(F-\alpha I)(y)-(F-\alpha I)(x)\|^2 &= \|F(y)-F(x)\|^2 \\ &\quad -2\alpha\langle F(y)-F(x), y-x\rangle + \alpha^2\|y-x\|^2 \\ &\leq \alpha^2\|y-x\|^2,\end{aligned}$$

i.e., $F-\alpha I$ is Lipschitz continuous on C. □

PROPOSITION 2.4. *Let the mapping $F: C \to R^n$ be invertible. Then F is co-coercive on C if and only if F^{-1} is Lipschitz continuous and strongly monotone on the range $F(C)$ of F.*

Proof. The proof of the proposition is straightforward from the definitions. □

As shown by Marcotte and Wu [8], a sufficient condition for a mapping F to be co-coercive on a closed convex set C is that the Jacobian $F'(x)$ and the *square* of the Jacobian $F'(x)^2$ be both positive semidefinite on C.

In the affine case, the concept of co-coercivity can be related to that of psd-plus matrices. By definition, a matrix M is psd-plus if it is positive semidefinite, and $Mx = 0$ whenever $\langle Mx, x\rangle = 0$. Recently, Luo and Tseng [7] characterized the class of psd-plus matrices as follows: a matrix M is psd-plus if and only if it can be decomposed into the product

$$M = E^t A E,$$

where A is a positive definite matrix. From this result, the following result can be shown.

PROPOSITION 2.5. *A matrix M is psd-plus if and only if the mapping $x \to Mx$ is co-coercive.*

Proof. The result holds trivially if $M = 0$. Assume that M is nonzero. If Mx is co-coercive, then we have

$$\langle Mx, x\rangle \geq \alpha\|Mx\|^2,$$

which implies that M is psd-plus.

Conversely, let M be a psd-plus matrix. Decompose M as $M = E^t A E$, where A is a positive definite matrix. Let α be the smallest eigenvalue associated with the symmetric part of A. The number α is positive and we can write

$$\langle Mx, x\rangle = \langle E^t A E x, x\rangle = \langle A(Ex), Ex\rangle \geq \alpha\|Ex\|^2. \tag{3}$$

Also, since M is nonzero, $\|E^t A\|$ is nonzero and we have

$$\|Mx\|^2 = \|E^t A E x\|^2 \leq \|E^t A\|^2\|Ex\|^2. \tag{4}$$

Coupling (3) and (4) gives

$$\langle Mx, x\rangle \geq \frac{\alpha}{\|E^t A\|^2}\|Mx\|^2,$$

and the mapping Mx is co-coercive on R^n. □

3. A general iterative scheme. The co-coercivity condition can be used to analyze the convergence behavior of iterative schemes based upon the auxiliary problem framework developed by Cohen [4]. The following lemma will be required in our analysis.

LEMMA 3.1. *Let h be a continuously differentiable function defined on a convex subset C of R^n.*

(i) *If h is strongly convex with modulus b over C, then*

$$h(x_2) - h(x_1) \geq \langle h'(x_1), x_2 - x_1 \rangle + \frac{1}{2} b \|x_2 - x_1\|^2 \quad \forall x_1, x_2 \in C.$$

(ii) *If the the gradient h' is Lipschitz continuous over C, with Lipschitz constant B, then*

$$h(x_2) - h(x_1) \leq \langle h'(x_1), x_2 - x_1 \rangle + \frac{1}{2} B \|x_2 - x_1\|^2 \quad \forall x_1, x_2 \in C.$$

Proof. (i) See [1].

(ii) For any x_1, $x_2 \in C$, define the function

$$\phi(t) = h((1-t)x_1 + tx_2) - h(x_1) - t\langle h'(x_1), x_2 - x_1 \rangle \quad \forall t \in [0,1].$$

We have

$$\phi'(t) = \langle h'((1-t)x_1 + tx_2) - h'(x_2), x_2 - x_1 \rangle$$

and

$$\begin{aligned}
\phi(1) &= h(x_2) - h(x_1) - \langle h'(x_1), x_2 - x_1 \rangle \\
&= \phi(0) + \int_0^1 \phi'(t)dt \quad \text{by the fundamental theorem of calculus} \\
&\leq 0 + \int_0^1 |\phi'(t)|dt \quad \text{since } \phi(0) = 0 \\
&\leq \int_0^1 B\|t(x_2 - x_1)\| \cdot \|x_2 - x_1\| dt \\
&= \frac{B}{2}\|x_2 - x_1\|^2. \qquad \square
\end{aligned}$$

We now introduce the basic algorithmic framework. From now on, we assume that the solution set of VIP is nonempty. Let ϵ be a positive parameter and consider, for a given iterate x^k, the auxiliary VIP that consists of finding a point x^{k+1} that satisfies the variational inequality

$$\langle \epsilon F(x^k) + \Gamma(x^{k+1}) - \Gamma(x^k), x - x^{k+1} \rangle + \epsilon(f(x) - f(x^{k+1})) \geq 0 \quad \forall x \in C, \tag{5}$$

where Γ is a mapping approximating the mapping F.

If we select Γ to be a gradient mapping, e.g., $\Gamma(x) = h'(x)$, we have the following convergence result.

THEOREM 3.2. *Let F be co-coercive over C (with modulus γ) and $\Gamma = h'$. Let h be strongly convex over C (with modulus b) and its derivative h' be Lipschitz continuous over C (with Lipschitz constant B). Then there exists a unique solution x^{k+1} to* (5).

If in addition we take $0 < \epsilon < 2b\gamma$, *then the sequence* $\{x^k\}$ *is bounded and converges to a solution of the VIP.*

Proof. Since the function h is strongly convex over C, the solution x^{k+1} of (5) is unique. Let x^* denote *any* fixed solution of the original variational inequality VIP and consider, for x in C, the function

$$\begin{aligned}\Lambda^*(x) &= h(x^*) - h(x) - \langle h'(x), x^* - x\rangle \\ &\geq (b/2)\|x - x^*\|^2,\end{aligned}$$

where the last inequality follows from the strong monotonicity of h'. We have

$$\begin{aligned}\Lambda^*(x^k) - \Lambda^*(x^{k+1}) &= h(x^{k+1}) - h(x^k) - \langle h'(x^k), x^{k+1} - x^k\rangle \\ &\quad + \langle h'(x^{k+1}) - h'(x^k), x^* - x^{k+1}\rangle \qquad (6) \\ &\geq \frac{b}{2}\|x^k - x^{k+1}\|^2 \\ &\quad + \epsilon\langle F(x^k), x^{k+1} - x^*\rangle + \epsilon[f(x^{k+1}) - f(x^*)], \qquad (7)\end{aligned}$$

where the second term of the inequality is obtained by setting $x = x^*$ in equation (5). If we set $x = x^{k+1}$ in (1) and combine (1) and (7), we obtain

$$\begin{aligned}\Lambda^*(x^k) - \Lambda^*(x^{k+1}) &\geq \frac{b}{2}\|x^k - x^{k+1}\|^2 + \epsilon\langle F(x^k) - F(x^*), x^{k+1} - x^*\rangle \\ &= \frac{b}{2}\|x^k - x^{k+1}\|^2 + T.\end{aligned}$$

Now

$$\begin{aligned}T &= \epsilon\langle F(x^k) - F(x^*), x^{k+1} - x^*\rangle \\ &= \epsilon\langle F(x^k) - F(x^*), x^k - x^*\rangle + \epsilon\langle F(x^k) - F(x^*), x^{k+1} - x^k\rangle.\end{aligned}$$

Since F is co-coercive, we obtain

$$\begin{aligned}T &\geq \epsilon[\gamma\|F(x^k) - F(x^*)\|^2 + \langle F(x^k) - F(x^*), x^{k+1} - x^k\rangle] \\ &\geq -\frac{\epsilon}{4\gamma}\|x^{k+1} - x^k\|^2.\end{aligned}$$

Therefore

$$\Lambda^*(x^k) - \Lambda^*(x^{k+1}) \geq \frac{1}{2}\left(b - \frac{\epsilon}{2\gamma}\right)\|x^{k+1} - x^k\|^2.$$

If $x^{k+1} = x^k$, it is easily shown that x^k is a solution of the variational inequality. Otherwise, the assumption $\epsilon < 2b\gamma$ implies that the sequence $\Lambda^*(x^k) - \Lambda^*(x^{k+1})$ is nonnegative, and we must have

$$\lim_{k\to\infty}\|x^{k+1} - x^k\| = 0.$$

Furthermore, since $\|x^k - x^*\|^2 \leq (2/b)\Lambda^*(x^k)$ and the sequence $\{\Lambda^*(x^k)\}$ is decreasing, we can affirm that the sequence $\{x^k\}$ is bounded. Let $\bar{x}$ be any cluster point of the sequence $\{x^k\}$. Taking the limit in (5), we conclude that $\bar{x}$ is a solution of the variational inequality.

If we replace x^* by $\bar{x}$, the analysis remains valid for $\bar{x}$ and its associated metric function $\bar{\Lambda}$. The sequence $\{\bar{\Lambda}(x^k)\}$ still strictly decreases and, from Lemma 3.1, we have

$$\bar{\Lambda}(x^k) \leq \frac{1}{2} B \|\bar{x} - x^k\|^2.$$

This shows that the sequence $\bar{\Lambda}(x^k)$ converges to zero. This result, together with the inequality

$$\bar{\Lambda}(x^k) \geq \frac{b}{2} \|x^k - \bar{x}\|^2,$$

allows us to conclude that the entire sequence $\{x^k\}$ converges to $\bar{x}$. □

Remark 3.3. The previous convergence result constitutes an extension of Cohen's result to the co-coercive case. Cohen [4] gave an example, involving a monotone mapping F, where the iterative method did not converge. In his example, however, the mapping F failed to be co-coercive.

Remark 3.4. A related algorithm has been studied in Mataoui [10] in the case $f = 0$, where the author proved only the existence of a convergent subsequence.

From a practical point of view, the solution of the auxiliary variational inequality cannot be obtained in closed form, and a trade-off must be achieved between the amount of work spent on solving the auxiliary problem and the accuracy of the corresponding solution. More precisely one can, at step k, find an approximate solution of the auxiliary problem, i.e., a point x^{k+1} of C satisfying

$$(8)\ \langle \epsilon F(x^k) + h'(x^{k+1}) - h'(x^k), x - x^{k+1} \rangle + \epsilon(f(x) - f(x^{k+1})) \geq -\rho_k \quad \forall x \in C',$$

where ρ_k is a positive number. If C is bounded we let $C' = C$. Otherwise, we define $C' = C \cap \{x | \|x\| \leq R\}$, where R is a suitably large constant. Although such a number always exists (recall that the sequence $\{\|x^k - x^*\|\}$ is bounded), it could be difficult to estimate in practice. One possibility is to update R whenever the sequence of iterates converges to a nonequilibrium point at the boundary of the set C'.

THEOREM 3.5. *If the conditions of Theorem* 3.2 *are fulfilled and the sequence* $\{\rho_k\}$ *satisfies*

$$\rho_k \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k < \infty,$$

then the sequence $\{x^k\}$ *converges to a solution of VIP.*

Proof. Following the steps of the proof of Theorem 3.2 we have

$$\Lambda^*(x^{k+1}) - \Lambda^*(x^k) \leq -\frac{1}{2}\left(b - \frac{\epsilon}{2\gamma}\right) \|x^{k+1} - x^k\|^2 + \rho_k.$$

Since $2b\gamma - \epsilon > 0$ and $\sum_{k=1}^{\infty} \rho_k < \infty$ we have, for any N and any solution x^* to VIP,

$$\frac{b}{2}\|x^{N+1} - x^*\|^2 - \Lambda^*(x^1) \leq \sum_{k=1}^{N} (\Lambda^*(x^{k+1}) - \Lambda^*(x^k)) \leq \sum_{k=1}^{\infty} \rho_k.$$

This implies that the sequence $\{x^k\}$ is bounded and that the limit of $\|x^{k+1} - x^k\|$ is zero. It is then easy to show, exactly as in the proof of Theorem 3.2, that the entire sequence converges to a solution of the variational inequality. □

4. A modified auxiliary mapping and its use in the decomposition of variational inequalities. In this section we present a more general iterative scheme that converges under less stringent monotonicity assumptions on the mapping F.

4.1. A general method. For a given $x^k \in C$, let us consider the variational inequality that consists of determining $x^{k+1} \in C$ such that

$$\langle \epsilon F(x^k) + \Gamma_\epsilon(x^{k+1}) - \Gamma_\epsilon(x^k), x - x^{k+1}\rangle + \epsilon(f(x) - f(x^{k+1})) \geq 0 \quad \forall x \in C, \tag{9}$$

where Γ_1 and Γ_2 are nonlinear mappings and $\Gamma_\epsilon = \epsilon\Gamma_1 + \Gamma_2$. We will focus our attention on the choice

$$\Gamma_1(x) = Mx + L(x),$$
$$\Gamma_2(x) = h'(x),$$

where M is a symmetric matrix, L a monotone mapping, and h is a convex function with Lipschitz continuous gradient. If $\Gamma_1 = 0$ we recover the preceding method. Let us denote by $\lambda_{\min}(\Delta)$ the smallest eigenvalue of a matrix Δ.

THEOREM 4.1. *Let h' be strongly monotone with modulus b over C and $F - L$ be co-coercive with modulus γ over C. Select ϵ such that*

$$0 < \epsilon \leq \frac{2b\gamma}{1 - 2\gamma\lambda_{\min}(M)} \qquad \text{if } \frac{1}{2\gamma} - \lambda_{\min}(M) > 0, \tag{10}$$

$$0 < \epsilon < \infty \qquad \text{if } \frac{1}{2\gamma} - \lambda_{\min}(M) \leq 0. \tag{11}$$

Then there exists a unique solution x^{k+1} to (9). Also, the sequence $\{x^k\}$ is bounded and converges to a solution of VIP.

Proof. Let x^* be any fixed solution of VIP. Let $G = I + (\epsilon/b)M$ and consider the function

$$\begin{aligned}\Lambda^*(x) &= h(x^*) - h(x) - \langle h'(x), x^* - x\rangle + \frac{\epsilon}{2}\langle M(x - x^*), x - x^*\rangle \\ &\geq \frac{b}{2}\|x - x^*\|^2 + \frac{\epsilon}{2}\langle M(x - x^*), x - x^*\rangle \\ &= \frac{b}{2}\|x - x^*\|_G^2.\end{aligned}$$

From the assumptions, the matrix G is positive definite and symmetric. We have

$$\begin{aligned}\Lambda^*(x^k) - \Lambda^*(x^{k+1}) &= h(x^{k+1}) - h(x^k) - \langle h'(x^k), x^{k+1} - x^k\rangle \\ &\quad + \langle h'(x^{k+1}) - h'(x^k), x^* - x^{k+1}\rangle + \frac{\epsilon}{2}\langle M(x^k - x^*), x^k - x^*\rangle \\ &\quad - \frac{\epsilon}{2}\langle M(x^{k+1} - x^*), x^{k+1} - x^*\rangle \\ &\geq \frac{b}{2}\|x^k - x^{k+1}\|^2 + \frac{\epsilon}{2}[\langle M(x^k - x^*), x^k - x^*\rangle \\ &\quad - \langle M(x^{k+1} - x^*), x^{k+1} - x^*\rangle] \\ &\quad + \epsilon[\langle F(x^k) + M(x^{k+1} - x^k) + L(x^{k+1}) - L(x^k), x^{k+1} - x^*\rangle] \\ &\quad + \epsilon[f(x^{k+1}) - f(x^*)],\end{aligned}$$

where the inequality results from setting $x = x^*$ in (9). On the other hand, if we set $x = x^{k+1}$ in (1) and combine the resulting inequality with the preceding, we obtain

$$\begin{aligned}\Lambda^*(x^k) - \Lambda^*(x^{k+1}) &\geq \frac{b}{2}\|x^k - x^{k+1}\|^2 \\ &\quad +\epsilon[\langle F(x^k) - F(x^*) + L(x^{k+1}) - L(x^k), x^{k+1} - x^*\rangle] \\ &\quad +\epsilon[\langle M(x^{k+1} - x^k), x^{k+1} - x^*\rangle \\ &\quad +\frac{1}{2}(\langle M(x^k - x^*), x^k - x^*\rangle - \langle M(x^{k+1} - x^*), x^{k+1} - x^*\rangle)] \\ &= A_1 + A_2 + A_3\end{aligned}$$

with

$$A_1 = \frac{b}{2}\|x^k - x^{k+1}\|^2,$$

$$\begin{aligned}A_2 &= \epsilon[\langle (F-L)(x^k) - (F-L)(x^*), x^{k+1} - x^*\rangle + \langle L(x^{k+1}) - L(x^*), x^{k+1} - x^*\rangle] \\ &\geq \epsilon[\langle (F-L)(x^k) - (F-L)(x^*), x^k - x^*\rangle \\ &\quad +\langle (F-L)(x^k) - (F-L)(x^*), x^{k+1} - x^k\rangle] \\ &\geq \epsilon[\gamma\|(F-L)(x^k) - (F-L)(x^*)\|^2 \\ &\quad +\langle (F-L)(x^k) - (F-L)(x^*), x^{k+1} - x^k\rangle] \\ &\geq -\frac{\epsilon}{4\gamma}\|x^{k+1} - x^k\|^2,\end{aligned}$$

and

$$A_3 = \frac{\epsilon}{2}\langle M(x^k - x^{k+1}), x^k - x^{k+1}\rangle.$$

Next

$$\begin{aligned}A_1 + A_3 &= \frac{b}{2}\|x^k - x^{k+1}\|^2 + \frac{\epsilon}{2}\langle M(x^k - x^{k+1}), x^k - x^{k+1}\rangle \\ &= \frac{b}{2}\|x^k - x^{k+1}\|_G^2 \\ &\geq \frac{b}{2}\lambda_{\min}(G)\|x^k - x^{k+1}\|^2\end{aligned}$$

and

$$\Lambda^*(x^k) - \Lambda^*(x^{k+1}) \geq \frac{1}{2}\left[b\lambda_{\min}(G) - \frac{\epsilon}{2\gamma}\right]\|x^k - x^{k+1}\|^2.$$

As in the preceding proofs, the sequence $\{\Lambda^*(x^k)\}$ is nonnegative, strictly decreasing (unless $x^{k+1} = x^k$), and must therefore converge. The inequality $\Lambda^*(x^k) \geq \frac{b}{2}\|x^k - x^*\|_G^2$ holds for all solutions x^* to VIP and implies that (i) the sequence $\{x^k\}$ is bounded and (ii) the sequence $\|x^{k+1} - x^k\|$ converges to zero. The proof of convergence of the entire sequence to a unique solution uses the same arguments as the proof of Theorem 3.2. □

4.2. Decomposition. The preceding result is important. Indeed, it allows the decomposition of monotone, but not necessarily co-coercive, variational inequalities. In particular, assume that F can be expressed as the sum of a co-coercive mapping F_1 and a monotone mapping F_2. If we set $\Gamma_1 = F_2$ in the modified auxiliary problem method, x^{k+1} is then characterized as the solution to the auxiliary variational inequality

$$\langle \epsilon(F_1(x^k) + F_2(x^{k+1})) + \Gamma_2(x^{k+1}) - \Gamma_2(x^k), x - x^{k+1} \rangle + \epsilon(f(x) - f(x^{k+1})) \geq 0$$

for all x in C. If F_2 happens to be symmetric or separable, the previous auxiliary problem is easier to solve than the original variational inequality.

Since any monotone mapping can be expressed as the sum of the zero mapping (which is trivially co-coercive) and itself, we recover as a special instance the general proximal variational inequality

$$\langle \epsilon F(x^{k+1}) + \Gamma_2(x^{k+1}) - \Gamma_2(x^k), x - x^{k+1} \rangle + \epsilon(f(x) - f(x^{k+1})) \geq 0 \quad \forall x \in C.$$

If, more specifically, $f(x) = 0$ and we take $\Gamma_2(x) = x$, then we recover Martinet's regularization method [9] as a special case of our general method.

4.3. Relation to the asymmetric projection algorithm. Let $f = 0$; let D be an $n \times n$ positive definite matrix; and consider the algorithm where, at iteration k, the original variational inequality is approximated by the affine, but not necessarily symmetric, variational inequality

$$\langle F(x^k) + D(x^{k+1} - x^k), x - x^{k+1} \rangle \geq 0 \quad \forall x \in C. \tag{12}$$

Tseng [12] showed that the convergence of this algorithmic scheme followed from general convergence conditions given by Gabay for a forward–backward splitting algorithm [5]. This "asymmetric projection" method is indeed a special instance of our general algorithm if one takes $M = 0$, $L(x) = [(D - D^t)/2]x$, and $h'(x) = [(D + D^t)/2]x$. Dividing by ϵ, the auxiliary variational inequality can be written as

$$\left\langle F(x^k) + \left(\frac{D - D^t}{2} + \frac{D + D^t}{2\epsilon} \right) (x^{k+1} - x^k), x - x^{k+1} \right\rangle \geq 0 \quad \forall x \in C \tag{13}$$

and Tseng's asymmetric projection method is obtained for $\epsilon = 1$.

Now denote by $\bar{D}$ the symmetric part of the matrix D. From Proposition 2.2, the mapping $F - L$ is co-coercive (with modulus γ) if and only if the mapping $\tilde{F} - \tilde{L}$ is co-coercive (with modulus $\tilde{\gamma}$), where $\tilde{L} = \bar{D}^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, $\tilde{F} = \bar{D}^{-\frac{1}{2}} F D^{-\frac{1}{2}}$, and $\tilde{\gamma} = \gamma / \|\bar{D}^{-\frac{1}{2}}\|^2$. Our convergence condition can now be expressed as $0 < \epsilon \leq 2\tilde{\gamma}$. This reduces to Tseng's condition $2\tilde{\gamma} \geq 1$ when $\epsilon = 1$.

4.4. Application to the traffic equilibrium problem. Let A be a matrix representing the arc-path incidence relationships over a multicommodity network. In this case, the feasible set C assumes the very simple form

$$\begin{aligned} \sum_p x_{pc} &= d_c, \\ x_{pc} &\geq 0, \end{aligned} \tag{14}$$

where p is a path index, c a commodity index, and d_c the demand for commodity c.

Let F be a monotone cost mapping from R^m into R^m. Bertsekas and Gafni [3] proposed the projection iteration

$$x^{k+1} = \mathrm{Proj}_D[x^k - \epsilon D^{-1} A^t F(Ax^k)] \tag{15}$$

for solving network equilibrium problems modeled as the variational inequality

$$\langle A^t F(Ax^*), x - x^* \rangle \geq 0 \quad \forall x \in C, \tag{16}$$

where $\epsilon > 0$; D is a positive definite, symmetric matrix; and Proj_D denotes the projection operator over the set C with respect to the norm induced by the matrix D. Under the assumptions that C be a polyhedral set, F be strongly monotone and Lipschitz continuous on C, and ϵ be sufficiently small, they proved the convergence of the previous projection algorithm. Their method clearly fits into our general framework, and convergence follows directly from the co-coercivity of the mapping $A^T F(Ax)$, which in general is not strongly monotone. In our analysis, the set C is only required to be a general convex set, not necessarily a polyhedron.

5. Partially co-coercive mappings and systems of variational inequalities. In this section we consider the system of variational inequalities

$$\langle F_1(x^*, y^*), x - x^* \rangle \geq 0 \quad \forall x \in C_x, \tag{17}$$

$$\langle F_2(x^*, y^*), y - y^* \rangle \geq 0 \quad \forall y \in C_y, \tag{18}$$

where C_x (respectively, C_y) is a nonempty, closed convex subset of R^n (respectively, R^m) and F_1, F_2 are continuous mappings from $C_x \times C_y$ into R^n and R^m, respectively. Furthermore, assume that $F_1(x, y)$ is strongly monotone on C_x with respect to the variable x, and denote by α_y its modulus.

DEFINITION 5.1. *The mapping $F(x, y) = (F_1(x, y), F_2(x, y))$ is partially co-coercive if for any $x_1, x_2 \in C_x$ and any $y_1, y_2 \in C_y$, there exists a constant γ such that*

$$\begin{aligned} &\langle F_1(x_1, y_1) - F_1(x_2, y_2), x_1 - x_2 \rangle + \langle F_2(x_1, y_1) - F_2(x_2, y_2), y_1 - y_2 \rangle \\ &\qquad \geq \gamma \| F_2(x_1, y_1) - F_2(x_2, y_2) \|^2. \end{aligned} \tag{19}$$

Consider the algorithm based on the iteration

$$\langle F_1(x^{k+1}, y^k), x - x^{k+1} \rangle \geq 0 \quad \forall x \in C_x, \tag{20}$$

$$\langle \epsilon F_2(x^{k+1}, y^k) + h'(y^{k+1}) - h'(y^k), y - y^{k+1} \rangle \geq 0 \quad \forall y \in C_y, \tag{21}$$

where the gradient h' is strongly monotone (with modulus b) over C_y and Lipschitz continuous over C_y with constant L.

THEOREM 5.2. *Let F_1 be strongly monotone on C_x with respect to the variable x with modulus α_y, $F = (F_1, F_2)$ be partially co-coercive with modulus γ, and either* (i) *α_y is independent of y or* (ii) *$F_1(x, y)$ is Lipschitz continuous in y and its Lipschitz constant is independent of x. Then the sequence $\{(x^k, y^k)\}$ generated by the algorithm converges to a solution of the system* (17), (18) *whenever $0 < \epsilon < 2b\gamma$.*

Proof. Let (x^*, y^*) be any fixed solution of the variational system (17), (18) and consider the function

$$\Lambda^*(y) = h(y^*) - h(y) - \langle h'(y), y^* - y \rangle. \tag{22}$$

We have

$$\begin{aligned}\Lambda^*(y^k) - \Lambda^*(y^{k+1}) &= h(y^{k+1}) - h(y^k) - \langle h'(y^k), y^{k+1} - y^k\rangle \\ &\quad + \langle h'(y^{k+1}) - h'(y^k), y^* - y^{k+1}\rangle \\ &\geq \frac{b}{2}\|y^k - y^{k+1}\|^2 + \epsilon\langle F_2(x^{k+1}, y^k), y^{k+1} - y^*\rangle.\end{aligned}$$

If we let $y = y^{k+1}$ in (18) and combine the corresponding inequality with the preceeding, the result is

$$\begin{aligned}\Lambda^*(y^k) - \Lambda^*(y^{k+1}) &\geq \frac{b}{2}\|y^k - y^{k+1}\|^2 + \epsilon\langle F_2(x^{k+1}, y^k) - F_2(x^*, y^*), y^{k+1} - y^*\rangle \\ &= \frac{b}{2}\|y^k - y^{k+1}\|^2 + T.\end{aligned}$$

Now

$$\begin{aligned}T &= \epsilon\langle F_2(x^{k+1}, y^k) - F_2(x^*, y^*), y^{k+1} - y^*\rangle \\ &= \epsilon\langle F_2(x^{k+1}, y^k) - F_2(x^*, y^*), y^k - y^*\rangle \\ &\quad + \epsilon\langle F_2(x^{k+1}, y^k) - F_2(x^*, y^*), y^{k+1} - y^k\rangle.\end{aligned}$$

Setting $x = x^{k+1}$ in (17) and $x = x^*$ in (20) and using the partial co-coercivity of F, we get

$$\begin{aligned}T &\geq \epsilon[\gamma\|F_2(x^{k+1}, y^k) - F_2(x^*, y^*)\|^2 + \langle F_2(x^{k+1}, y^k) - F_2(x^*, y^*), y^{k+1} - y^k\rangle] \\ &\geq -\frac{\epsilon}{4\gamma}\|y^{k+1} - y^k\|^2\end{aligned}$$

and

$$\Lambda^*(y^k) - \Lambda^*(y^{k+1}) \geq \frac{1}{2}\left(b - \frac{\epsilon}{2\gamma}\right)\|y^{k+1} - y^k\|^2.$$

Since $\epsilon < 2b\gamma$, $\Lambda^*(y^k) - \Lambda^*(y^{k+1})$ is positive and the sequence $\{\Lambda^*(y^k)\}$, which is strictly decreasing unless $y^k = y^{k+1}$, must converge to some number. Now, the fact that $\Lambda^*(y^k) \geq \frac{b}{2}\|y^k - y^*\|^2$ and $\{y^k\}$ is bounded imply that $\|y^{k+1} - y^k\|$ converges to zero. Thus there exists a subsequence $\{y^{k'}\}$ that converges to some limit point $\bar{y}$ of C_y.

Now let $x(y)$ satisfy the variational inequality

$$\langle F_1(x(y), y), x - x(y)\rangle \geq 0 \quad \forall x \in C_x \tag{23}$$

and let $\bar{x} = x(\bar{y})$. From our assumptions, the mapping $x(y)$ is continuous in y (see Harker and Pang [6], for instance). Since $y^{k'} \to \bar{y}$, it follows that $x^{k'+1} \to \bar{x}$ and $(\bar{x}, \bar{y})$ satisfies (17) with $(x^*, y^*) = (\bar{x}, \bar{y})$. On the other hand, since $\|y^k - y^{k+1}\|$ converges to zero, $(\bar{x}, \bar{y})$ satisfies (18). This implies that $(\bar{x}, \bar{y})$ is a solution of the system (17), (18).

The previous convergence analysis holds for any solution (x^*, y^*) of the system of variational inequalities. Taking $(x^*, y^*) = (\bar{x}, \bar{y})$, we can prove that the entire sequence $\{(x^k, y^k)\}$ converges to $(\bar{x}, \bar{y})$. Using the Lipschitz constant L of h', we can indeed write

$$\frac{b}{2}\|\bar{y} - y^k\|^2 \leq \Lambda^*(y^k) \leq \frac{1}{2}L\|\bar{y} - y^k\|^2.$$

This, together with the convergence of the sequence $y^{k'}$ to $\bar{y}$, implies that the entire sequence $\{y^k\}$ converges to $\bar{y}$. Then it follows from the continuity of $x(y)$ that the entire sequence $\{x^{k+1}\}$ converges to $\bar{x}$. □

Although the previous method can be applied to any asymmetric variational inequality satisfying the partial co-coercivity condition, it is more specifically suited to problems that are "partially" asymmetric, i.e., where the mapping F_1 is a gradient mapping, in which case (20) reduces to an optimization problem.

6. A primal-dual framework. In this section we consider the variational inequality

$$\langle F(x^*), x - x^* \rangle \geq 0 \quad \forall x \in E = C \cap \{x \in R^n | B(x) \leq 0\}, \tag{24}$$

where C is a nonempty closed convex subset of R^n, F is strongly monotone (with modulus a) on E, and $B(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ is a constraint mapping explicitly defined by the convex, Lipschitz continuous, and continuously differentiable functions f_i, $i = 1, \ldots, m$. Assume that there exists $x_0 \in C$ such that $f_i(x_0) < 0$, $i = 1, \ldots, m$ (Slater's constraint qualification). Then the variational inequality (24) is equivalent to the Kuhn–Tucker-like system

$$\langle F(x^*) + (\nabla B(x^*))^t y^*, x - x^* \rangle \geq 0 \quad \forall x \in C, \tag{25}$$

$$\langle -B(x^*), y - y^* \rangle \geq 0 \quad \forall y \geq 0. \tag{26}$$

Let $F_1(x, y) = F(x) + (\nabla B(x))^t y$, $F_2(x, y) = -B(x)$, and $\Gamma = (F_1, F_2)$. We have, for y_1, y_2 nonnegative and x_1, x_2 in C,

$$\begin{aligned} S &\stackrel{\text{def}}{=} \langle F_1(x_1, y_1) - F_1(x_2, y_2), x_1 - x_2 \rangle + \langle F_2(x_1, y_1) - F_2(x_2, y_2), y_1 - y_2 \rangle \\ &= \langle F(x_1) - F(x_2), x_1 - x_2 \rangle + \langle (\nabla B(x_1))^t y_1 - (\nabla B(x_2))^t y_2, x_1 - x_2 \rangle \\ &\quad - \langle B(x_1) - B(x_2), y_1 - y_2 \rangle. \end{aligned}$$

Since y_1 and y_2 are nonnegative and the functions f_i are convex, the functions $x \to \langle B(x), y_1 \rangle$ and $x \to \langle B(x), y_2 \rangle$ are convex. Therefore we can write

$$\langle (\nabla B(x_1))^t y_1 - (\nabla B(x_2))^t y_2, x_1 - x_2 \rangle \geq -\langle B(x_1) - B(x_2), y_1 - y_2 \rangle \tag{27}$$

and

$$S \geq \langle F(x_1) - F(x_2), x_1 - x_2 \rangle \geq a \|x_1 - x_2\|^2.$$

Denoting by L_B the Lipschitz constant of B, we obtain

$$S \geq (a/L_B^2) \|B(x_1) - B(x_2)\|^2 = (a/L_B^2) \|F_2(x_1, y_1) - F_2(x_2, y_2)\|^2,$$

which implies that the mapping Γ is partially co-coercive.

We can apply the iterate (20), (21) of the previous section to the system (25), (26) to obtain the primal-dual algorithm

$$\langle F(x^{k+1}) + (\nabla B(x^{k+1}))^t y^k, x - x^{k+1} \rangle \geq 0 \quad \forall x \in C_x, \tag{28}$$

$$\langle -\epsilon B(x^{k+1}) + h'(y^{k+1}) - h'(y^k), y - y^{k+1} \rangle \geq 0 \quad \forall y \geq 0. \tag{29}$$

In particular, if we take $h(y) = \|y\|^2/2$, the solution of the auxiliary variational inequality is given by the closed-form expression

$$y^{k+1} = \max\{0, y^k + \epsilon B(x^{k+1})\}, \tag{30}$$

and our algorithm reduces to a Uzawa-type method (see Bensoussan, Lions, and Temam [2]).

If F and ∇B are separable our primal-dual algorithm lends itself naturally to decomposition. Otherwise, decomposition can still be achieved by applying the auxiliary problem technique to the pair of variational inequalities (25), (26).

REFERENCES

[1] A. Auslender, *Optimisation: Méthodes Numériques*, Masson, Paris, 1973.

[2] A. Bensoussan, J. L. Lions, and R. Temam, *Sur les méthodes de décomposition et de coordination et applications*, Sur les Méthodes Numériques en Sciences Physiques et Économiques, J. L. Lions and G. I. Marchouk, eds., Dunod, Paris, 1974.

[3] P. Bertsekas and M. Gafni, *Projection methods for variational inequalities with application to the traffic assignment problem*, Math. Programming Stud., 17 (1982), pp. 139–159.

[4] G. Cohen, *Auxiliary problem principle extended to variational inequalities*, J. Optim. Theory Appl., 59 (1988), pp. 325–333.

[5] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, M. Fortin and R. Glowinski, eds., North-Holland, Amsterdam, 1983.

[6] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequalities and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[7] Z. Luo and P. Tseng, *A decomposition property for a class of square matrices*, Appl. Math. Lett., 4 (1991), pp. 67–69.

[8] P. Marcotte and J. H. Wu, *On the convergence of projection methods: Application to the decomposition of affine variational inequalities*, J. Optim. Theory Appl., 85 (1995), pp. 347–362.

[9] B. Martinet, *Régularisation d'inéquations variationnelles par approximations successives*, Revue française d'informatique et de recherche opérationnelle, R-3 (1970), pp. 154–159.

[10] M. A. Mataoui, *Contributions à la Décomposition et à l'Agrégation des Problèmes Variationnels*, Thèse de Doctorat, École des Mines de Paris, Paris, 1990.

[11] J. M. Ortega and W. O. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[12] P. Tseng, *Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming*, Math. Programming, 48 (1990), pp. 249–264.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 727–747, August 1996
© 1996 Society for Industrial and Applied Mathematics
012

COMPUTING THE MOORE–PENROSE INVERSE FOR THE COVARIANCE MATRIX IN CONSTRAINED NONLINEAR ESTIMATION*

WOLFGANG M. HARTMANN† AND ROBERT E. HARTWIG‡

Abstract. A new algorithm is developed to compute the Moore–Penrose inverse of the Lagrangian matrix which is used to compute the covariance matrix of parameter estimates in constrained nonlinear optimization. The algorithm takes into account the bordered structure of the Lagrangian matrix and that the projected Hessian is available at no cost at the end of the optimization process. For many applications and especially for an increasing number of active constraints at the optimum, the new algorithm will be considerably more efficient than the traditional one.

Key words. covariance matrix, constrained nonlinear estimation, MP inverse of Lagrangian matrix

AMS subject classification. 62G15

1. Introduction.

The covariance matrix of parameter estimates in nonlinear optimization is widely used in statistics. For example, the square roots of the diagonal elements are used as *approximate standard errors* of parameters in maximum likelihood estimation [3]. For *unconstrained nonlinear estimation* there are several ways to compute the covariance matrix. The most important is

$$\mathbf{COV} = \sigma^2 \mathcal{G}^{-1}, \tag{1}$$

where $\mathcal{G}$ is (an approximation of) the Hessian matrix of the objective (likelihood) function, evaluated at the optimal parameter estimates x^*, and σ^2 is a scalar expressing the error variance which in practice may be defined in many ways. The second-order necessary condition for a local minimizing point x^* requires $\mathcal{G}$ to be positive semidefinite and for a maximizing point to be negative semidefinite. Furthermore, if $\mathcal{G}$ is nonsingular then the local optimum is *isolated*; i.e., x^* is the only solution in a neighbourhood of x^*. In many applications (e.g., neural networks) the Hessian $\mathcal{G}$ is rank deficient at the optimum x^* and the inverse must be replaced by some type of generalized inverse, preferably the Moore–Penrose inverse $\mathcal{G}^\dagger$.

For *constrained nonlinear estimation*, a covariance matrix can be computed as

$$\mathbf{COV} = \sigma^2 \mathbf{\Phi} \mathcal{G} \mathbf{\Phi}, \tag{2}$$

where $\mathbf{\Phi}$ is the (1,1) block of the (generalized) inverse of the bordered matrix $\mathbf{M}$ [3, p. 38]:

$$\mathbf{M}^\dagger = \begin{bmatrix} \mathcal{G} & \mathcal{C} \\ \mathcal{C}^T & \mathbf{0} \end{bmatrix}^\dagger = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Psi} \\ \mathbf{\Psi}^T & \mathbf{\Omega} \end{bmatrix} \in \mathcal{R}^{(n+m)\times(n+m)}. \tag{3}$$

Here $\mathcal{G}$ is (an approximation of) the $n \times n$ Hessian matrix of the Lagrange function and $\mathcal{C}$ is the $n \times m$ matrix containing the coefficients of m linear(ized) constraints that are active at the optimizer x^*. Both the Hessian $\mathcal{G}$ and the constraint matrix $\mathcal{C}$ are

* Received by the editors July 14, 1994; accepted for publication (in revised form) January 17, 1995.

† SAS Institute, Inc., SAS Campus Drive, Cary, NC 27513 (saswmh@unx.sas.com).

‡ Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (hartwig@math.ncsu.edu).

evaluated at the optimal parameter estimates x^*. For linear constraints the Hessian of the Lagrange function and the Hessian of the objective function are identical. Most textbooks on nonlinear estimation assume that

$$\det(\mathbf{M}) \neq 0, \quad \det(\mathcal{G}) \neq 0, \quad \det(\mathcal{C}^T\mathcal{G}^{-1}\mathcal{C}) \neq 0 \tag{4}$$

and obtain

$$\begin{aligned} \mathbf{\Phi} &= \mathcal{G}^{-1} - \mathcal{G}^{-1}\mathcal{C}(\mathcal{C}^T\mathcal{G}^{-1}\mathcal{C})^{-1}\mathcal{C}^T\mathcal{G}^{-1} \\ &= \mathbf{Z}\mathbf{\Gamma}^{-1}\mathbf{Z}^T \in \mathcal{R}^{n\times n}, \end{aligned} \tag{5}$$

where Γ is the projected Hessian matrix (see below) and $\mathbf{Z}$ is an orthogonal basis matrix for the null space of the active constraints in $\mathcal{C}^T$ (see, e.g., [3, p. 38], [6, p. 237], [7, p. 245]). For constrained nonlinear estimation we can no longer assume that the Hessian $\mathcal{G}$ is positive (for minimization) or negative (for maximization) (semi-)definite, and the Hessian may be singular and indefinite even for an *isolated* optimizer x^*. If at least one of these nonsingularity assumptions is violated, most optimization software computes a generalized (but not the Moore–Penrose) inverse of $\mathbf{M}$ by successive "sweeping" (see [13] and [18]) of the bordered matrix as long as nonzero diagonal pivots are left. Afterward the rows and columns corresponding to zero pivots are zeroed out. For various statistical applications the Moore–Penrose inverse $\mathbf{M}^\dagger$ can be more useful than just an inner inverse $\mathbf{M}^-$; see, e.g., [17].

The algorithms that we develop here should be considered only for the case where the Hessian $\mathcal{G}$ is indefinite and the projected Hessian $\mathbf{\Gamma}$ is singular. If the Hessian is positive semidefinite, we simply obtain

$$\mathbf{COV} = \sigma^2\mathbf{\Phi}, \quad \text{with} \quad \mathbf{\Phi} = \mathbf{Z}\mathbf{\Gamma}^\dagger\mathbf{Z}^T. \tag{6}$$

We assume familiarity with the notation of Ben-Israel and Greville [1]. As always, ranges and row spaces are denoted by $R(.)$ and $RS(.)$, respectively, and all matrices will be real.

2. Developing an efficient formula for Φ. In this section we aim to develop an efficient way to compute the Moore–Penrose inverse of $\mathbf{M}$ for the general case where all three nonsingularity assumptions $\det(\mathbf{M}) \neq 0$, $\det(\mathcal{G}) \neq 0$, and $\det(\mathcal{C}^T\mathcal{G}^{-1}\mathcal{C}) \neq 0$ may be violated. In particular, we want to exploit the partitioning of $\mathbf{M}$ and the properties of optimal solutions usually given by nonlinear optimization software. The second-order necessary condition for the optimality of parameter estimates x^* requires the *projected Hessian* $\mathbf{\Gamma}$ of the Lagrange function to be positive (semi-)definite [6, p. 210]. In the following we will assume that the *projected Hessian* $\mathbf{\Gamma}$ is computed using the QR decomposition of m linearly independent constraints $\mathcal{C} \in \mathcal{R}^{n\times m}$, $\operatorname{rank}(\mathcal{C}) = m \leq n$, which are active at x^*,

$$\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{t\times t}, \tag{7}$$

$$\text{where} \quad \mathbf{Q}^T\mathcal{C} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}, \quad \text{with} \quad \mathbf{Q} = [\mathbf{Y} \quad \mathbf{Z}], \quad t = n - m$$

and $\mathbf{Q}^{-1} = \mathbf{Q}^T$, $\mathbf{Y} \in \mathcal{R}^{n\times m}$, $\mathbf{Z} \in \mathcal{R}^{n\times t}$, and $\mathbf{R} \in \mathcal{R}^{m\times m}$ is invertible upper triangular. Using the up- and downdate algorithm of Gill et al. [8] during the optimization process, we obtain the matrices $\mathbf{Y}$ and $\mathbf{Z}$ explicitly rather than in the product form

of Householder transformations. These are reasonable assumptions which are met by many nonlinear optimization programs (e.g., [8, 9, 10, 14, 15]). However, the algorithm we describe can be modified using other elimination methods; see the discussion in [6, pp. 230–236].

The three assumptions $\det(\mathbf{M}) \neq 0$, $\det(\mathcal{G}) \neq 0$, and $\det(\mathcal{C}^T\mathcal{G}^{-1}\mathcal{C}) \neq 0$ can be combined into only one assumption using a well-known fact of Lagrangian methods: If the projected Hessian $\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z}$ is positive definite, one can obtain the (1,1) block $\mathbf{\Phi}$ of $\mathbf{M}^{-1}$ [6, p. 237] as

$$\mathbf{\Phi} = \mathbf{Z}(\mathbf{Z}^T\mathcal{G}\mathbf{Z})^{-1}\mathbf{Z}^T = \mathbf{Z}\mathbf{\Gamma}^{-1}\mathbf{Z}^T. \tag{8}$$

If the Hessian $\mathcal{G}$ is positive semidefinite, say $\mathcal{G} = \mathbf{V}^T\mathbf{V}$, we obtain

$$\mathbf{\Phi} = \mathbf{Z}\mathbf{\Gamma}^{\dagger}\mathbf{Z}^T \quad \text{and} \quad \mathbf{COV} = \sigma^2\mathbf{\Phi}. \tag{9}$$

When the Hessian $\mathcal{G}$ is indefinite and the projected Hessian $\mathbf{\Gamma}$ is singular we can use the following theorem (the notation of matrices in the theorem is slightly changed).

THEOREM 2.1. *Let*

$$\mathbf{N} = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{B} & \mathbf{D} \end{bmatrix}$$

with $\mathbf{C}\mathbf{C}^{\dagger} = \mathbf{I}$, $\mathbf{B}^{\dagger}\mathbf{B} = \mathbf{I}$, *and* $RS(\mathbf{C}) \cap RS(\mathbf{D}) = (\mathbf{0})$ *and* $R(\mathbf{B}) \cap R(\mathbf{D}) = (\mathbf{0})$. *Then*

$$\mathbf{N}^{\dagger} = \begin{bmatrix} \mathbf{0} & \mathbf{X}^{\dagger} \\ \mathbf{Y}^{\dagger} & -\mathbf{Y}^{\dagger}\mathbf{A}\mathbf{X}^{\dagger} + (\mathbf{I} - \mathbf{Y}^{\dagger}\mathbf{C})\mathbf{D}^{\dagger}(\mathbf{I} - \mathbf{B}\mathbf{X}^{\dagger}) \end{bmatrix}, \tag{10}$$

where $\mathbf{X} = (\mathbf{I} - \mathbf{D}\mathbf{D}^{\dagger})\mathbf{B}$ *and* $\mathbf{Y} = \mathbf{C}(\mathbf{I} - \mathbf{D}^{\dagger}\mathbf{D})$. *In particular, if* $\mathbf{N}^T = \mathbf{N}$ *we may take* $\mathbf{B} = \mathbf{C}^T$ *and* $\mathbf{X} = \mathbf{Y}^T$.

Proof. We denote the block matrix in (10) by $\mathbf{T}$ and shall show that $\mathbf{T}$ satisfies the four Moore–Penrose conditions.

- First, note that

$$RS(\mathbf{C}) \cap RS(\mathbf{D}) = (\mathbf{0}) \Leftrightarrow R(\mathbf{C}(\mathbf{I} - \mathbf{D}^{\dagger}\mathbf{D})) = R(\mathbf{C})$$

and

$$R(\mathbf{B}) \cap R(\mathbf{D}) = (\mathbf{0}) \Leftrightarrow RS((\mathbf{I} - \mathbf{D}\mathbf{D}^{\dagger})\mathbf{B}) = RS(\mathbf{B}).$$

Hence, by assumption, $\mathbf{Y}\mathbf{Y}^{\dagger} = \mathbf{I}$ and $\mathbf{X}^{\dagger}\mathbf{X} = \mathbf{I}$.

- Therefore, we verify the following: $\mathbf{Y}\mathbf{D}^{\dagger} = \mathbf{0} \Rightarrow \mathbf{D}\mathbf{Y}^{\dagger} = \mathbf{0}$, $\mathbf{C}\mathbf{D}^{\dagger}\mathbf{D} = \mathbf{C} - \mathbf{Y}$, $\mathbf{C}\mathbf{Y}^{\dagger} = (\mathbf{C} - \mathbf{C}\mathbf{D}^{\dagger}\mathbf{D})\mathbf{Y}^{\dagger} = \mathbf{Y}\mathbf{Y}^{\dagger} = \mathbf{I}$, $\mathbf{D}\mathbf{D}^{\dagger}\mathbf{B} = \mathbf{B} - \mathbf{X}$, $\mathbf{C}\mathbf{Y}^{\dagger}\mathbf{C} = \mathbf{C}$, $\mathbf{X}^{\dagger}\mathbf{B} = \mathbf{X}^{\dagger}\mathbf{X} = \mathbf{I}$, $\mathbf{B}\mathbf{X}^{\dagger}\mathbf{B} = \mathbf{B}$. Also, $\mathbf{D}^{\dagger}\mathbf{X} = \mathbf{0} \Rightarrow \mathbf{X}^{\dagger}\mathbf{D} = \mathbf{0}$.
- MP condition (3): $\mathbf{N}\mathbf{T} = (\mathbf{N}\mathbf{T})^T$:

$$\mathbf{N}\mathbf{T} = \begin{bmatrix} \mathbf{C}\mathbf{Y}^{\dagger} & \mathbf{A}\mathbf{X}^{\dagger} - \mathbf{C}\mathbf{Y}^{\dagger}\mathbf{A}\mathbf{X}^{\dagger} + \mathbf{C}(\mathbf{I} - \mathbf{Y}^{\dagger}\mathbf{C})\mathbf{D}^{\dagger}(\mathbf{I} - \mathbf{B}\mathbf{X}^{\dagger}) \\ \mathbf{D}\mathbf{Y}^{\dagger} & \mathbf{B}\mathbf{X}^{\dagger} - \mathbf{D}\mathbf{Y}^{\dagger}\mathbf{A}\mathbf{X}^{\dagger} + \mathbf{D}(\mathbf{I} - \mathbf{Y}^{\dagger}\mathbf{C})\mathbf{D}^{\dagger}(\mathbf{I} - \mathbf{B}\mathbf{X}^{\dagger}) \end{bmatrix}$$

and since $\mathbf{D}(\mathbf{I} - \mathbf{Y}^{\dagger}\mathbf{C})\mathbf{D}^{\dagger}(\mathbf{I} - \mathbf{B}\mathbf{X}^{\dagger}) = \mathbf{D}\mathbf{D}^{\dagger} - \mathbf{B}\mathbf{X}^{\dagger} + \mathbf{X}\mathbf{X}^{\dagger}$ we obtain

$$\mathbf{N}\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{D}^{\dagger} + \mathbf{X}\mathbf{X}^{\dagger} \end{bmatrix} = (\mathbf{N}\mathbf{T})^T.$$

- MP condition (4): $\mathbf{TN} = (\mathbf{TN})^T$
  It follows at once by left–right symmetry that

$$\mathbf{TN} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^\dagger\mathbf{D} + \mathbf{Y}^\dagger\mathbf{Y} \end{bmatrix} = (\mathbf{TN})^T.$$

- MP condition (1): $\mathbf{NTN} = \mathbf{N}$:

$$\begin{aligned} \mathbf{NTN} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{DD}^\dagger + \mathbf{XX}^\dagger \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{B} & \mathbf{D} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{DD}^\dagger\mathbf{B} + \mathbf{XX}^\dagger\mathbf{B} & \mathbf{D} + \mathbf{XX}^\dagger\mathbf{D} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{B} & \mathbf{D} \end{bmatrix}. \end{aligned}$$

- MP condition (2): $\mathbf{TNT} = \mathbf{T}$:

$$\begin{aligned} \mathbf{TNT} &= \begin{bmatrix} \mathbf{0} & \mathbf{X}^\dagger \\ \mathbf{Y}^\dagger & -\mathbf{Y}^\dagger\mathbf{AX}^\dagger + (\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})\mathbf{D}^\dagger(\mathbf{I} - \mathbf{BX}^\dagger) \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{DD}^\dagger + \mathbf{XX}^\dagger \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{X}^\dagger(\mathbf{DD}^\dagger + \mathbf{XX}^\dagger) \\ \mathbf{Y}^\dagger & -\mathbf{Y}^\dagger\mathbf{AX}^\dagger(\mathbf{DD}^\dagger + \mathbf{XX}^\dagger) + (\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})\mathbf{D}^\dagger(\mathbf{I} - \mathbf{BX}^\dagger)(\mathbf{DD}^\dagger + \mathbf{XX}^\dagger) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{X}^\dagger \\ \mathbf{Y}^\dagger & -\mathbf{Y}^\dagger\mathbf{AX}^\dagger + (\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})\mathbf{D}^\dagger(\mathbf{I} - \mathbf{BX}^\dagger) \end{bmatrix} \end{aligned}$$

since

$$\begin{aligned} &(\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})\mathbf{D}^\dagger(\mathbf{I} - \mathbf{BX}^\dagger)(\mathbf{DD}^\dagger + \mathbf{XX}^\dagger) \\ &= (\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})\mathbf{D}^\dagger(\mathbf{DD}^\dagger + \mathbf{XX}^\dagger - \mathbf{BX}^\dagger\mathbf{DD}^\dagger - \mathbf{BX}^\dagger) \\ &= (\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})(\mathbf{D}^\dagger - \mathbf{D}^\dagger\mathbf{XX}^\dagger - \mathbf{D}^\dagger\mathbf{BX}^\dagger) \\ &= (\mathbf{I} - \mathbf{Y}^\dagger\mathbf{C})\mathbf{D}^\dagger(\mathbf{I} - \mathbf{BX}^\dagger). \qquad \square \end{aligned}$$

We note in passing that block formula (10) with $\mathbf{A} = \mathbf{0}$ is essentially equivalent to the following "horizontal" result.

LEMMA. *Suppose* $\mathbf{E}$ *and* $\mathbf{F}$ *are projections such that* $RS(\mathbf{F}) \cap RS(\mathbf{D}) = (\mathbf{0})$ *and* $R(\mathbf{E}) \cap R(\mathbf{D}) = (\mathbf{0})$. *Then*

$$[(\mathbf{I} - \mathbf{E})\mathbf{D}(\mathbf{I} - \mathbf{F})]^\dagger = \{\mathbf{I} - [\mathbf{F}(\mathbf{I} - \mathbf{D}^\dagger\mathbf{D})]^\dagger\}\mathbf{D}^\dagger\{\mathbf{I} - [(\mathbf{I} - \mathbf{DD}^\dagger)\mathbf{E}]^\dagger\}. \tag{11}$$

Let us now use Theorem 2.1 to develop a formula for $\mathbf{\Phi}$ based on the partitioning of matrix $\mathbf{M}$. Our steps are the following.

1. Apply the QR decomposition of $\mathcal{C}$ to $\mathbf{M}$, which gives

$$\mathbf{N} = \begin{bmatrix} \mathbf{Y}^T & \mathbf{0} \\ \mathbf{Z}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathcal{G} & \mathcal{C} \\ \mathcal{C}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Y} & \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_m \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{E} & \mathbf{R} \\ \mathbf{E}^T & \mathbf{\Gamma} & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{12}$$

where $\mathbf{A} = \mathbf{Y}^T\mathcal{G}\mathbf{Y} \in \mathcal{R}^{m\times m}$, $\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{t\times t}$, $\mathbf{E} = \mathbf{Y}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{m\times t}$, and $\mathbf{R} = \mathbf{Y}^T\mathcal{C} \in \mathcal{R}^{m\times m}$ is invertible and upper triangular. It follows at once that

$$p = \operatorname{rank}(\mathbf{M}) = \operatorname{rank}(\mathbf{N}) = 2m + r, \quad \text{where} \quad r = \operatorname{rank}(\mathbf{\Gamma}). \tag{13}$$

Consequently, the three matrices $\mathbf{M}$, $\mathbf{N}$, and $\mathbf{\Gamma}$ have the same number $s = n - m - r = t - r$ of zero eigenvalues.

2. Apply Theorem 2.1 to $\mathbf{N}$:

$$\mathbf{N} = \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{E} & \mathbf{R} \\ \hline \mathbf{E}^T & \boldsymbol{\Gamma} & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} \end{array}\right] = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{B} & \mathbf{D} \end{bmatrix}$$

with $\mathbf{B}^T = [\mathbf{E} \quad \mathbf{R}] \in \mathcal{R}^{m\times n}$, and $\mathbf{D} = \left[\begin{smallmatrix} \boldsymbol{\Gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right] \in \mathcal{R}^{n\times n}$. This results in

$$\mathbf{N}^\dagger = \begin{bmatrix} \mathbf{0}_m & \mathbf{X}^\dagger \\ (\mathbf{X}^\dagger)^T & -(\mathbf{X}^\dagger)^T\mathbf{A}\mathbf{X}^\dagger + (\mathbf{I}_n - (\mathbf{X}^\dagger)^T\mathbf{B}^T)\mathbf{D}^\dagger(\mathbf{I}_n - \mathbf{B}\mathbf{X}^\dagger) \end{bmatrix}, \tag{14}$$

where $\mathbf{X} = (\mathbf{I} - \mathbf{D}\mathbf{D}^\dagger)\mathbf{B} \in \mathcal{R}^{n\times m}$. To obtain the $(1,1)$ block $\boldsymbol{\Phi}$ of $\mathbf{M}^\dagger$ in (3) we only need the (1,1) block of $\mathbf{N}^\dagger$ of size $n \times n$:

$$\begin{aligned} \boldsymbol{\Phi} &= [\mathbf{Y} \quad \mathbf{Z}] \begin{bmatrix} \mathbf{0}_m & \mathbf{X}_1^\dagger \\ (\mathbf{X}_1^\dagger)^T & -(\mathbf{X}_1^\dagger)^T\mathbf{A}\mathbf{X}_1^\dagger + (\mathbf{I}_t - (\mathbf{X}_1^\dagger)^T\mathbf{E})\boldsymbol{\Gamma}^\dagger(\mathbf{I}_t - \mathbf{E}^T\mathbf{X}_1^\dagger) \end{bmatrix} \begin{bmatrix} \mathbf{Y}^T \\ \mathbf{Z}^T \end{bmatrix} \\ &= \mathbf{Z}(\mathbf{X}_1^\dagger)^T\mathbf{Y}^T + \mathbf{Y}\mathbf{X}_1^\dagger\mathbf{Z}^T - \mathbf{Z}(\mathbf{X}_1^\dagger)^T\mathbf{A}\mathbf{X}_1^\dagger\mathbf{Z}^T + \mathbf{Z}(\mathbf{I} - (\mathbf{X}_1^\dagger)^T\mathbf{E})\boldsymbol{\Gamma}^\dagger(\mathbf{I} - \mathbf{E}^T\mathbf{X}_1^\dagger)\mathbf{Z}^T, \end{aligned} \tag{15}$$

where $\mathbf{X}_1^\dagger \in \mathcal{R}^{m\times t}$ contains the first t columns of $\mathbf{X}^\dagger$. Needless to say, this is the generalization of (8). Substituting expressions for $\boldsymbol{\Gamma}^\dagger$ and $\mathbf{X}_1^\dagger$ yields the desired result.

3. If the Hessian $\mathcal{G}$ is positive semidefinite, then $\mathbf{X}_1 = \mathbf{0}$ and we obtain as before

$$\boldsymbol{\Phi} = \mathbf{Z}\boldsymbol{\Gamma}^\dagger\mathbf{Z}^T \quad \text{and} \quad \mathbf{COV} = \sigma^2\boldsymbol{\Phi}.$$

If $\mathcal{G}$ is indefinite and $\boldsymbol{\Gamma}$ is singular we must develop an efficient algorithm to evaluate $\boldsymbol{\Phi}$ in (15). In Algorithm A2 in the next section we will show that we obtain neither $\boldsymbol{\Gamma}^\dagger$ nor $\mathbf{X}^\dagger$ explicitly. Indeed, let us close this section with a few comments on the computation of $\mathbf{X}^\dagger$. We recall that if

$$\begin{aligned} \mathbf{X} &= (\mathbf{I} - \mathbf{D}\mathbf{D}^\dagger)\mathbf{B} \\ &= \begin{bmatrix} \mathbf{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\dagger & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{E}^T \\ \mathbf{R}^T \end{bmatrix} = \begin{bmatrix} (\mathbf{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\dagger)\mathbf{E}^T \\ \mathbf{R}^T \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T \\ \mathbf{R}^T \end{bmatrix} \in \mathcal{R}^{n\times m} \end{aligned}$$

then

$$\mathbf{X}^\dagger = \mathbf{H}^{-1}[\mathbf{E}(\mathbf{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\dagger) \quad \mathbf{R}] = \mathbf{H}^{-1}[\mathbf{F} \quad \mathbf{R}] \in \mathcal{R}^{m\times n} \tag{16}$$

with

$$\mathbf{H} = \mathbf{R}\mathbf{R}^T + \mathbf{E}(\mathbf{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\dagger)\mathbf{E}^T = \mathbf{R}\mathbf{R}^T + \mathbf{F}\mathbf{F}^T, \tag{17}$$

where $\mathbf{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\dagger$ is idempotent. In the next section we will show that we do not need the inverse of $\mathbf{H}$ explicitly to obtain $\mathbf{X}^\dagger$. Moreover, since we are interested only in the upper left $n \times n$ part $\boldsymbol{\Phi}$ of $\mathbf{COV}$ we need only the first t columns, say $\mathbf{X}_1^\dagger$, of $\mathbf{X}^\dagger$, i.e.,

$$\mathbf{X}_1^\dagger = \mathbf{H}^{-1}\mathbf{E}(\mathbf{I} - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\dagger) = \mathbf{H}^{-1}\mathbf{F} \in \mathcal{R}^{m\times t}. \tag{18}$$

It should be noted that $\mathbf{X}_1$ vanishes exactly when $\operatorname{rank}(\boldsymbol{\Gamma}) = \operatorname{rank}(\mathcal{G}\mathbf{Z})$ or exactly when $\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi} = \boldsymbol{\Phi}$. For details see Appendix 2. Usually we do not have any information on the inertia of $\mathcal{G}$ in constrained optimization. Rather than computing the inertia of $\mathcal{G}$ we compute $\mathbf{X}_1^\dagger$ formally. If $\mathbf{X}_1^\dagger = \mathbf{0}$ then $\boldsymbol{\Phi} = \mathbf{Z}\boldsymbol{\Gamma}^\dagger\mathbf{Z}^T$.

3. Algorithms for computing matrix Φ via MP inversion.
Some unpartitioned approaches.

1. *Nonsingular case.* If the three assumptions

$$\det(\mathbf{M}) \neq 0, \quad \det(\mathcal{G}) \neq 0, \quad \det(\mathcal{C}^T\mathcal{G}^{-1}\mathcal{C}) \neq 0$$

are satisfied, one may use either

$$\mathbf{\Phi} = \mathcal{G}^{-1} - \mathcal{G}^{-1}\mathcal{C}(\mathcal{C}^T\mathcal{G}^{-1}\mathcal{C})^{-1}\mathcal{C}^T\mathcal{G}^{-1},$$

which is preferred in sparse implementations, or use

$$\mathbf{\Phi} = \mathbf{Z}(\mathbf{Z}^T\mathcal{G}\mathbf{Z})^{-1}\mathbf{Z}^T = \mathbf{Z}\mathbf{\Gamma}^{-1}\mathbf{Z}^T$$

in the dense case. In either case $\mathbf{COV} = \sigma^2\mathbf{\Phi}$. If one of the three assumptions is violated then one computes

- either an inner inverse $\mathbf{M}^-$ satisfying $\mathbf{M}\mathbf{M}^-\mathbf{M} = \mathbf{M}$
- or the Moore–Penrose inverse $\mathbf{M}^\dagger$ of the augmented matrix

$$\mathbf{M} = \begin{bmatrix} \mathcal{G} & \mathcal{C} \\ \mathcal{C}^T & \mathbf{0} \end{bmatrix}.$$

2. *Algorithm* A0: *Use the sweep operator.* An *inner inverse* of $\mathbf{M}$ is computed by repeatedly "sweeping" [13, 18] the augmented matrix as long as nonzero diagonal pivots are left. Afterward the rows and columns corresponding to zero pivots are zeroed out. We do not recommend this approach since the introduction of roundoff errors during the "sweep" process of ill-conditioned matrices makes the detection of rank deficiency a very difficult problem.

3. *Algorithm* A1: *Use the EV decomposition.* The Moore–Penrose inverse of the symmetric matrix $\mathbf{M}$ is computed using the real Schur (i.e., eigenvalue) decomposition

$$\mathbf{M}^\dagger = \mathbf{V}_M \begin{bmatrix} \mathbf{\Lambda}_M^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}_M^T = \mathbf{V}_{M_1}\mathbf{\Lambda}_M^{-1}\mathbf{V}_{M_1}^T$$

with $\mathbf{V}_{M_1} \in \mathcal{R}^{(n+m)\times p}$, $\mathbf{\Lambda}_M = \operatorname{diag}(\lambda_1, \ldots, \lambda_p) \in \mathcal{R}^{p\times p}$, $p = \operatorname{rank}(\mathbf{M}) = \operatorname{rank}(\mathbf{N}) = 2m + r$, $r = \operatorname{rank}(\mathbf{\Gamma})$. This approach corresponds to the singular value decomposition for the unsymmetric case [12, p. 243] and consists of two substeps.[1]

- Solving the symmetric eigenvalue problem for $\mathbf{M}$. Computing the matrices $\mathbf{V}_M$ and $\mathbf{\Lambda}_M$ requires $5(n+m)^3$ floating point operations (flops).
- Performing the symmetric matrix product $\mathbf{V}_{M_1}\mathbf{\Lambda}_M^{-1}\mathbf{V}_{M_1}^T$ requires $n^2p/2$ flops using the fact that we need only the upper left $n \times n$ part of $\mathbf{M}^\dagger$.

Therefore the total cost of computing $\mathbf{M}^\dagger$ by EV decomposition is $5(n+m)^3 + n^2p/2 = 5(n+m)^3 + n^2(2m+r)/2$ flops.

4. *Algorithm* A1: *Use the QR decomposition.*

- Apply the *complete* orthogonal decomposition (which is shown in Appendix 1 for $\mathbf{\Gamma}$) to the symmetric matrix $\mathbf{M}$:

$$\mathbf{M} = \mathbf{Q}_M \begin{bmatrix} \mathbf{L}_M^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P}_M^T = \mathbf{Y}_M[\mathbf{L}_M^T \quad \mathbf{0}]\mathbf{P}_M^T$$

[1] Here, as in the following discussions of numbers of floating point operations, only the cubic terms are counted. We use the older definition of *flop* for one floating point multiplication or division including an addition, as was used in the first edition of [12] and [11].

with nonsingular upper triangular $\mathbf{L}_M^T \in \mathcal{R}^{p\times p}$ and orthogonal $\mathbf{Q}_M = [\mathbf{Y}_M \quad \mathbf{Z}_M] \in \mathcal{R}^{(n+m)\times(n+m)}$ and $\mathbf{P}_M \in \mathcal{R}^{(n+m)\times(n+m)}$. This takes $2(n+m)^2p-(n+m)p^2-p^3/3 = 2(n+m)^2(2m+r)-(n+m)(2m+r)^2-(2m+r)^3/3$ flops.

- The Moore–Penrose inverse of $\mathbf{M}$,

$$\mathbf{M}^\dagger = \mathbf{P}_M \begin{bmatrix} (\mathbf{L}_M^T)^{-1} \\ \mathbf{0} \end{bmatrix} \mathbf{Y}_M^T,$$

is computed as shown in Appendix 1 for $\mathbf{\Gamma}^\dagger$. This takes $5(n+m)p^2/2 + 2(n+m)p(n+m-p)-p^3 = 5(n+m)(2m+r)^2/2+2(n+m)(2m+r)(n-m-r)-(2m+r)^3$ flops.

Therefore the total cost of computing $\mathbf{M}^\dagger$ by the complete orthogonal decomposition is $4(n+m)^2p-(n+m)p^2/2-4p^3/3 = 4(n+m)^2(2m+r)-(n+m)(2m+r)^2/2-4(2m+r)^3/3$ flops.

We now formulate a new algorithm for computing a **COV** matrix which uses the block structure of the Lagrangian matrix $\mathbf{M}$ to process smaller matrices. This assumes that the QR decomposition of a set of m linear independent linear(ized) constraints in $\mathcal{C}$ is given by

$$\mathbf{Q}^T\mathcal{C} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}, \quad \text{with} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Y}^T \\ \mathbf{Z}^T \end{bmatrix} = \mathbf{Q}^{-T},$$

where $\mathbf{Y} \in \mathcal{R}^{n\times m}$, $\mathbf{Z} \in \mathcal{R}^{n\times t}$, and $\mathbf{R} \in \mathcal{R}^{m\times m}$ is invertible upper triangular. We further assume that these matrices as well as the projected Hessian $\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{t\times t}$ are available at no cost at the end of the optimization process and are given in explicit form. This is a reasonable assumption since $\mathbf{Y}$ and $\mathbf{Z}$ are usually obtained by up- and downdating in an active set method [8] and the projected Hessian is needed for the second-order convergence test.

Algorithm A2: *The partitioned approach.*

1. At the start of this (or any other) algorithm one does not know whether $\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z}$ is singular or not. For the majority of applications one would want to use a fast and stable algorithm to determine the singularity of $\mathbf{\Gamma}$, if any, and to compute $\mathbf{\Gamma}^{-1}$ when $\det(\mathbf{\Gamma}) \neq 0$. We prefer to use the *diagonal pivoting factorization* method by Bunch and Kaufman [2] (see [5, Chap. 5, subroutines DSPFA and DSPSL]).

- First we factorize $\mathbf{\Gamma}$,

$$\mathbf{\Gamma} = \mathbf{U}\mathbf{D}\mathbf{U}^T,$$

where $\mathbf{U}$ is unit upper triangular and $\mathbf{D}$ is a diagonal block matrix with 1×1 and 2×2 blocks. This algorithm also provides the inertia (number of positive, negative, and zero eigenvalues) of the projected Hessian. In practice, we have already computed this factorization for the second-order convergence test. Using DSPFA for the factorization requires $t^3/6$ flops.

- If $\det(\mathbf{\Gamma}) \neq 0$ we can compute $\mathbf{\Gamma}^{-1}$ with a slightly modified form of DSPSL: First we solve the unit diagonal lower triangular system

$$\mathbf{W}^T = \mathbf{U}^{-T}\mathbf{Z}^T \quad \text{for} \quad \mathbf{W} \in \mathcal{R}^{n\times t}$$

and afterward we obtain

$$\mathbf{\Phi} = \mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T.$$

This step requires $nt^2/2$ flops for computing $\mathbf{W}$ and $n^2t/2$ flops for computing the symmetric matrix product.

Having found $\mathbf{\Gamma}^{-1}$ we have $\mathbf{\Phi}$ and stop the algorithm. If $\det(\mathbf{\Gamma}) = 0$ there are two alternatives.

- If $\mathcal{G}$ is positive semidefinite compute $\mathbf{\Phi} = \mathbf{Z}\mathbf{\Gamma}^{\dagger}\mathbf{Z}^T$ by using an orthogonal decomposition of $\mathbf{\Gamma}$ as in step 3 of this algorithm and by applying the Householder transformations of the outer orthogonal matrices of the decomposition directly on $\mathbf{Z}$ and $\mathbf{Z}^T$, thereby skipping most of the remaining steps of the algorithm.
- If $\mathcal{G}$ is indefinite perform all steps of this algorithm for evaluating (15).

2. Compute $\mathbf{A}$ and $\mathbf{E}$ by the following matrix products shown in Table 1.

TABLE 1

Matrix	Dim	Number of flops
$\mathbf{Y}^T\mathcal{G}$	$m \times n$	n^2m
$\mathbf{A} = (\mathbf{Y}^T\mathcal{G})\mathbf{Y}$	$m \times m$	$nm^2/2$
$\mathbf{E} = (\mathbf{Y}^T\mathcal{G})\mathbf{Z}$	$m \times t$	nmt

Computing $\mathbf{A}$ and $\mathbf{E}$ requires $2n^2m - nm^2/2$ flops in total. We do not need to compute $\mathbf{A}$ and $\mathbf{E}$ if $\mathcal{G}$ is positive semidefinite.

3. We apply the *complete* orthogonal decomposition (see Appendix 1) to the symmetric matrix $\mathbf{\Gamma} \in \mathcal{R}^{t\times t}$, $r = \operatorname{rank}(\mathbf{\Gamma})$. Here we use the same notation as in Algorithm A1 although the dimension of the matrices has changed radically, $\mathbf{M} \in \mathcal{R}^{(n+m)\times(n+m)}$ versus $\mathbf{\Gamma} \in \mathcal{R}^{t\times t}$:

$$\mathbf{\Gamma} = \mathbf{Q}_\Gamma \begin{bmatrix} \mathbf{L}_\Gamma^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P}_\Gamma^T = \mathbf{Y}_\Gamma [\mathbf{L}_\Gamma^T \quad \mathbf{0}] \mathbf{P}_\Gamma^T$$

with nonsingular upper triangular $\mathbf{L}_\Gamma^T \in \mathcal{R}^{r\times r}$ and orthogonal $\mathbf{Q}_\Gamma = [\mathbf{Y}_\Gamma \quad \mathbf{Z}_\Gamma] \in \mathcal{R}^{t\times t}$ and $\mathbf{P}_\Gamma \in \mathcal{R}^{t\times t}$. This takes $2t^2r - tr^2 - r^3/3$ flops.

After or while performing the first step of the complete orthogonal decomposition we apply the r Householder transformations in $\mathbf{Q}_\Gamma$ to the matrices $\mathbf{E}$ and $\mathbf{Z}$, forming

$$\mathbf{Q}_\Gamma^T\mathbf{E}^T = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_\Gamma^T\mathbf{Z}^T = \begin{bmatrix} \mathbf{M}_3 \\ \mathbf{M}_4 \end{bmatrix},$$

where $\mathbf{M}_1 \in \mathcal{R}^{r\times m}$, $\mathbf{M}_2 \in \mathcal{R}^{s\times m}$, $\mathbf{M}_3 \in \mathcal{R}^{r\times n}$, and $\mathbf{M}_4 \in \mathcal{R}^{s\times n}$. Applying $\mathbf{Q}_\Gamma$ to $\mathbf{E} \in \mathcal{R}^{m\times t}$ requires $(2t-r)rm$ flops and applying $\mathbf{Q}_\Gamma$ on $\mathbf{Z} \in \mathcal{R}^{n\times t}$ requires $(2t-r)rn$ flops [12, p. 197]. If $\mathcal{G}$ is positive semidefinite we only need to compute $\mathbf{M}_3$ (see the next step).

4. If $\mathcal{G}$ is positive semidefinite we perform the final matrix products for

$$\mathbf{\Phi} = \mathbf{Z}\mathbf{\Gamma}^{\dagger}\mathbf{Z}^T = \mathbf{Z}\mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{M}_3 \\ \mathbf{M}_4 \end{bmatrix} = \mathbf{Z}\mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1}\mathbf{M}_3 \\ \mathbf{0} \end{bmatrix}.$$

Computing $(\mathbf{L}_\Gamma^T)^{-1}\mathbf{M}_3 \in \mathcal{R}^{r\times n}$ via backward substitution takes $nr^2/2$ flops, applying the r Householder transformations in $\mathbf{P}_\Gamma$ takes $2nrs = 2n^2r - 2nmr - 2nr^2$ flops, and left multiplying with $\mathbf{Z}$ takes $n^2t/2$ flops exploiting symmetry of $\mathbf{\Phi}$.

5. In this step we want to estimate the matrix $\mathbf{W}_1 = \mathbf{X}_1^{\dagger}\mathbf{Z}^T$ by solving a linear least-squares problem for $\mathbf{W}_1 \in \mathcal{R}^{m\times n}$. We showed at the end of §2 that

$$\mathbf{X} = \begin{bmatrix} (\mathbf{I} - \mathbf{\Gamma}\mathbf{\Gamma}^{\dagger})\mathbf{E}^T \\ \mathbf{R}^T \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T \\ \mathbf{R}^T \end{bmatrix} \in \mathcal{R}^{n\times m},$$

where

$$\mathbf{F}^T = (\mathbf{I} - \mathbf{\Gamma}\mathbf{\Gamma}^{\dagger})\mathbf{E}^T = \mathbf{Q}_\Gamma \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_s \end{bmatrix} \mathbf{Q}_\Gamma^T\mathbf{E}^T = \mathbf{Q}_\Gamma\mathbf{J}^T$$

with

$$\mathbf{J}^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_s \end{bmatrix} \mathbf{Q}_\Gamma^T \mathbf{E}^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_s \end{bmatrix} \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_2 \end{bmatrix}.$$

Therefore we obtain

$$\mathbf{F} = \mathbf{J}\mathbf{Q}_\Gamma^T = [\mathbf{0} \quad \mathbf{M}_2^T]\mathbf{Q}_\Gamma^T.$$

Now we compute $\mathbf{W}_1 = \mathbf{X}_1^\dagger \mathbf{Z}^T$ as the solution of the least-squares problem

$$\begin{bmatrix} \mathbf{F}^T \\ \mathbf{R}^T \end{bmatrix} \mathbf{W}_1 = \begin{bmatrix} \mathbf{Z}^T \\ \mathbf{0} \end{bmatrix}.$$

This is equivalent to solving the least-squares problem

$$\begin{bmatrix} \mathbf{J}^T \\ \mathbf{R}^T \end{bmatrix} \mathbf{W}_1 = \begin{bmatrix} \mathbf{Q}_\Gamma^T \mathbf{Z}^T \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_3 \\ \mathbf{M}_4 \\ \mathbf{0} \end{bmatrix},$$

which amounts to solving the linear system

$$(\mathbf{R}\mathbf{R}^T + \mathbf{J}\mathbf{J}^T)\mathbf{W}_1 = \mathbf{J}\mathbf{Q}_\Gamma^T \mathbf{Z}^T.$$

Now because of $\mathbf{J} = [\mathbf{0} \quad \mathbf{M}_2^T]$ the latter reduces to

$$(\mathbf{R}\mathbf{R}^T + \mathbf{M}_2^T \mathbf{M}_2)\mathbf{W}_1 = \mathbf{M}_2^T \mathbf{M}_4, \quad \mathbf{M}_2 \in \mathcal{R}^{s \times m}, \ \mathbf{M}_4 \in \mathcal{R}^{s \times n}.$$

We exploit the triangular shape of $\mathbf{R} \in \mathcal{R}^{m \times m}$ and use the orthogonal anti-Cholesky ($\mathbf{U}\mathbf{U}^T$) factor update algorithm (rank s update) of Appendix 1 to compute the anti-Cholesky factor of the $m \times m$ coefficient matrix and solve the linear system. Computing the anti-Cholesky factor of $(\mathbf{R}\mathbf{R}^T + \mathbf{M}_2^T \mathbf{M}_2)$ takes $m^2 s = m^2(n - m - r) = m^2(t - r)$ flops, multiplying $\mathbf{M}_2^T$ with $\mathbf{M}_4$ takes $nms = nm(t - r)$ flops, and solving the $m \times m$ linear system with n right-hand sides $\mathbf{M}_2^T \mathbf{M}_4$ by forward and backward substitution takes nm^2 flops. Therefore this step requires $n^2 m + nm^2 - m^3 - (n + m)mr$ flops in total.

6. Finally, compute the three (symmetric) matrices $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{S}_3$ necessary to obtain $\mathbf{\Phi} = \mathbf{S}_1 + \mathbf{S}_2 + \mathbf{S}_3$ as in (15):

$$\begin{aligned} \mathbf{S}_1 &= \mathbf{Z}(\mathbf{X}_1^\dagger)^T \mathbf{Y}^T + \mathbf{Y}\mathbf{X}_1^\dagger \mathbf{Z}^T = \mathbf{W}_1^T \mathbf{Y}^T + \mathbf{Y}\mathbf{W}_1, \\ \mathbf{S}_2 &= -\mathbf{Z}(\mathbf{X}_1^\dagger)^T \mathbf{A}\mathbf{X}_1^\dagger \mathbf{Z}^T = -\mathbf{W}_1^T \mathbf{A}\mathbf{W}_1, \\ \mathbf{S}_3 &= \mathbf{Z}(\mathbf{I} - (\mathbf{X}_1^\dagger)^T \mathbf{E})\mathbf{\Gamma}^\dagger (\mathbf{I} - \mathbf{E}^T \mathbf{X}_1^\dagger)\mathbf{Z}^T \\ &= (\mathbf{Z} - \mathbf{W}_1^T \mathbf{E})\mathbf{\Gamma}^\dagger (\mathbf{Z}^T - \mathbf{E}^T \mathbf{W}_1). \end{aligned}$$

The matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ are obtained simply by matrix multiplication. Computing $\mathbf{S}_1$ takes $n^2 m$ and computing the quadratic form for $\mathbf{S}_2$ takes $n^2 m/2 + nm^2$ flops. For matrix $\mathbf{S}_3$ we use the orthogonal decomposition of $\mathbf{\Gamma}$ and solve the following least-squares problem for the $t \times n$ matrix $\mathbf{T}$:

$$\begin{aligned} \mathbf{T} &= \mathbf{\Gamma}^\dagger (\mathbf{Z}^T - \mathbf{E}^T \mathbf{W}_1) \\ &= \mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{Q}_\Gamma^T (\mathbf{Z}^T - \mathbf{E}^T \mathbf{W}_1) \\ &= \mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \left(\begin{bmatrix} \mathbf{M}_3 \\ \mathbf{M}_4 \end{bmatrix} - \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix} \mathbf{W}_1 \right) \\ &= \mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1} (\mathbf{M}_3 - \mathbf{M}_1 \mathbf{W}_1) \\ \mathbf{0} \end{bmatrix}. \end{aligned}$$

Multiplying $\mathbf{M}_1$ with $\mathbf{W}_1$ takes nmr flops. Computing the $r \times n$ matrix $(\mathbf{L}_\Gamma^T)^{-1}(\mathbf{M}_3 - \mathbf{M}_1\mathbf{W}_1)$ requires $nr^2/2$ flops. Applying the r Householder transformations stored in $\mathbf{P}_\Gamma$ to this matrix takes $2nrs$ flops. Now we obtain $\mathbf{S}_3 = (\mathbf{Z} - \mathbf{W}_1^T\mathbf{E})\mathbf{T}$ by matrix multiplication using $nmt + n^2t/2$ flops exploiting symmetry of $\mathbf{S}_3$. Therefore, this step makes use of $n^3/2 + 2n^2m + 2n^2r - nmr - 3nr^2/2$ flops in total.

The cubic terms in the flop counts made in steps 2, 3, 5, and 6 of Algorithm A2 amount to $n^3/2 + 5n^2m + nm^2/2 - m^3 + (6n^2 - m^2 - 6nm)r - 7nr^2/2 - r^3/3$. If $\mathcal{G}$ is positive semidefinite we only need to perform part of step 3 and step 4, which takes $n^2(n-m)/2 + 6n(n-m)r - 2m(n-m)r + 2m^2r - 7nr^2/2 + mr^2 - r^3/3$ flops.

4. Algorithm A3: An alternative approach to Algorithm A2. For certain values of m, n, and r, it may be advantageous to use an even finer partitioning of the blocks.

1. As in (12) we apply the QR decomposition to $\mathbf{M}$, obtaining $\mathbf{N}$:

$$\mathbf{N} = \begin{bmatrix} \mathbf{Y}^T & \mathbf{0} \\ \mathbf{Z}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathcal{G} & \mathcal{C} \\ \mathcal{C}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Y} & \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_m \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{E} & \mathbf{R} \\ \mathbf{E}^T & \Gamma & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

2. Diagonalize the symmetric projected Hessian Γ:

$$\mathbf{P}^T\Gamma\mathbf{P} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$ with $r = \operatorname{rank}(\Gamma) \leq n - m$ and Λ contains the nonzero eigenvalues λ_i of Γ. We apply the following orthogonal transformation to $\mathbf{N}$:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{E} & \mathbf{R} \\ \mathbf{E}^T & \Gamma & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{E}_1 & \mathbf{E}_2 & \mathbf{R} \\ \mathbf{E}_1^T & \Lambda & \mathbf{0} & \mathbf{0} \\ \mathbf{E}_2^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{EP} = [\mathbf{E}_1 \quad \mathbf{E}_2] \in \mathcal{R}^{m \times t}$. The columns of matrix $\mathbf{E}_1 \in \mathcal{R}^{m \times r}$ correspond to the nonzero eigenvalues of Γ, and the columns of the (small) matrix $\mathbf{E}_2 \in \mathcal{R}^{m \times s}$ correspond to the zero eigenvalues of Γ. Solving the symmetric eigenvalue problem requires $5t^3$ flops. Computing $\mathbf{A} = \mathbf{Y}^T\mathcal{G}\mathbf{Y} \in \mathcal{R}^{m \times m}$, $\mathbf{E} = \mathbf{Y}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{m \times t}$, and $\mathbf{EP} = [\mathbf{E}_1 \quad \mathbf{E}_2] \in \mathcal{R}^{m \times t}$ takes $2n^2m - nm^2/2 + mt^2$ flops.

3. Temporarily permute the third and fourth block row and block column to give

$$\hat{\mathbf{N}} = \left[\begin{array}{ccc|c} \mathbf{A} & \mathbf{E}_1 & \mathbf{R} & \mathbf{E}_2 \\ \mathbf{E}_1^T & \Lambda & \mathbf{0} & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{E}_2^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] = \begin{bmatrix} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hat{\mathbf{B}}^T & \mathbf{0} \end{bmatrix}, \quad \text{with} \quad \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{E}_1 & \mathbf{R} \\ \mathbf{E}_1^T & \Lambda & \mathbf{0} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

The matrix $\hat{\mathbf{A}} \in \mathcal{R}^{(2m+r)\times(2m+r)}$ has full rank, $\operatorname{rank}(\hat{\mathbf{A}}) = \operatorname{rank}(\hat{\mathbf{N}}) = 2m + r$, and is easily inverted to the form

$$\hat{\mathbf{A}}^{-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{R}^{-T} \\ \mathbf{0} & \times & \times \\ \mathbf{R}^{-1} & \times & \times \end{bmatrix}.$$

4. Now apply Proposition 2 of [16]: If

$$\hat{\mathbf{N}} = \begin{bmatrix} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hat{\mathbf{B}}^T & \mathbf{0} \end{bmatrix}$$

and $\operatorname{rank}(\hat{\mathbf{N}}) = \operatorname{rank}(\hat{\mathbf{A}})$ then

$$\hat{\mathbf{N}}^\dagger = [\mathbf{I} \quad \hat{\mathbf{A}}^\dagger\hat{\mathbf{B}}]^\dagger \hat{\mathbf{A}}^\dagger \begin{bmatrix} \mathbf{I} \\ \hat{\mathbf{B}^T}\hat{\mathbf{A}}^\dagger \end{bmatrix}^\dagger.$$

Since $\hat{\mathbf{A}}$ is nonsingular it follows that

$$\mathbf{K} = \hat{\mathbf{A}}^\dagger\hat{\mathbf{B}} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{B}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \times \\ \mathbf{0} & \times & \times \\ \mathbf{R}^{-1} & \times & \times \end{bmatrix} \begin{bmatrix} \mathbf{E}_2 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{R}^{-1}\mathbf{E}_2 \end{bmatrix}.$$

Hence

$$[\mathbf{I} \quad \hat{\mathbf{A}}^\dagger\hat{\mathbf{B}}] = \left[\begin{array}{ccc|c} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{R}^{-1}\mathbf{E}_2 \end{array}\right].$$

Also, since

$$[\mathbf{I} \quad \mathbf{K}]^\dagger = \begin{bmatrix} \mathbf{I} \\ \mathbf{K}^T \end{bmatrix} (\mathbf{I} + \mathbf{K}\mathbf{K}^T)^{-1},$$

we have

$$\mathbf{I} + \mathbf{K}\mathbf{K}^T = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} + (\mathbf{R}^{-1}\mathbf{E}_2)(\mathbf{R}^{-1}\mathbf{E}_2)^T \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}} \end{bmatrix}$$

with

$$\hat{\mathbf{H}} = \mathbf{I} + \mathbf{R}^{-1}\mathbf{E}_2\mathbf{E}_2^T\mathbf{R}^{-T} = \mathbf{R}^{-1}(\mathbf{R}\mathbf{R}^T + \mathbf{E}_2\mathbf{E}_2^T)\mathbf{R}^{-T} = \mathbf{R}^{-1}\mathbf{H}\mathbf{R}^{-T}$$

and

$$\mathbf{H} := \mathbf{R}\mathbf{R}^T + \mathbf{E}_2\mathbf{E}_2^T \in \mathcal{R}^{m\times m},$$

which is positive definite. Therefore

$$[\mathbf{I} \quad \mathbf{K}]^\dagger = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & (\mathbf{R}^{-1}\mathbf{E}_2)^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}}^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}}^{-1} \\ \mathbf{0} & \mathbf{0} & (\mathbf{R}^{-1}\mathbf{E}_2)^T\hat{\mathbf{H}}^{-1} \end{bmatrix}.$$

Because

$$\mathbf{R}\hat{\mathbf{H}} = \mathbf{H}\mathbf{R}^{-T} \quad \text{and} \quad \hat{\mathbf{H}}^{-1}\mathbf{R}^{-1} = \mathbf{R}^T\mathbf{H}^{-1},$$

we now have

$$(\mathbf{R}^{-1}\mathbf{E}_2)^T\hat{\mathbf{H}}^{-1} = (\mathbf{R}^{-1}\mathbf{E}_2)^T(\mathbf{I} + (\mathbf{R}^{-1}\mathbf{E}_2)(\mathbf{R}^{-1}\mathbf{E}_2)^T)^{-1} = \mathbf{E}_2^T\mathbf{H}^{-1}\mathbf{R}.$$

5. Substituting into Proposition 2 of [16] we arrive at

$$\begin{aligned}\hat{\mathbf{N}}^{\dagger} &= [\mathbf{I} \quad \hat{\mathbf{A}}^{-1}\hat{\mathbf{B}}]^{\dagger}\hat{\mathbf{A}}^{-1}\begin{bmatrix}\mathbf{I}\\ \hat{\mathbf{B}}^T\hat{\mathbf{A}}^{-1}\end{bmatrix}^{\dagger}\\ &= \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}}^{-1}\\ \mathbf{0} & \mathbf{0} & \mathbf{K}^T\mathbf{R}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{R}^{-T}\\ \mathbf{0} & \mathbf{\Lambda}^{-1} & -\mathbf{\Lambda}^{-1}\mathbf{E}_1^T\mathbf{R}^{-T}\\ \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{E}_1\mathbf{\Lambda}^{-1} & \mathbf{R}^{-1}\mathbf{\Delta}\mathbf{R}^{-T}\end{bmatrix}\cdots\\ &= \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{W} & \mathbf{K}\\ \mathbf{0} & \mathbf{\Lambda}^{-1} & -\mathbf{\Lambda}^{-1}\mathbf{E}_1^T\mathbf{W} & -\mathbf{\Lambda}^{-1}\mathbf{E}_1^T\mathbf{K}\\ \mathbf{W}^T & -\mathbf{W}^T\mathbf{E}_1\mathbf{\Lambda}^{-1} & \mathbf{W}^T\mathbf{\Delta}\mathbf{W} & \mathbf{W}^T\mathbf{\Delta}\mathbf{K}\\ \mathbf{K}^T & -\mathbf{K}^T\mathbf{E}_1\mathbf{\Lambda}^{-1} & \mathbf{K}^T\mathbf{\Delta}\mathbf{W} & \mathbf{K}^T\mathbf{\Delta}\mathbf{K}\end{bmatrix},\end{aligned}$$

where

$$\mathbf{\Delta} := \mathbf{E}_1\mathbf{\Lambda}^{-1}\mathbf{E}_1^T - \mathbf{A} \in \mathcal{R}^{m\times m},$$

$$\mathbf{W} := \mathbf{H}^{-1}\mathbf{R} \in \mathcal{R}^{m\times m}, \quad \mathbf{K} := \mathbf{H}^{-1}\mathbf{E}_2 \in \mathcal{R}^{m\times s},$$

$$\mathbf{H} = \mathbf{R}\mathbf{R}^T + \mathbf{E}_2\mathbf{E}_2^T = \hat{\mathbf{R}}\hat{\mathbf{R}}^T \in \mathcal{R}^{m\times m}.$$

This, in essence, is a finer version of (14). Therefore, $\mathbf{W}$ and $\mathbf{K}$ are computed by the anti-Cholesky factor update algorithm (see Appendix 1). The orthogonal rank s update of the anti-Cholesky factor $\mathbf{R}$ takes m^2s flops. Solving $\mathbf{W}$ and $\mathbf{K}$ with forward and backward substitution requires together $m^2(m+s)$ flops; together these are $m^3 + 2m^2s$ flops. Computing the symmetric matrix $\mathbf{\Delta} := \mathbf{E}_1\mathbf{\Lambda}^{-1}\mathbf{E}_1^T - \mathbf{A} \in \mathcal{R}^{m\times m}$ takes an additional $m^2r/2$ flops. Therefore, performing this step requires $2nm^2 - m^3 - 3m^2r/2$ flops.

6. To obtain the Moore–Penrose inverse of $\mathbf{M}$ we must repermute the third and fourth row and column blocks in $\hat{\mathbf{N}}^{\dagger}$ and reapply the orthogonal transformations of the second and first steps of this method:

$$\begin{aligned}\mathbf{\Phi} &= \mathbf{Q}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{P}\end{bmatrix}\hat{\mathbf{N}}^{\dagger}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{P}^T\end{bmatrix}\mathbf{Q}^T\\ &= [\mathbf{Y} \quad \mathbf{Z}]\begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{P}_1 & \mathbf{P}_2\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{K}\\ \mathbf{0} & \mathbf{\Lambda}^{-1} & -\mathbf{\Lambda}^{-1}\mathbf{E}_1^T\mathbf{K}\\ \mathbf{K}^T & -\mathbf{K}^T\mathbf{E}_1\mathbf{\Lambda}^{-1} & \mathbf{K}^T\mathbf{\Delta}\mathbf{K}\end{bmatrix}\cdots\\ &= [\mathbf{Y} \quad \mathbf{Z}\mathbf{P}_1 \quad \mathbf{Z}\mathbf{P}_2]\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{K}\\ \mathbf{0} & \mathbf{\Lambda}^{-1} & -\mathbf{\Lambda}^{-1}\mathbf{E}_1^T\mathbf{K}\\ \mathbf{K}^T & -\mathbf{K}^T\mathbf{E}_1\mathbf{\Lambda}^{-1} & \mathbf{K}^T\mathbf{\Delta}\mathbf{K}\end{bmatrix}\begin{bmatrix}\mathbf{Y}^T\\ \mathbf{P}_1^T\mathbf{Z}^T\\ \mathbf{P}_2^T\mathbf{Z}^T\end{bmatrix}\\ &= \mathbf{Y}\mathbf{K}\mathbf{P}_2^T\mathbf{Z}^T + \mathbf{Z}\mathbf{P}_2\mathbf{K}^T\mathbf{Y}^T + \mathbf{Z}(\mathbf{P}_1 - \mathbf{P}_2\mathbf{K}^T\mathbf{E}_1)\mathbf{\Lambda}^{-1}\mathbf{P}_1^T\mathbf{Z}^T\\ &\quad + \mathbf{Z}(\mathbf{P}_2\mathbf{K}^T\mathbf{\Delta} - \mathbf{P}_1\mathbf{\Lambda}^{-1}\mathbf{E}_1^T)\mathbf{K}\mathbf{P}_2^T\mathbf{Z}^T\\ &= (\hat{\mathbf{P}}_2\mathbf{Y}^T + \mathbf{Y}\hat{\mathbf{P}}_2^T) - (\hat{\mathbf{P}}_2\mathbf{E}_1\mathbf{\Lambda}^{-1}\hat{\mathbf{P}}_1^T + \hat{\mathbf{P}}_1\mathbf{\Lambda}^{-1}\mathbf{E}_1\hat{\mathbf{P}}_2^T)\\ &\quad + \hat{\mathbf{P}}_1\mathbf{\Lambda}^{-1}\hat{\mathbf{P}}_1^T + \hat{\mathbf{P}}_2\mathbf{\Delta}\hat{\mathbf{P}}_2^T\\ &= (\hat{\mathbf{Y}} + \hat{\mathbf{Y}}^T) - (\mathbf{E}_{21} + \mathbf{E}_{21}^T) + \mathbf{P}_{11} + \mathbf{P}_{22}.\end{aligned}$$

The resubstitution can be done by computing the matrix products as shown in Table 2.

TABLE 2

Matrix	Dim	Number of flops
$\hat{\mathbf{P}}_1 = \mathbf{Z}\mathbf{P}_1$	$n \times r$	nrt
$\hat{\mathbf{P}}_2 = \mathbf{Z}\mathbf{P}_2\mathbf{K}^T$	$n \times m$	$n^2 s$
$\hat{\mathbf{Y}} = \hat{\mathbf{P}}_2\mathbf{Y}^T$	$n \times n$	$n^2 m$
$\mathbf{E}_{21} = \hat{\mathbf{P}}_2\mathbf{E}_1\mathbf{\Lambda}^{-1}\hat{\mathbf{P}}_1^T$	$n \times r$	$(nm + n^2)r$
$\mathbf{P}_{11} = \hat{\mathbf{P}}_1\mathbf{\Lambda}^{-1}\hat{\mathbf{P}}_1^T$	$n \times n$	$n^2 r/2$
$\mathbf{P}_{22} = \hat{\mathbf{P}}_2\mathbf{\Delta}\hat{\mathbf{P}}_2^T$	$n \times n$	$nm^2 + n^2 m/2$

Therefore, performing this step requires $n^3 + n^2 m/2 + nm^2 + 3n^2 r/2$ flops.

5. Comparison of computer time. Since Algorithms A2 and A3 process the projected Hessian $\mathbf{\Gamma} \in \mathcal{R}^{t\times t}$ rather than the large Lagrangian matrix $\mathbf{M} \in \mathcal{R}^{(n+m)\times(n+m)}$, they can be more efficient in those cases where m is larger than a threshold, depending on the size of n and the number r of nonzero eigenvalues of $\mathbf{\Gamma}$. Table 3 contains the number of flops needed to compute the matrix $\mathbf{\Phi}$ using Algorithms A1, A2, and A3. We also report the flop count of Algorithm AA, which uses complete orthogonal decomposition to evaluate $\mathbf{\Phi} = \mathbf{Z}\mathbf{\Gamma}^{\dagger}\mathbf{Z}^T$ and which can be used only when $\mathbf{\Gamma}$ is nonsingular or $\mathcal{G}$ is positive semidefinite (satisfying Theorem 6.1 in Appendix 2).

TABLE 3
Number of flops needed to compute $\mathbf{\Phi}$.

Alg	Number of flops
A1_EV	$5(n+m)^3 + n^2(2m+r)/2$
A1_QR	$4(n+m)^2(2m+r) - (n+m)(2m+r)^2/2 - 4(2m+r)^3/3$
A2_QR	$n^3/2 + 5n^2 m + nm^2/2 - m^3 + (6n^2 - m^2 - 6nm)r - 7nr^2/2 - r^3/3$
A3_EV	$n^3 + 7n^2 m/2 + nm^2/2 + 3(n^2 - m^2)r/2$
AA_QR	$n^2(n-m)/2 + 6n(n-m)r - 2m(n-m)r + 2m^2 r - 7nr^2/2 + mr^2 - r^3/3$

In Figures 1 and 2 we show the curves CA1_EV(n, m, r), CA1_QR(n, m, r), CA2_QR(n, m, r), and CA3_EV(n, m, r) for a constant $n = 100$, $m = 1, \ldots, n-2$ and the two choices $r = n - m - 1$ and $r = (n-m)/2$. Of course, the savings made in Algorithms A2 and A3 are bigger for larger m. The number of flops used in Algorithms A1_EV and A1_QR grows as m^3, unlike Algorithms A2 and A3. The curves for Algorithms A2 and A3 are almost identical, indicating the fact that the current implementations of the two algorithms perform almost equally. We think, however, that we will be able to improve the performance of a modified version of Algorithm A3 in the future.

6. Implementation and example. An SAS IML (interactive matrix language) macro program was written to implement Algorithms A1, A2, and A3. Its listing is available upon request from the first author. The macro uses the output data set of the SAS OR procedure NLP (**N**on**L**inear **P**rogramming; see [15]) containing the Hessian matrix $\mathcal{G}$ and the set of linear independent active constraints $\mathcal{C}$ evaluated at the solution x^* of a (constrained) nonlinear optimization problem obtained by PROC NLP. After reading $\mathcal{G}$ and $\mathcal{C}$, the macro computes the covariance matrix and approximate standard errors using the four Algorithms A1 (QR and EV), A2 (QR), and A3 (EV). The difficulty in constructing a test example lies in the fact that simple and more efficient algorithms can be used for the two most common situations ($\mathbf{\Gamma}$ is nonsingular or $\mathcal{G}$ is positive semidefinite).

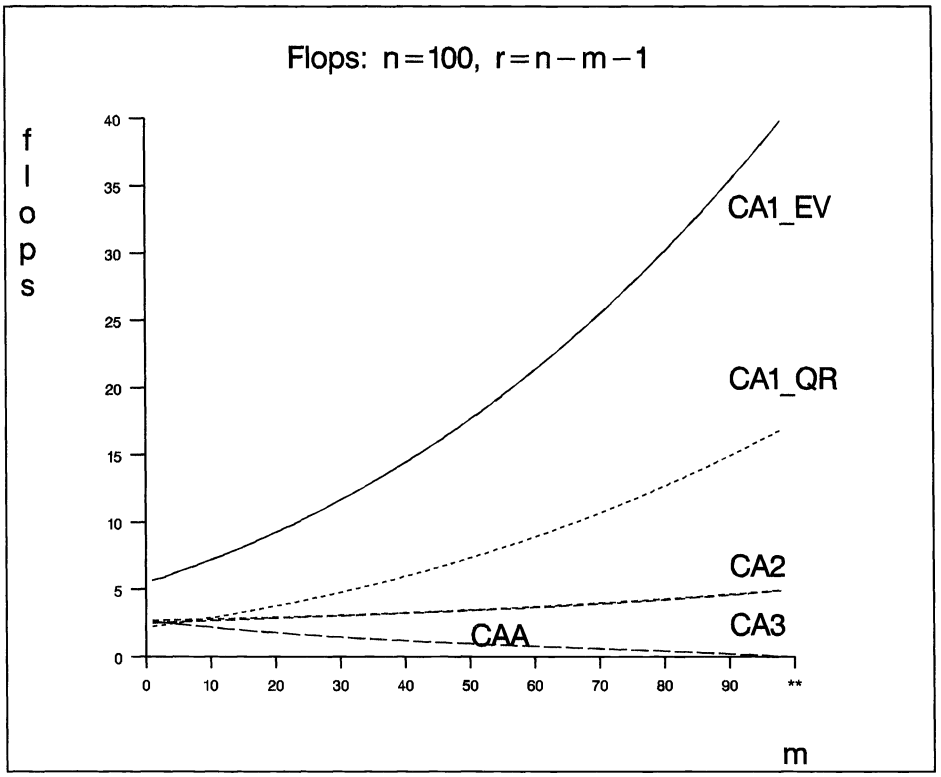


FIG. 1. *Algorithms* A1, A2, A3, AA: *Number of flops for* $n = 100$ *and* $r = n - m - 1$.

In Appendix 3 of this paper we develop a method to construct example matrices $\mathcal{G} \in \mathcal{R}^{n\times n}$ and $\mathbf{Z} \in \mathcal{R}^{n\times t}$ satisfying the following conditions.

- Matrix $\mathcal{G}$ is indefinite and singular. We chose $\mathcal{G}$ to be diagonal with the same number of positive, negative, and zero diagonal elements, i.e., $n = 3s$, $s = \operatorname{rank}(\mathcal{G})$.
- Matrix $\mathbf{Z}$ is constructed so that $\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{t\times t}$ is positive semidefinite with $r = \operatorname{rank}(\mathbf{\Gamma}) = \operatorname{rank}(\mathcal{G}\mathbf{Z}) - 1 = s - 1$.

Using the matrices $\mathcal{G}$ and $\mathbf{Z}$ we construct a quadratic minimization problem,

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathcal{G}\mathbf{x} + \mathbf{g}^T\mathbf{x} \longrightarrow \min_x,$$

subject to $m > r$ linear constraints

$$c_i(\mathbf{x}) = \sum_{j=1}^{n} c_{ij}x_j \geq 0, \quad i = 1, \ldots, m,$$

as follows.

- Using an algorithm described in [4] (IML module ORTVEC), the m columns of the $n \times m$ matrix $\mathbf{Y}$ are generated orthogonal to the t columns of $\mathbf{Z}$, satisfying $\mathbf{Y}^T\mathbf{Z} = \mathbf{0}$ and $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}_m$. Setting the coefficients c_{ij} of the m linear constraints $c_i(\mathbf{x}) \geq 0$ to the columns of $\mathbf{Y}$, $\mathcal{C} = \mathbf{Y}$, ensures that the point $\mathbf{x} = 0$ satisfies all m linear constraints.

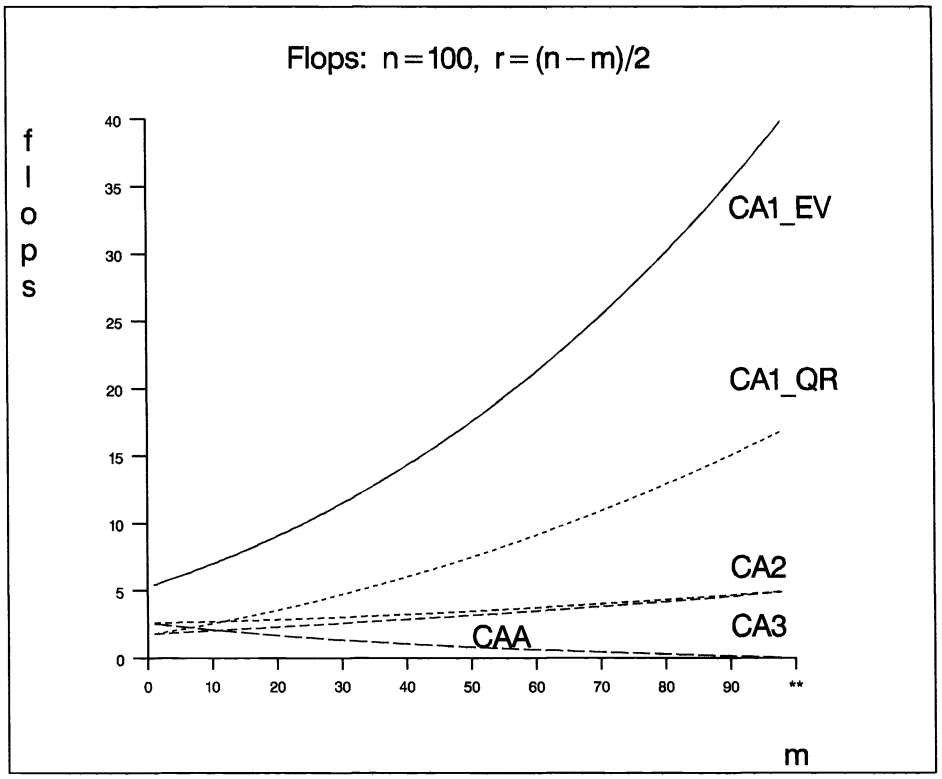


FIG. 2. *Algorithms* A1, A2, A3, AA: *Number of flops for* $n = 100$ *and* $r = (n - m)/2$.

- We select the linear term $\mathbf{g}$ of the quadratic problem as the sum of the m columns $\mathbf{y}_i$ of $\mathbf{Y}$, $\mathbf{g} = \sum_{i=1}^{m} \mathbf{y}_i$, ensuring that the point $\mathbf{x}^* = 0$ satisfies the following first- and second-order optimality conditions:
  - the projected gradient is zero, $\mathbf{Z}^T(\mathcal{G}\mathbf{x}^* + \mathbf{g}) = \mathbf{Z}^T\mathbf{g} = \mathbf{0}$,
  - the Lagrange multipliers λ_i of the m active constraints are all equal to one, $\lambda^* = \mathbf{Y}^T(\mathcal{G}\mathbf{x}^* + \mathbf{g}) = \mathbf{Y}^T\mathbf{g} = \mathbf{1}$,
  - the projected Hessian $\mathbf{\Gamma}$ is positive semidefinite by construction.

We constructed the four examples E1=($n = 30$, $m = 15$, $r = 9$), E2=($n = 90$, $m = 40$, $r = 29$), E3=($n = 150$, $m = 80$, $r = 49$), and E4=($n = 300$, $m = 150$, $r = 99$) and computed the covariance matrices using our IML implementation of the four algorithms. We counted the number of flops necessary to compute the covariance matrices with the four algorithms using the formulas developed above (see Tables 4 and 5) and we measured the real time necessary to compute the covariance matrices on our HP720 work station (see Table 6).

TABLE 4
Estimates of flop counts.

(n,m,r)	CA1_EV	CA1_QR	CA2	CA3	CAA
(30,15,9)	473175	202586	94527	86738	23517
(90,40,29)	11426450	4869430	2456055	2217750	722895
(150,80,49)	63186250	27028646	12129159	11338350	2845239
(300,150,99)	473580000	202674843	94888017	87648750	24335667

TABLE 5
Ratios of flop counts.

(n,m,r)	A1E/A1Q	A1E/A2	A1E/A3	A1Q/A2	A1Q/A3	A2/A3
(30,15,9)	2.33	5.01	5.46	2.14	2.34	1.09
(90,40,29)	2.35	4.65	5.15	1.98	2.20	1.11
(150,80,49)	2.34	5.21	5.57	2.23	2.38	1.07
(300,150,99)	2.34	4.99	5.40	2.14	2.31	1.08

TABLE 6
Real time in seconds.

(n,m,r)	TA1_EV	TA1_QR	TA2	TA3
(30,15,9)	0.20	0.16	0.12	0.11
(90,40,29)	3.60	3.20	1.66	1.52
(150,80,49)	20.8	27.11	7.10	7.09
(300,150,99)	235.	349.	75.6	56.3

We encountered some important differences.

- IML is an interactive matrix language that compiles and performs its code statement by statement. For each new defined matrix, memory is allocated at the time the statement is performed. Since the processing of the large augmented matrix requires only two statements, but our partitioned algorithms require a series of statements, IML penalizes the latter.
- Our test matrices are constructed with *diagonal* Hessian matrices. Although neither of the subroutines used in the computation takes advantage of the off-diagonal zeros directly, the diagonalization seems to exploit this situation much better than the (pivoted) triangularization. The RISC processor of the HP 720 performs floating point additions slower (three cycles) than multiplications (two cycles), and floating point divisions are about five times slower than multiplications. The performance of the HP 720 is also very sensitive with respect to the use of cache memory for large applications. Our QR decomposition with column pivoting is implemented as shown in the LINPACK manual; i.e., the columns are really swapped and the column norms are updated. This comes at a considerable additional running time expense which is not counted in our definition of *flops* and which seems to be significant on a RISC processor.

Since $\mathcal{G}$ is not positive semidefinite and $\mathbf{\Gamma}$ is singular we cannot apply Algorithm AA to these examples.

Appendix 1: The complete orthogonal decomposition of Γ. Our algorithms use the *complete* orthogonal decomposition of the singular matrix $\mathbf{\Gamma}$ to express the Moore–Penrose inverse $\mathbf{\Gamma}^{\dagger}$, $r = \operatorname{rank}(\mathbf{\Gamma}) < n - m$, $s = n - m - r > 0$. The same algorithm can be used to compute the complete orthogonal decomposition of the big matrix $\mathbf{M}$. The subscript Γ is used to prevent confusion of the two equivalent tasks.

1. *First step of complete orthogonal decomposition.* We use the QR decomposition of $\mathbf{\Gamma}$ with column pivoting,

$$\mathbf{\Gamma} = \mathbf{Q}_{\Gamma} \begin{bmatrix} \mathbf{R}_{\Gamma} \\ \mathbf{0} \end{bmatrix} \mathbf{\Pi}^T = [\mathbf{Y}_{\Gamma} \quad \mathbf{Z}_{\Gamma}] \begin{bmatrix} \mathbf{R}_{\Gamma 1} & \mathbf{R}_{\Gamma 2} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{\Pi}^T,$$

where $\mathbf{R}_{\Gamma} = [\mathbf{R}_{\Gamma 1} \quad \mathbf{R}_{\Gamma 2}] \in \mathcal{R}^{r \times t}$ is upper trapezoidal, $\mathbf{R}_{\Gamma 1} \in \mathcal{R}^{r \times r}$ is upper triangular and invertible, $\mathbf{R}_{\Gamma 2} \in \mathcal{R}^{r \times s}$, $\mathbf{Q}_{\Gamma} = [\mathbf{Y}_{\Gamma} \quad \mathbf{Z}_{\Gamma}]$ is orthogonal, $\mathbf{Y}_{\Gamma} \in \mathcal{R}^{t \times r}$, $\mathbf{Z}_{\Gamma} \in \mathcal{R}^{t \times s}$,

and $\mathbf{\Pi}$ permutes the columns of $\mathbf{\Gamma}$. Computing $\mathbf{R}_\Gamma$ and $\mathbf{Q}_\Gamma$ with r Householder transformations takes $2t^2r - 2tr^2 + 2r^3/3$ flops.

2. *Second step of complete orthogonal decomposition.* This is shown in [12, p. 236], and more efficiently in problem P5.2.5 on p. 220. Here we use the transpose $\mathbf{L}_\Gamma$ of the upper trapezoidal matrix $\mathbf{R}_\Gamma = [\mathbf{R}_{\Gamma 1} \quad \mathbf{R}_{\Gamma 2}]$,

$$\mathbf{L}_\Gamma = \begin{bmatrix} \mathbf{L}_{\Gamma 1} \\ \mathbf{L}_{\Gamma 2} \end{bmatrix} = \mathbf{R}_\Gamma^T \in \mathcal{R}^{t\times r}, \quad \text{with} \quad \operatorname{rank}(\mathbf{L}_\Gamma) = \operatorname{rank}(\mathbf{L}_{\Gamma 1}) = r,$$

$\mathbf{L}_{\Gamma 1} \in \mathcal{R}^{r\times r}$ lower triangular, $\mathbf{L}_{\Gamma 2} \in \mathcal{R}^{s\times r}$. The lower trapezoidal matrix $\mathbf{L}_\Gamma \in \mathcal{R}^{t\times r}$ is premultiplied with r Householder transformations $\mathbf{P}_{\Gamma 1}, \ldots, \mathbf{P}_{\Gamma r}$,

$$\mathbf{P}_{\Gamma r} \cdots \mathbf{P}_{\Gamma 1} \begin{bmatrix} \mathbf{L}_{\Gamma 1} \\ \mathbf{L}_{\Gamma 2} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_\Gamma \\ \mathbf{0} \end{bmatrix},$$

each zeroing out one of the r columns of $\mathbf{L}_{\Gamma 2}$ and producing the nonsingular lower triangular matrix $\mathbf{L}_\Gamma \in \mathcal{R}^{r\times r}$. Therefore, we obtain

$$\mathbf{\Gamma} = \mathbf{Q}_\Gamma \begin{bmatrix} \mathbf{L}_\Gamma^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P}_\Gamma^T = \mathbf{Y}_\Gamma [\mathbf{L}_\Gamma^T \quad \mathbf{0}] \mathbf{P}_\Gamma^T$$

with $\mathbf{P}_\Gamma = \mathbf{\Pi} \mathbf{P}_{\Gamma r} \cdots \mathbf{P}_{\Gamma 1} \in \mathcal{R}^{t\times t}$ and upper triangular $\mathbf{L}_\Gamma^T$. Applying each of the r Householder transformations $\mathbf{P}_j$ takes $2(r-j)(s+1)$ flops, which is in total $r^2 s$ flops. Therefore, the two steps together require $2t^2r - tr^2 - r^3/3$ flops.

3. *Equivalent problem: Update of Cholesky factor.* The problem of zeroing out the s rows in $\mathbf{L}_2$ is equivalent to a sequence of s rank-one updates of an anti-Cholesky ($\mathbf{UU}^T$) factor $\mathbf{L}_1$. Assuming we already have the factor $\mathbf{L}_1$ of a positive definite matrix $\mathbf{H}_1$, $\mathbf{H}_1 = \mathbf{L}_1^T \mathbf{L}_1$, and that we must compute the anti-Cholesky factor $\mathbf{L}$ of $\mathbf{H} = \mathbf{H}_1 + \mathbf{L}_2^T \mathbf{L}_2$,

$$\mathbf{H} = \mathbf{L}^T \mathbf{L} = \mathbf{H}_1 + \mathbf{L}_2^T \mathbf{L}_2 = [\mathbf{L}_1^T \quad \mathbf{L}_2^T] \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{bmatrix}.$$

We obtain $\mathbf{L}$ like $\mathbf{L}_\Gamma$ in the second step of the complete orthogonal decomposition by a sequence of r Householder transformations $\mathbf{P}_j$, which requires $r^2 s = r^2(n-m-r)$ flops.

4. *Computing the Moore–Penrose inverse $\mathbf{\Gamma}^\dagger$ explicitly.*

$$\mathbf{\Gamma}^\dagger = \mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{Q}_\Gamma^T = \mathbf{P}_\Gamma \begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1} \\ \mathbf{0} \end{bmatrix} \mathbf{Y}_\Gamma^T.$$

(i) Obtain $\mathbf{Y}_\Gamma$ in $\mathbf{Q}_\Gamma = [\mathbf{Y}_\Gamma \quad \mathbf{Z}_\Gamma]$ explicitly by applying the r Householder transformations obtained in the first step to

$$\begin{bmatrix} \mathbf{I}_r \\ \mathbf{0} \end{bmatrix}.$$

Forming $\mathbf{Y}_\Gamma$ requires $2tr^2 - r^3$ flops.

(ii) Solve the $r \times r$ lower triangular system $(\mathbf{L}_\Gamma^T)^{-1} \mathbf{Y}_\Gamma^T$ with t right-hand sides using backward substitution, which yields an $r \times t$ intermediate matrix. This takes $tr^2/2$ flops.

(iii) Left-apply the r Householder transformations in $\mathbf{P}_\Gamma$ on the $r \times t$ intermediate matrix

$$\begin{bmatrix} (\mathbf{L}_\Gamma^T)^{-1}\mathbf{Y}_\Gamma^T \\ \mathbf{0} \end{bmatrix},$$

which results in the symmetric matrix $\mathbf{\Gamma}^\dagger \in \mathcal{R}^{t\times t}$. This requires $2trs = 2t^2r - 2tr^2$ flops.

Given the complete orthogonal decomposition of $\mathbf{\Gamma}$, computing $\mathbf{\Gamma}^\dagger$ therefore amounts to $2t^2r + tr^2/2 - r^3$ flops.

Appendix 2: Second theorem. In this appendix we address the question of when $\mathbf{\Phi G \Phi} = \mathbf{\Phi}$ and show that this may be characterized by the vanishing of the matrix $\mathbf{X}_1$. Assuming the previous notation we have the following theorem.

THEOREM 6.1. *The following are equivalent:*

(i) $\mathbf{\Phi G \Phi} = \mathbf{\Phi}$, (ii) $\mathbf{X}_1 = \mathbf{0}$,
(iii) $\operatorname{rank}(\mathbf{\Gamma}) = \operatorname{rank}(\mathcal{G}\mathbf{Z})$, (iv) $\mathbf{\Phi} = \mathbf{Z}\mathbf{\Gamma}^\dagger\mathbf{Z}^T$.

Proof.
(i) $\Rightarrow$ (ii). Recall that

$$\mathcal{G} = \mathbf{Q}\begin{bmatrix} \mathbf{A} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{\Gamma} \end{bmatrix}\mathbf{Q}^T$$

and suppose that

$$\mathbf{\Phi} = \mathbf{Q}\begin{bmatrix} \mathbf{0} & \mathbf{X}^\dagger \\ (\mathbf{X}^\dagger)^T & \mathbf{S} \end{bmatrix}\mathbf{Q}^T,$$

where from (15) $\mathbf{S} = -(\mathbf{X}_1^\dagger)^T\mathbf{A}\mathbf{X}_1^\dagger + (\mathbf{I} - (\mathbf{X}_1^\dagger)^T\mathbf{E})\mathbf{\Gamma}^\dagger(\mathbf{I} - \mathbf{E}^T\mathbf{X}_1^\dagger)$. Then $\mathbf{Q}^T(\mathbf{\Phi G \Phi})\mathbf{Q} = \mathbf{Q}^T\mathbf{\Phi}\mathbf{Q}$ becomes

$$\begin{bmatrix} \mathbf{X}_1^\dagger\mathbf{\Gamma}(\mathbf{X}_1^\dagger)^T & \mathbf{X}_1^\dagger\mathbf{F} \\ \mathbf{F}^T(\mathbf{X}_1^\dagger)^T & (\mathbf{X}_1^\dagger)^T\mathbf{J} + \mathbf{S}\mathbf{F} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{X}_1^\dagger \\ (\mathbf{X}_1^\dagger)^T & \mathbf{S} \end{bmatrix}, \tag{19}$$

where $\mathbf{F} = \mathbf{E}^T\mathbf{X}_1^\dagger + \mathbf{\Gamma}\mathbf{S}$ and $\mathbf{J} = \mathbf{A}\mathbf{X}_1^\dagger + \mathbf{E}\mathbf{S}$. Now $\mathbf{X}_1^\dagger = \mathbf{H}^{-1}\mathbf{U}$, where $\mathbf{U} = \mathbf{E}(\mathbf{I} - \mathbf{\Gamma}\mathbf{\Gamma}^\dagger)$, and hence $\mathbf{X}_1^\dagger\mathbf{\Gamma} = \mathbf{0}$. Next equating (1,2) blocks in (19) shows that $\mathbf{X}_1^\dagger = \mathbf{X}_1^\dagger\mathbf{F} = \mathbf{X}_1^\dagger\mathbf{E}^T\mathbf{X}_1^\dagger$. This reduces to $\mathbf{U}\mathbf{E}^T\mathbf{H}^{-1}\mathbf{U} = \mathbf{U}$. But since $\mathbf{U}\mathbf{E} = \mathbf{U}\mathbf{U}^T$ we arrive at $\mathbf{U}\mathbf{U}^T\mathbf{H}^{-1}\mathbf{U} = \mathbf{U}$. Premultiplication by $\mathbf{U}^\dagger$ now gives $\mathbf{U}^T\mathbf{H}^{-1}\mathbf{U} = \mathbf{U}^\dagger\mathbf{U}$, which is idempotent. Thus $\mathbf{U}^T\mathbf{H}^{-1}\mathbf{U}\mathbf{U}^T\mathbf{H}^{-1}\mathbf{U} = \mathbf{U}^T\mathbf{H}^{-1}\mathbf{U}$, in which we substitute $\mathbf{H} - \mathbf{R}\mathbf{R}^T$ for $\mathbf{U}\mathbf{U}^T$, giving

$$\mathbf{U}^T\mathbf{H}^{-1}\mathbf{U} - (\mathbf{U}^T\mathbf{H}^{-1}\mathbf{R})(\mathbf{R}^T\mathbf{H}^{-1}\mathbf{U}) = \mathbf{U}^T\mathbf{H}^{-1}\mathbf{U}$$

or $(\mathbf{U}^T\mathbf{H}^{-1}\mathbf{R})(\mathbf{U}^T\mathbf{H}^{-1}\mathbf{R})^T = \mathbf{0}$. By the star cancellation law we end up with $\mathbf{U}^T\mathbf{H}^{-1}\mathbf{R} = \mathbf{0}$ forcing $\mathbf{U} = \mathbf{0}$. This clearly implies (ii).

(ii) $\Rightarrow$ (iii). Because $\mathbf{\Gamma}$ is symmetric, $\mathbf{\Gamma}\mathbf{\Gamma}^\dagger = \mathbf{\Gamma}^\dagger\mathbf{\Gamma}$ and $\mathbf{U} = \mathbf{E}(\mathbf{I} - \mathbf{\Gamma}\mathbf{\Gamma}^\dagger)$. This shows that $RS(\mathbf{E}) \subseteq RS(\mathbf{\Gamma})$ iff $\mathbf{U} = \mathbf{0}$ iff $\mathbf{X}_1 = \mathbf{0}$. Next recall that

$$\mathbf{Q}^T\mathcal{G}\mathbf{Z} = \begin{bmatrix} \mathbf{Y}^T \\ \mathbf{Z}^T \end{bmatrix}\mathcal{G}\mathbf{Z} = \begin{bmatrix} \mathbf{E} \\ \mathbf{\Gamma} \end{bmatrix},$$

from which follows that

$$RS(\mathcal{G}\mathbf{Z}) = RS\left(\begin{bmatrix} \mathbf{E} \\ \mathbf{\Gamma} \end{bmatrix}\right). \tag{20}$$

Now from (20) we see that if $RS(\mathbf{E}) \subseteq RS(\mathbf{\Gamma})$ then $RS(\mathcal{G}\mathbf{Z}) = RS(\mathbf{\Gamma})$ and conversely, as desired.

(ii) $\Rightarrow$ (iv). If $\mathbf{X}_1 = \mathbf{0}$, then $\mathbf{S} = \mathbf{\Gamma}^\dagger$ and $\mathbf{\Phi}$ collapses to

$$\mathbf{Q}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}^\dagger \end{bmatrix}\mathbf{Q}^T = \mathbf{Z}\mathbf{\Gamma}^\dagger\mathbf{Z}^T.$$

(iv) $\Rightarrow$ (i). It is easily seen that

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}^\dagger \end{bmatrix}\begin{bmatrix} \mathbf{A} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{\Gamma} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}^\dagger \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}^\dagger \end{bmatrix}. \qquad \square$$

Appendix 3: Constructing test matrices G and Z. Here we describe our method for constructing a column orthogonal matrix $\mathbf{Z} \in \mathcal{R}^{n\times t}$ that transforms an indefinite Hessian matrix $\mathcal{G}$ into a positive semidefinite projected Hessian matrix $\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z} \in \mathcal{R}^{t\times t}$ with prespecified rank r.

1. For simplicity, we chose the Hessian $\mathcal{G} \in \mathcal{R}^{n\times n}$ as a diagonal matrix:

$$\mathcal{G} = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{\Delta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{D} = \operatorname{diag}(d_1, \ldots, d_s), \quad \mathbf{\Delta} = \operatorname{diag}(\delta_1, \ldots, \delta_s)$$

with $d_i > 0$, and $\delta_i > 0$, for all $i = 1, \ldots, s$. Therefore, $\mathcal{G}$ has s eigenvalues d_i greater than zero, s eigenvalues $-\delta_i$ smaller than zero, and s zero eigenvalues. We randomly selected the elements d_i and δ_i within the interval $[.5, 1.5]$ and ensuring that $\min_{1\le i\le s}\{d_i/\delta_i\}$ is satisfied for only one index i. Later we will see that this fact determines the rank r of $\mathbf{\Gamma}$ to be $r = s - 1$.

2. We will now construct $\mathbf{Z} \in \mathcal{R}^{n\times t}$ so that

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \mathbf{Z}_3 \end{bmatrix}, \quad \text{with} \quad \mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3 \in \mathcal{R}^{s\times t},$$

resulting in a positive semidefinite $\mathbf{\Gamma}$,

$$\mathbf{\Gamma} = \mathbf{Z}^T\mathcal{G}\mathbf{Z} = \mathbf{Z}_1^T\mathbf{D}\mathbf{Z}_1 - \mathbf{Z}_2^T\mathbf{\Delta}\mathbf{Z}_2 \in \mathcal{R}^{t\times t},$$

and satisfying

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{Z}_1^T\mathbf{Z}_1 + \mathbf{Z}_2^T\mathbf{Z}_2 + \mathbf{Z}_3^T\mathbf{Z}_3 = I_t.$$

3. First construct $\mathbf{Z}_3$. We select $\mathbf{Z}_3 = [\mathbf{H} \quad \mathbf{K}] \in \mathcal{R}^{s\times t}$, where $\mathbf{H} \in \mathcal{R}^{s\times s}$ is a rank-one matrix and $\mathbf{K} \in \mathcal{R}^{s\times(t-s)}$ is column orthogonal. We set $\mathbf{H} = uv^T$, selecting $u, v \in \mathcal{R}^s$ with the property $\|u\|^2\|v\|^2 < 1$, ensuring that

$$\mathbf{I}_s - \mathbf{H}^T\mathbf{H} = \mathbf{I}_s - \|u\|^2 vv^T$$

is positive semidefinite because the eigenvalues of $\mathbf{H}^T\mathbf{H}$ are $\|u\|^2\|v\|^2$ and zero. In our examples we selected u and v randomly with $\|u\|^2 = \|v\|^2 = 1/2$. Using the Gram–Schmidt orthogonalization technique [4] we generate stepwise the $t - s$ orthogonal columns of $\mathbf{K} \in \mathcal{R}^{s\times(t-s)}$ so that $\mathbf{K}^T u = 0$ and $\mathbf{K}^T\mathbf{K} = \mathbf{I}_{t-s}$. Therefore,

$$\mathbf{Z}_3^T\mathbf{Z}_3 = \left[\begin{array}{c|c} \mathbf{H}^T\mathbf{H} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{t-s} \end{array}\right] \in \mathcal{R}^{t\times t}.$$

4. Now construct $\mathbf{Z}_1$ and $\mathbf{Z}_2$. Setting

$$\mathbf{Z}_1 = [\beta\mathbf{P} \quad \mathbf{0}] \quad \text{and} \quad \mathbf{Z}_2 = [\gamma\mathbf{P} \quad \mathbf{0}]$$

with $\mathbf{P} \in \mathcal{R}^{s\times s}$ and scalars β and γ we obtain

$$\mathbf{\Gamma} = \mathbf{Z}_1^T\mathbf{D}\mathbf{Z}_1 - \mathbf{Z}_2^T\mathbf{\Delta}\mathbf{Z}_2 = \left[\begin{array}{c|c} \mathbf{P}^T(\beta^2\mathbf{D} - \gamma^2\mathbf{\Delta})\mathbf{P} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right],$$

and we choose β and γ so that $\mathbf{P}^T(\beta^2\mathbf{D} - \gamma^2\mathbf{\Delta})\mathbf{P}$ is positive semidefinite. For $\beta = 1$ we obtain

$$\gamma = \min_{1\le i\le s} \sqrt{\frac{d_i}{\delta_i}}.$$

The number of indices i for which this minimum is satisfied determines the rank of $\mathbf{P}^T(\beta^2\mathbf{D} - \gamma^2\mathbf{\Delta})\mathbf{P}$ and therefore the rank of $\mathbf{\Gamma}$. We now select $\mathbf{P}$ so that $\mathbf{Z}^T\mathbf{Z} = I_t$, i.e.,

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{Z}_1^T\mathbf{Z}_1 + \mathbf{Z}_2^T\mathbf{Z}_2 + \mathbf{Z}_3^T\mathbf{Z}_3 = \left[\begin{array}{c|c} (1+\gamma^2)\mathbf{P}^T\mathbf{P} + \mathbf{H}^T\mathbf{H} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{2s} \end{array}\right],$$

and we obtain

$$\mathbf{P} = \left[\frac{\mathbf{I} - \mathbf{H}^T\mathbf{H}}{1+\gamma^2}\right]^{1/2} = \mathbf{P}^T.$$

Since the eigenvalues of $\mathbf{I} - \mathbf{H}^T\mathbf{H}$ are $\{1 - \{0, \|u\|^2\|v\|^2\}\}$ and $\|u\|^2\|v\|^2 < 1$ by definition, the matrix $\mathbf{\Gamma}$ is positive semidefinite and $\mathbf{Z}^T\mathbf{Z} = I_t$.

Acknowledgments. We wish to thank Linda Kaufman for many helpful comments which resulted in a greatly improved paper.

REFERENCES

[1] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*, John Wiley, New York, 1974.

[2] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 31 (1977), pp. 163–179.

[3] J. S. Cramer, *Econometric Applications of Maximum Likelihood Methods*, Cambridge University Press, Cambridge, UK, 1966.

[4] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[5] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[6] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, UK, 1987.

[7] A. R. Gallant, *Nonlinear Statistical Models*, John Wiley, New York, 1987.

[8] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Procedures for optimization problems with a mixture of bounds and general linear constraints*, ACM Trans. Math. Software, 10 (1984), pp. 282–298.

[9] P. E. Gill, S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for LSSOL (Version 1.0): A Fortran Package for Constrained Linear Least-Squares and Convex Quadratic Programming*, Tech. report SOL 86-1, Department for Operations Research, Stanford University, Stanford, CA, 1986.

[10] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming*, Tech. report SOL 86-2, Department for Operations Research, Stanford University, Stanford, CA, 1986.

[11] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Numerical Linear Algebra and Optimization*, Vol. 1, Addison–Wesley, Redwood City, CA, 1991.
[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[13] J. H. GOODNIGHT, *The Sweep Operator: Its Importance in Statistical Computing*, SAS Tech. report R-106, SAS Institute, Inc., Cary, NC, 1978.
[14] N. M. GOULD, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic problem*, Math. Prog., 32 (1985), pp. 90–99.
[15] W. M. HARTMANN, *The NLP Procedure: Extended User's Guide*, SAS Institute, Inc., Cary, NC, 1992.
[16] R. E. HARTWIG, *Rank factorization and Moore–Penrose inversion*, Indust. Math., 26 (1976), pp. 49–63.
[17] J. O. RAMSAY, *Confidence regions for multidimensional scaling analysis*, Psychometrika, 43 (1978), pp. 145–160.
[18] E. L. STIEFEL, *Numerical Analysis*, Academic Press, New York, 1966.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 748–768, August 1996
© 1996 Society for Industrial and Applied Mathematics
013

THE EFFECTIVE ENERGY TRANSFORMATION SCHEME AS A SPECIAL CONTINUATION APPROACH TO GLOBAL OPTIMIZATION WITH APPLICATION TO MOLECULAR CONFORMATION*

ZHIJUN WU†

Abstract. This paper discusses a generalization of the function transformation scheme used in Coleman, Shalloway, and Wu [*Comput. Optim. Appl.*, 2 (1993), pp. 145–170; *J. Global Optim.*, 4 (1994), pp. 171–185] and Shalloway [*Global Optimization*, C. Floudas and P. Pardalos, eds., Princeton University Press, 1992, pp. 433–477; *Global Optim.*, 2 (1992), pp. 281–311] for global energy minimization applied to the molecular conformation problem. A mathematical theory for the method as a special continuation approach to global optimization is established. We show that the method can transform a nonlinear objective function into a class of gradually deformed, but "smoother" or "easier" functions. An optimization procedure can then be successively applied to the new functions to trace their solutions back to the original function. Two types of transformation are defined: isotropic and anisotropic. We show that both transformations can be applied to a large class of nonlinear partially separable functions, including energy functions for molecular conformation. Methods to compute the transformation for these functions are given.

Key words. global/local minimization, continuation methods, integral transformation, molecular conformation

AMS subject classifications. 49M37, 68Q22, 92C40

1. Introduction.

We are interested in solving global minimization problems for molecular conformation, especially protein folding.

How proteins fold is one of the key biophysical problems of the decade. This problem is fundamental for almost all theoretical studies of proteins and protein-related life processes. It has many applications in the biotechnology industry, such as the structure-based drug design for the treatment of important diseases like polio, cancer, and AIDS.

Optimization approaches to the protein folding problem are based on the hypothesis that the protein native structure corresponds to the global minimum of the protein energy. The problem can be attacked computationally by minimizing the protein energy over all possible protein structures. The structure with the lowest energy is presumed to be the most stable protein structure.

Mathematically, for a protein molecule of n atoms, let $x = \{x_i \in \mathbf{R}^3,\ i = 1, \ldots, n\}$ represent the molecular structure with each x_i specifying the spatial position of atom i. Then the energy minimization problem for protein folding is to globally minimize a nonlinear function $f(x)$ for all $x \in \mathbf{S}$, i.e.,

$$\min_{x \in \mathbf{S}} \quad f(x), \tag{1}$$

where $\mathbf{S}$ is the set of all possible molecular structures. The objective function $f(x)$ is the energy function for the protein. An important energy function type considered in

* Received by the editors August 16, 1993; accepted for publication (in revised form) January 19, 1995. This research was partially supported by the Cornell Theory Center, which receives funding from members of its Corporate Research Institute, the National Science Foundation, the Advanced Research Projects Agency, the National Institutes of Health, New York State, and IBM Corporation.

† Advanced Computing Research Institute, Cornell University, Ithaca, NY 14853. Current address: Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (zwu@mcs.anl.gov).

this paper is of the pairwise interaction form

$$f(x) = \sum_{i=1,j>i}^{n} h_{ij}(\|r_{ij}\|), \tag{2}$$

where $r_{ij} = x_i - x_j$ and h_{ij} is the pairwise energy function determined by $\|r_{ij}\|$, the distance between atoms i and j. Most potential energy terms for proteins can be written in form (2), such as the potentials for nonbonded interactions, the bonded potentials of bond lengths, and the potentials for hydrogen bonding. A widely used expression for nonbonded interactions, for example, is of the following Lennard–Jones plus Coulomb form:

$$h_{ij}(\|r_{ij}\|) = \epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{\|r_{ij}\|}\right)^{12} - 2\left(\frac{\sigma_{ij}}{\|r_{ij}\|}\right)^{6}\right] + \frac{q_i q_j}{\epsilon\|r_{ij}\|}, \tag{3}$$

where ϵ_{ij}, σ_{ij}, ϵ, q_i, and q_j are all physical constants (see [2]).

Problem (1) is very difficult to solve in general. The reasons are as follows. First, in theory even simple versions of the problem have been proven to be $\mathcal{NP}$-complete [9]. Second, in practice the objective function often contains exponentially many local minimizers; therefore, the search for the global minimizer can be computationally intractable. Finally, the protein molecules tend to be very large, typically containing $\mathcal{O}(10{,}000)$ atoms. For such large problems, the required computation is unaffordable using general global optimization methods.

However, because of its great practical importance, problem (1) has been intensively studied in many areas of computational science and optimization. New algorithms on both sequential and parallel machines have been developed; a variety of small to medium sizes of problems have been studied [3]–[6], [11]–[16], [18]–[22]. In recent efforts smoothing techniques have been specifically designed for molecular conformation via global minimization. Examples include the diffusion equation method [11, 12, 15], and the packet annealing method [19, 20], as well as the effective energy simulated annealing method [4, 5]. The basic idea behind these methods is to use special techniques to smooth a given energy function so that the search for a global minimizer becomes more tractable. The methods usually use function transformation schemes to transform a given energy function into a class of new functions. A solution tracing procedure is then applied to the new functions to locate a solution for the original function.

In this paper, we discuss an important generalization of the effective energy transformation scheme introduced in [4, 5, 19, 20]. Instead of applying the transformation to the probability distribution function, we transform the functions directly, generalizing the method to a broader class of functions. More importantly, with this generalization, a mathematical theory for the transformation as a special continuation approach to global optimization is established. We show that the method can transform a nonlinear objective function into a class of gradually deformed, but "smoother" or "easier" functions. An optimization procedure can then be applied to the new functions successively to trace their solutions back to the original function. Two types of transformation are defined: isotropic and anisotropic. We show that both transformation types can be applied to a large class of nonlinear partially separable functions, which includes typical energy functions for molecular conformation. Methods to compute the transformation for these functions are given.

The paper is organized as follows. Section 2 introduces the basic approach and describes the function transformation method. Section 3 studies the mathematical properties of the transformation as a special continuation process. Section 4 characterizes the "smoothness" property and shows that the transformed function becomes "smoother" in the sense that the small high frequency variations in the original function are averaged out after the transformation. The numerical applicability of the transformation is discussed in §5. The transformation is extended to the anisotropic type in §6. The formulas to compute the transformation for molecular conformation energy functions are derived. Finally, §7 contains concluding remarks.

2. The approach. In this section, we describe our function transformation idea which in turn defines our basic approach to global optimization.

Suppose that we have a "poorly-behaved" nonlinear function with many local minimizers. Because of "nonsmoothness," this type of function can be very hard to minimize either locally or globally. To overcome this difficulty, we suggest using a special technique to transform the objective function into a class of gradually deformed, but "smoother" or "easier" functions. An optimization procedure can then be applied to these new functions successively to trace their solutions back to the original function.

To deform the function, we define a parametrized integral transformation as follows.

DEFINITION 1. *Given a nonlinear function f, the transformation $\langle f\rangle_\lambda$ for f is defined such that for all x,*

$$\langle f\rangle_\lambda(x) = C_\lambda \int f(x')\, e^{-\|x-x'\|^2/\lambda^2}\, dx' \tag{4}$$

or, equivalently,

$$\langle f\rangle_\lambda(x) = C_\lambda \int f(x-x')\, e^{-\|x'\|^2/\lambda^2}\, dx', \tag{5}$$

where λ is a positive number and C_λ is a normalization constant such that

$$C_\lambda \int e^{-\|x\|^2/\lambda^2}\, dx = 1. \tag{6}$$

Note that in contrast to the approaches in [4, 5, 19, 20], the transformation here applies directly to the given function instead of its probability distribution. This simplifies the transformation and also makes it much easier to compute and analyze.

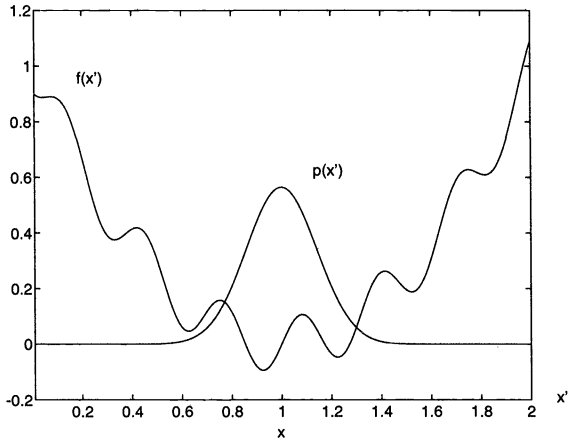
To understand this transformation, consider that given a random function $g(x')$ and a probability distribution function $p(x')$ for the random variable x', the expectation of the function g with respect to p is

$$\langle g\rangle_p = \int g(x')\, p(x')\, dx'. \tag{7}$$

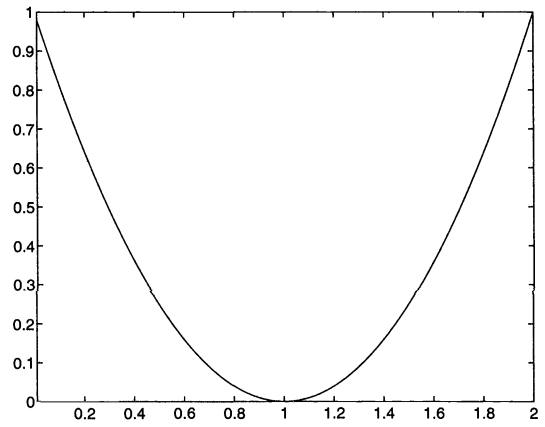
In light of (7), transformation (4) yields a function value for $\langle f\rangle_\lambda$ at any x equal to the expectation for f sampled by a Gaussian distribution function centered at x.

For example, consider the following nonlinear function:

$$f(x) = (x-1)^2 + 0.1\sin 20(x-1), \tag{8}$$



(a)



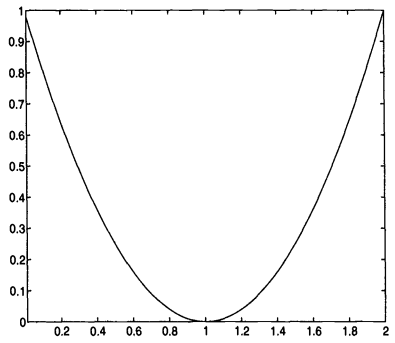
(b)

FIG. 1. *A one-dimensional transformation example.*

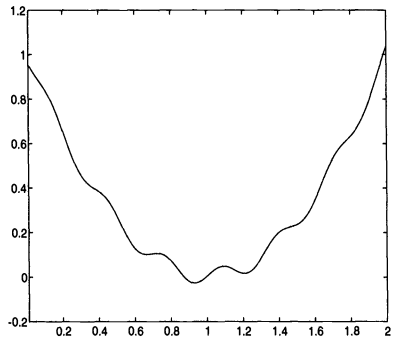
which is a quadratic function augmented with a "noise" function. The transformation for this function can be computed as follows:

$$\langle f\rangle_\lambda(x) = (x-1)^2 + \frac{\lambda^2}{2} + 0.1e^{-(20\lambda)^2/4}\sin 20(x-1). \tag{9}$$

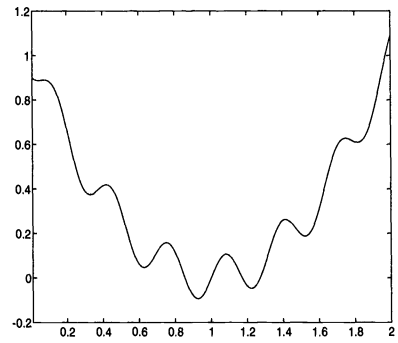
The function value $\langle f\rangle_\lambda(x)$ for fixed x is equal to the integration with respect to the product of two functions, the original function $f(x')$, and the Gaussian distribution function $p(x') = \mathcal{C}_\lambda e^{-\|x-x'\|^2/\lambda^2}$ (see Figure 1 (a)). The parameter λ determines the size of the dominant region of the Gaussian. Since the most significant part of the integration is that within the dominant region of the Gaussian, $\langle f\rangle_\lambda(x)$ can be viewed as the average value for the original function f within a small λ-neighborhood around x. If λ is equal to zero the transformed function is exactly the original function. For positive λ, the original function variations in small regions are averaged out, and the transformed function will become "smoother" (see Figure 1 (b)).



(a) $\lambda = 0.2$



(b) $\lambda = 0.1$



(c) $\lambda = 0.0$

FIG. 2. *A class of gradually deformed functions.*

Figure 2 shows how the function $\langle f \rangle_\lambda$ in (9) behaves with increasing λ. Observe that when $\lambda = 0.0$ the function is the original function; when we increase λ to 0.1, the function becomes "smoother"; when λ is increased further to 0.2, the function becomes entirely "smooth."

Figure 3 illustrates what the transformation implies for optimization. A standard optimization procedure, a quasi-Newton method [7, 8], is applied to the three functions in Figure 2. Figure 3 (a), (b), and (c) contain the corresponding solutions x^* obtained with different choices of initial guesses x°. Although globally convergent, the method

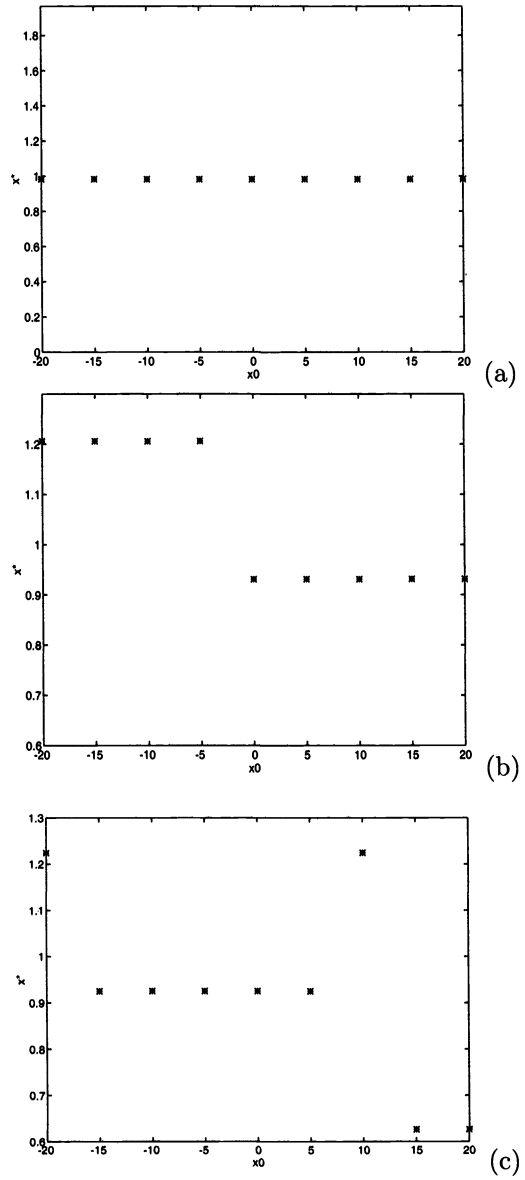


FIG. 3. *The solutions for the functions in Figure 2 obtained by the quasi-Newton method with different initial guesses.*

may not find the right solution if the "noise" is large. So for the function in Figure 2 (c), the method converged to the right solution only when the initial guess was close enough to the solution. When the initial guess was far from the solution, the method failed to find the right solution (see Figure 3 (c)). Although the function in Figure 2 (b) is "smoother", the behavior of the method was essentially unchanged. However, for the function in Figure 2 (a), the method always converged to the right solution (see Figure 3 (a)). If we apply the method to the functions in Figure 2 (a) to (c)

successively, and at each step take the solution for the previous function as the starting point, the solutions for all these functions can then be obtained.

The previous experiment suggests a general global optimization method: to optimize a difficult function, use the transformation technique to deform the function into a class of "smoother" or "easier" functions, and then apply an optimization procedure to the functions successively to trace their solutions back to the original function.

3. Continuation. What is the difference between the suggested approach and general homotopy methods? The answer is that this approach is indeed a special type of homotopy method. But the transformation is different from conventional homotopies and has the following three special features. First, the transformed functions are not arbitrarily deformed functions. They all are approximations to the original function in the sense that they are coarse estimates. Second, the transformation is defined by a special parametrized integral transformation. If we increase the value of the parameter, the transformed function will become "smoother" with small variations gradually removed, but maintaining the overall function structure. Finally, if we apply an optimization procedure to a transformed function, the obtained solution usually is close to the solution for the original function. All these features are good for global optimization (also for robust local optimization), but are not necessarily the properties of conventional homotopies.

We show in the following that the proposed transformation is indeed a well-defined homotopy and determines for any initial solution a unique solution curve containing the stationary points for the transformed functions.

ASSUMPTION 1. *The objective function f is twice continuously differentiable, and transformation* (4) *is well defined for the function as well as all its derivatives.*

ASSUMPTION 2. *Let g be the gradient of f and Δ the Laplace operator*

$$\Delta = \sum_i \frac{\partial^2}{\partial x_i^2}. \tag{10}$$

Then the operation Δ can be applied to g, and transformation (4) *is well defined for all derivatives involved. Also, $\Delta g(x)$ is uniformly bounded and satisfies a Lipschitz condition:*

$$\|\Delta g(x_1) - \Delta g(x_2)\| \leq L \|x_1 - x_2\| \qquad \forall x_1,\ x_2. \tag{11}$$

ASSUMPTION 3. *The transformation $\langle \nabla^2 f \rangle_\lambda(x)$ satisfies a Lipschitz condition:*

$$\|\langle \nabla^2 f \rangle_\lambda(x_1) - \langle \nabla^2 f \rangle_\lambda(x_2)\| \leq L \|x_1 - x_2\| \qquad \forall x_1,\ x_2, \tag{12}$$

and its inverse is uniformly bounded.

Note that to guarantee Assumptions 1 to 3, a sufficient condition on f is that f and its derivatives are all integrable in terms of parametrized integration (4).

We first state two sets of standard results for the proposed transformation in the following lemmas without proof.

LEMMA 1. *Under Assumption* 1, $\forall \lambda, x$,

$$\nabla \langle f \rangle_\lambda(x) = \langle \nabla f \rangle_\lambda(x), \tag{13}$$

$$\nabla^2 \langle f \rangle_\lambda(x) = \langle \nabla^2 f \rangle_\lambda(x). \tag{14}$$

LEMMA 2. *Under Assumption* 1, $\forall x$,

$$\lim_{\lambda \to 0} \langle f \rangle_\lambda(x) = f(x), \tag{15}$$

$$\lim_{\lambda \to 0} \nabla \langle f \rangle_\lambda(x) = \nabla f(x), \tag{16}$$

$$\lim_{\lambda \to 0} \nabla^2 \langle f \rangle_\lambda(x) = \nabla^2 f(x). \tag{17}$$

For convenience, we define a function $h(\lambda, x)$, $\lambda \in \Gamma$ and $x \in S$, such that $\forall (\lambda, x) \in \Gamma \times S$,

$$h(\lambda, x) = \langle f \rangle_\lambda(x), \tag{18}$$

where $\Gamma = [0, \lambda_0]$ for some $\lambda_0 < \infty$ and S is a vector space. With this definition, the condition for x to be a stationary point of $\langle f \rangle_\lambda$ for some $\lambda \in \Gamma$ is

$$h'_x(\lambda, x) = 0. \tag{19}$$

THEOREM 1. *Let* $f : \mathbf{R} \to \mathbf{R}$ *and* h *be defined as in* (18). *Then under Assumptions* 1 *and* 2, $h''_{x\lambda}(\lambda, x)$ *exists and is uniformly bounded for all* $\lambda \in \Gamma$ *and* $x \in S$, *and also satisfies a Lipschitz condition in* x:

$$|h''_{x\lambda}(\lambda, x_1) - h''_{x\lambda}(\lambda, x_2)| \leq L\ |x_1 - x_2| \qquad \forall x_1, x_2 \in S. \tag{20}$$

In addition,

$$h''_{x\lambda}(\lambda, x) = \frac{\lambda}{2} \langle f''' \rangle_\lambda(x). \tag{21}$$

Proof. Let $p(\lambda, x)$ be the Gaussian distribution function defined as follows:

$$p(\lambda, x) = c_\lambda\ e^{-x^2/\lambda^2}, \tag{22}$$

where $c_\lambda = 1/(\lambda\sqrt{\pi})$. Then by the definition of $\langle f \rangle_\lambda$,

$$h(\lambda, x) = \langle f \rangle_\lambda(x) = \int f(x') p(\lambda, (x - x'))dx'. \tag{23}$$

By Lemma 1,

$$h'_x(\lambda, x) = \int g(x') p(\lambda, (x - x'))dx'. \tag{24}$$

After differentiating (24) with respect to λ, it follows that

$$\begin{aligned} h''_{x\lambda}(\lambda, x) &= \int g(x') \left(\frac{2(x - x')^2}{\lambda^3} - \frac{1}{\lambda} \right) p(\lambda, (x - x'))dx' \\ &= \frac{1}{\lambda} \int \left(\frac{2}{\lambda^2}(x - x')^2 - 1 \right) g(x') p(\lambda, (x - x'))dx' \\ &= \frac{1}{\lambda} \left(\frac{2}{\lambda^2}(\Delta_1 - \Delta_2 - \langle g \rangle_\lambda(x) \right), \end{aligned} \tag{25}$$

where

$$\Delta_1 = \int x(x-x')g(x')p(\lambda,(x-x'))dx', \tag{26}$$

$$\Delta_2 = \int x'(x-x')g(x')p(\lambda,(x-x'))dx'. \tag{27}$$

It is easy to verify that

$$\Delta_1 = -\frac{\lambda^2}{2}x\langle g'\rangle_\lambda(x), \tag{28}$$

$$\Delta_2 = -\frac{\lambda^2}{2}(\langle g\rangle_\lambda(x) + \langle xg'\rangle_\lambda(x)). \tag{29}$$

Also note that

$$\begin{aligned} &x\langle g'\rangle_\lambda(x) - \langle xg'\rangle_\lambda(x) \\ &\quad = \int (x-x')g'(x')p(\lambda,(x-x'))dx' \\ &\quad = \frac{\lambda^2}{2}\int g'(x')dp(\lambda,(x-x')) \\ &\quad = \frac{\lambda^2}{2}\left(g'(x')p(\lambda,(x-x'))|_{-\infty}^{+\infty} - \int g''(x')p(\lambda,(x-x'))dx'\right) \\ &\quad = -\frac{\lambda^2}{2}\langle g''\rangle(x). \end{aligned} \tag{30}$$

Therefore,

$$\begin{aligned} \Delta_1 - \Delta_2 &= -\frac{\lambda^2}{2}x\langle g'\rangle_\lambda(x) + \frac{\lambda^2}{2}(\langle g\rangle_\lambda(x) + \langle xg'\rangle_\lambda(x)) \\ &= -\frac{\lambda^2}{2}(x\langle g'\rangle_\lambda(x) - \langle xg'\rangle_\lambda(x)) + \frac{\lambda^2}{2}\langle g\rangle_\lambda(x) \\ &= \frac{\lambda^4}{4}\langle g''\rangle_\lambda(x) + \frac{\lambda^2}{2}\langle g\rangle_\lambda(x). \end{aligned} \tag{31}$$

Replacing $(\Delta_1 - \Delta_2)$ in (25) by (31), we see that

$$\begin{aligned} h''_{x\lambda}(\lambda,x) &= \frac{1}{\lambda}\left(\frac{\lambda^2}{2}\langle g''\rangle_\lambda(x) + \langle g\rangle_\lambda(x) - \langle g\rangle_\lambda(x)\right) \\ &= \frac{\lambda}{2}\langle f'''\rangle_\lambda(x). \end{aligned} \tag{32}$$

By Assumption 2, $\langle f'''\rangle_\lambda(x)$ is well defined and uniformly bounded. So, $h''_{x\lambda}(\lambda,x)$ exists and is uniformly bounded for all $\lambda \in \Gamma$ and $x \in S$. Since $f'''(x)$ satisfies a Lipschitz condition by Assumption 2, $\langle f'''\rangle_\lambda(x)$ also satisfies a Lipschitz condition:

$$|\langle f'''\rangle_\lambda(x_1) - \langle f'''\rangle_\lambda(x_2)| \le L_0\,|x_1 - x_2| \qquad \forall x_1,\ x_2. \tag{33}$$

Let $L = \frac{\lambda_0}{2}L_0$. Then (20) follows immediately. So $h''_{x\lambda}(\lambda,x)$ satisfies a Lipschitz condition in x. □

THEOREM 2. *Let $f : \mathbf{R}^n \to \mathbf{R}$ and h be defined as in* (18). *Then under Assumptions* 1 *and* 2, $h''_{x\lambda}(\lambda, x)$ *exists and is uniformly bounded for all* $\lambda \in \Gamma$ *and* $x \in S$, *and also satisfies a Lipschitz condition in* x:

$$\|h''_{x\lambda}(\lambda, x_1) - h''_{x\lambda}(\lambda, x_2)\| \leq L \, \|x_1 - x_2\| \qquad \forall x_1, x_2 \in S. \tag{34}$$

In addition,

$$h''_{x\lambda}(\lambda, x) = \frac{\lambda}{2} \langle \Delta g \rangle_\lambda(x). \tag{35}$$

Proof. Let $p_n(\lambda, x)$ be the Gaussian distribution function

$$p_n(\lambda, x) = c_\lambda^n \, e^{-\|x\|^2/\lambda^2}, \tag{36}$$

where $c_\lambda^n = 1/(\lambda\sqrt{\pi})^n$. Then by the definition of $\langle f \rangle_\lambda$,

$$h(\lambda, x) = \langle f \rangle_\lambda(x) = \int f(x') p_n(\lambda, (x - x')) dx'. \tag{37}$$

By Lemma 1,

$$h'_x(\lambda, x) = \int g(x') p_n(\lambda, (x - x')) dx'. \tag{38}$$

Differentiate (38) with respect to λ to obtain

$$\begin{aligned} h''_{x\lambda}(\lambda, x) &= \int g(x') \left(\frac{2\|x - x'\|^2}{\lambda^3} - \frac{n}{\lambda} \right) p_n(\lambda, (x - x')) dx' \\ &= \frac{1}{\lambda} \left(\frac{2}{\lambda^2} \sigma - n \langle g \rangle_\lambda(x) \right), \end{aligned} \tag{39}$$

where

$$\begin{aligned} \sigma &= \int \|x - x'\|^2 g(x') p_n(\lambda, (x - x')) dx' \\ &= \sum_{i=1}^n \int (x_i - x'_i)^2 g(x') p_n(\lambda, (x - x')) dx' \\ &= \sum_{i=1}^n \sigma_i. \end{aligned} \tag{40}$$

Let $\bar{x}_i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)^T$. Then

$$\begin{aligned} \sigma_i &= \int \left(\int (x_i - x'_i)^2 g(x'_i, \bar{x}'_i) p(\lambda, (x_i - x'_i)) dx'_i \right) \\ &\quad \cdot p_{n-1}(\lambda, (\bar{x}_i - \bar{x}'_i)) d\bar{x}'_i \\ &= \int (\Delta_{1i} - \Delta_{2i}) p_{n-1}(\lambda, (\bar{x}_i - \bar{x}'_i)) d\bar{x}'_i, \end{aligned} \tag{41}$$

where

$$\Delta_{1i} = \int x_i (x_i - x'_i) g(x'_i, \bar{x}'_i) p(\lambda, (x_i - x'_i)) dx'_i, \tag{42}$$

$$\Delta_{2i} = \int x'_i (x_i - x'_i) g(x'_i, \bar{x}'_i) p(\lambda, (x_i - x'_i)) dx'_i. \tag{43}$$

From the proof of Theorem 1,

$$\Delta_{1i} - \Delta_{2i} = \frac{\lambda^4}{4} \int \frac{\partial^2 g}{\partial x_i^2}(x_i', \bar{x}_i') p(\lambda, (x_i - x_i'))dx_i'$$

$$+ \frac{\lambda^2}{2} \int g(x_i', \bar{x}_i') p(\lambda, (x_i - x_i'))dx_i'. \tag{44}$$

Substitute (44) back into (41) to obtain

$$\sigma_i = \frac{\lambda^4}{4} \left\langle \frac{\partial^2 g}{\partial x_i^2} \right\rangle_\lambda (x) + \frac{\lambda^2}{2} \langle g \rangle_\lambda (x). \tag{45}$$

Then

$$\sigma = \frac{\lambda^4}{4} \sum_{i=1}^{n} \left\langle \frac{\partial^2 g}{\partial x_i^2} \right\rangle_\lambda (x) + \frac{n\lambda^2}{2} \langle g \rangle_\lambda (x) \tag{46}$$

and

$$\begin{aligned} h''_{x\lambda}(\lambda, x) &= \frac{\lambda}{2} \sum_{i=1}^{n} \left\langle \frac{\partial^2 g}{\partial x_i^2} \right\rangle_\lambda (x) \\ &= \frac{\lambda}{2} \left\langle \sum_{i=1}^{n} \frac{\partial^2 g}{\partial x_i^2} \right\rangle_\lambda (x) \\ &= \frac{\lambda}{2} \langle \Delta g \rangle_\lambda (x). \end{aligned} \tag{47}$$

Similar to the proof of Theorem 1, it follows immediately that $h''_{x\lambda}(\lambda, x)$ exists and is uniformly bounded for all $\lambda \in \Gamma$ and $x \in S$, and also satisfies a Lipschitz condition in x. □

Finally we state and prove the main theorem in this section as follows.

THEOREM 3. *Let f be a function for which Assumptions 1, 2, and 3 all hold. Then for any stationary point x_0 of $\langle f \rangle_{\lambda_0}$, there is a continuous and differentiable curve $x(\lambda)$, $\lambda \in \Gamma$, such that $x_0 = x(\lambda_0)$ and $\forall \lambda \in \Gamma$, $x(\lambda)$ is a stationary point of $\langle f \rangle_\lambda$. The curve $x(\lambda)$ is also the unique solution of the initial value problem*

$$x' = -\frac{\lambda}{2} \langle \nabla^2 f \rangle_\lambda^{-1}(x) \langle \Delta g \rangle_\lambda (x), \tag{48}$$

$$x_0 = x(\lambda_0). \tag{49}$$

Proof. Since x_0 is a stationary point of $\langle f \rangle_{\lambda_0}$,

$$h'_x(\lambda_0, x_0) = 0. \tag{50}$$

By Assumptions 1, 2, 3, Lemmas 1, 2, and Theorem 2, function h'_x is continuously differentiable at all $(\lambda, x) \in \Gamma \times S$. So by the implicit function theorem, there is a continuously differentiable function $x(\lambda)$ at a neighborhood of λ_0, such that $x_0 = x(\lambda_0)$ and

$$h'_x(\lambda, x(\lambda)) = 0 \tag{51}$$

for all λ in the neighborhood.

We now show that $x(\lambda)$ is also defined uniquely in Γ.

By differentiating (51), we see that $x(\lambda)$ is a solution to the following initial value problem:

$$x' = -h''_{xx}(\lambda, x)^{-1} h''_{x\lambda}(\lambda, x), \tag{52}$$

$$x_0 = x(\lambda_0), \tag{53}$$

which, by Lemma 1 and Theorem 2, is equivalent to the problem (48)–(49). Then it suffices to show that the right-hand side of (52) satisfies a Lipschitz condition in x on $\Gamma \times S$, which guarantees a unique solution $x(\lambda)$ in Γ by the standard ordinary differential equation theory [10].

Under Assumption 3, for $h''_{xx}(\lambda, x)^{-1}$, $\exists M_1, L_1$, such that

$$\|h''_{xx}(\lambda, x)^{-1}\| \leq M_1 \qquad \forall (\lambda, x) \in \Gamma \times S, \tag{54}$$

$$\|h''_{xx}(\lambda, x_1)^{-1} - h''_{xx}(\lambda, x_2)^{-1}\| \leq L_1 \, \|x_1 - x_2\| \qquad \forall x_1, x_2 \in S. \tag{55}$$

By Theorem 2, for $h''_{x\lambda}(\lambda, x)$, $\exists M_2, L_2$, such that

$$\|h''_{x\lambda}(\lambda, x)\| \leq M_2 \qquad \forall (\lambda, x) \in \Gamma \times S, \tag{56}$$

$$\|h''_{x\lambda}(\lambda, x_1) - h''_{x\lambda}(\lambda, x_2)\| \leq L_2 \, \|x_1 - x_2\| \qquad \forall x_1, x_2 \in S. \tag{57}$$

Let

$$G(\lambda, x) = -h''_{xx}(\lambda, x)^{-1} h''_{x\lambda}(\lambda, x). \tag{58}$$

Then it is easy to verify that $G(\lambda, x)$ satisfies a Lipschitz condition in x on $\Gamma \times S$:

$$\|G(\lambda, x_1) - G(\lambda, x_2)\| \leq L \, \|x_1 - x_2\| \qquad \forall x_1, x_2 \in S \tag{59}$$

with $L = M_1 L_2 + M_2 L_1$, which completes the proof. □

4. Smoothness. In §2 we illustrated that the transformed functions are "smoother" than the original function in the sense that they vary slower and may even have fewer local minimizers. In the following, we more rigorously characterize the "smoothness" of the transformation.

Let $\hat{f}$ be the Fourier transformation for function f and $\widehat{\langle f \rangle_\lambda}$ for function $\langle f \rangle_\lambda$. Recall that the transformation $\langle f \rangle_\lambda$ for f is just a convolution of f and p, where p is the Gaussian distribution function

$$p(x) = C_\lambda \, e^{-\|x\|^2/\lambda^2}. \tag{60}$$

Therefore, the Fourier transformation of $\langle f \rangle_\lambda$ is equal to the product of the Fourier transformations of f and p. The Fourier transformation of the Gaussian distribution function is

$$\hat{g}(\omega) = e^{-\frac{\lambda^2 \|\omega\|^2}{4}}, \tag{61}$$

where ω is the frequency. So, we have

$$\widehat{\langle f \rangle_\lambda}(\omega) = e^{-\frac{\lambda^2 \|\omega\|^2}{4}} \hat{f}(\omega). \tag{62}$$

We see from (62) that if $\lambda \to 0$ then $\widehat{\langle f \rangle_\lambda}$ converges to $\hat{f}$ and $\langle f \rangle_\lambda$ converges to f. This is exactly the fact we stated in Lemma 2.

Also by (62), $\widehat{\langle f\rangle_\lambda}(\omega)$ will be very small if ω is large and λ is fixed. This implies that high-frequency components of the original function become very small after the transformation. This is why the transformed function is "smoother." In addition, for larger λ values, wider ranges of high-frequency components of the original function practically vanish after the transformation and, therefore, the transformed function becomes increasingly smooth as λ increases. We state these properties formally in the following theorem.

THEOREM 4. *Let f, $\hat{f}$, $\langle f\rangle_\lambda$, and $\widehat{\langle f\rangle_\lambda}$ all be given and well defined. Then $\forall\varepsilon > 0$, $\exists\delta \propto 1/\lambda$ for fixed λ, such that $\forall\omega$ with $\|\omega\| > \delta$,*

$$\frac{|\widehat{\langle f\rangle_\lambda}(\omega)|}{|\hat{f}(\omega)|} < \varepsilon. \tag{63}$$

Proof. Given $\varepsilon > 0$, for fixed λ, let $\delta = 2\sqrt{\ln(1/\varepsilon)}/\lambda$. Then $\forall\omega$ with $\|\omega\| > \delta$,

$$\frac{|\widehat{\langle f\rangle_\lambda}(\omega)|}{|\hat{f}(\omega)|} = e^{-\frac{\lambda^2\|\omega\|^2}{4}} < e^{-\frac{\lambda^2\delta^2}{4}} = \varepsilon. \qquad \square \tag{64}$$

From this theorem we learn that the relative size of $\widehat{\langle f\rangle_\lambda}(\omega)$ can be made arbitrarily small for all ω with $\|\omega\|$ greater than a small value δ. Since δ is inversely proportional to λ, high-frequency components are removed when λ is large.

5. Numerical applicability. The definition of the transformation (4) involves high-dimensional integration which cannot be computed in general (except perhaps by the Monte Carlo method which is not appropriate for our purposes because it is too expensive). So the transformation may not be applicable to arbitrary functions, at least numerically. However, this transformation is computationally feasible for a large class of nonlinear partially separable functions, and especially for typical molecular conformation and protein folding energy functions.

We state several useful properties of the transformation in the following.

First, for the sum of functions

$$f = \sum_i f_i, \tag{65}$$

the transformation of f is equal to the sum of the transformations of the f_i's:

$$\langle f\rangle_\lambda = \sum_i \langle f_i\rangle_\lambda. \tag{66}$$

Second, for the product of functions

$$g = \prod_i g_i, \tag{67}$$

where the g_i's do not share common variables, the transformation of g is equal to the product of the transformations of the g_i's:

$$\langle g\rangle_\lambda = \prod_i \langle g_i\rangle_\lambda. \tag{68}$$

Finally, for a large subclass of nonlinear partially separable functions, called the generalized multilinear functions,

$$f = \sum_i \prod_j g_j^i, \tag{69}$$

where the g_j^i's are one-dimensional nonlinear functions, we have

$$\langle f \rangle_\lambda = \sum_i \prod_j \langle g_j^i \rangle_\lambda. \tag{70}$$

Since transformation $\langle g_j^i \rangle_\lambda$ for all i and j involves only one-dimensional integration, the transformation for a generalized multilinear function can be computed using a standard quadrature rule.

In particular, let us consider a typical n-atom molecular conformation energy function,

$$f(x) = \sum_{i=1, j>i}^{n} h_{ij}(\|x_i - x_j\|), \tag{71}$$

where $x = \{x_i \in \mathbf{R}^3,\ i = 1, \ldots, n\}$ and h_{ij} is the pairwise energy function determined by $\|x_i - x_j\|$, the distance between atoms i and j. By (66), the transformation of this energy function is equal to the sum of the transformations of the pairwise energy functions. However, the computation for the pairwise transformation still cannot be carried out directly because there is still more than one variable in each term. Nevertheless, the following theorem provides a feasible way to compute the molecular energy transformation.

THEOREM 5. *Let f be defined as in* (71). *Then the transformation of f can be computed using the formula*

$$\langle f \rangle_\lambda(x) = \sum_{i=1, j>i}^{n} \langle h_{ij} \rangle_{\sqrt{2}\lambda}(\|r_{ij}\|), \tag{72}$$

where $r_{ij} = x_i - x_j$ and

$$\langle h_{ij} \rangle_{\sqrt{2}\lambda}(\|r_{ij}\|) = c_{\sqrt{2}\lambda} \int h_{ij}(\|r_{ij} - r'_{ij}\|) e^{-\|r'_{ij}\|^2/2\lambda^2} dr'_{ij}. \tag{73}$$

Proof. We show the case when $x_i \in \mathbf{R}\ \forall i$. The general case can be proved similarly.

By the definition of $\langle f \rangle_\lambda$, in form (5), for any x,

$$\begin{aligned} &\langle f \rangle_\lambda(x) \\ &= \sum_{i=1, j>i}^{n} \langle h_{ij} \rangle_\lambda(\|x_i - x_j\|) \\ &= \sum_{i=1, j>i}^{n} c_\lambda^2 \int h_{ij}(\|(x_i - x'_i) - (x_j - x'_j)\|) e^{-\frac{\|x'_i\|^2 + \|x'_j\|^2}{\lambda^2}} dx'_i dx'_j, \end{aligned} \tag{74}$$

where c_λ is such that

$$c_\lambda \int e^{-\|x_i\|^2/\lambda^2} dx_i = 1 \qquad \forall i. \tag{75}$$

Note that for $x_i \in R$, $c_\lambda = 1/(\lambda\sqrt{\pi})$. Make the following variable transformation:

$$r_{ij} = x_i - x_j, \qquad \bar{r}_{ij} = x_i + x_j. \tag{76}$$

Then for $x_i \in R$, $dx_i dx_j = \frac{1}{2} dr_{ij} d\bar{r}_{ij}$. Now it is easy to verify that

$$\begin{aligned}
\langle f\rangle_\lambda(x) &= \sum_{i=1,j>i}^{n} c_\lambda^2 \int h_{ij}(\|(x_i - x_i') - (x_j - x_j')\|) e^{-\frac{\|x_i'\|^2 + \|x_j'\|^2}{\lambda^2}} dx_i' dx_j' \\
&= \sum_{i=1,j>i}^{n} \frac{c_\lambda^2}{2} \int h_{ij}(\|r_{ij} - r_{ij}'\|) e^{-\frac{\|r_{ij}'\|^2 + \|\bar{r}_{ij}'\|^2}{2\lambda^2}} dr_{ij}' d\bar{r}_{ij}' \\
&= \sum_{i=1,j>i}^{n} c_{\sqrt{2}\lambda}^2 \int e^{-\frac{\|\bar{r}_{ij}'\|^2}{2\lambda^2}} d\bar{r}_{ij}' \int h_{ij}(\|r_{ij} - r_{ij}'\|) e^{-\frac{\|r_{ij}'\|^2}{2\lambda^2}} dr_{ij}' \\
&= \sum_{i=1,j>i}^{n} c_{\sqrt{2}\lambda} \int h_{ij}(\|r_{ij} - r_{ij}'\|) e^{-\frac{\|r_{ij}'\|^2}{2\lambda^2}} dr_{ij}' \\
&= \sum_{i=1,j>i}^{n} \langle h_{ij}\rangle_{\sqrt{2}\lambda}(\|r_{ij}\|). \qquad \square
\end{aligned} \tag{77}$$

The integral for $\langle h_{ij}\rangle_{\sqrt{2}\lambda}(\|r_{ij}\|)$ involves only variable r_{ij} and can be computed with a standard numerical integration technique; therefore, the transformation $\langle f\rangle_\lambda(x)$ can be computed in this fashion.

Note that the integral for $\langle f\rangle_\lambda(x)$ must exist, for otherwise the transformation will have no definition. In practice, if the integral for a given f does not exist, a modified function may need to be considered instead. For example, the energy function given in (3) cannot be integrated directly because the function goes to infinity as r_{ij} goes to zero. Usually, this can be cured by replacing the function for small r_{ij} with finite interpolation (see [4, 12, 19]).

Note also that the result in Theorem 5 applies only to energy functions that can be formulated in form (71). Most potential energy terms for proteins can be expressed as pairwise forms, e.g, the Lennard–Jones potential, the electrostatic potential, the interaction potential for bonded atoms, etc. [2, 17]. However, protein energy functions do contain terms that are not pairwise distance functions, e.g., the torsional potential usually is given as a function of dihedral angles. To transform this type of function, special approximation techniques may need to be used. We will not address this issue in this work.

6. Anisotropic vs. isotropic. The transformation we have discussed so far is of the isotropic type in the sense that it averages function variations equally along all directions in the variable space. In practice, we might want to average different sizes of function variations along different directions, i.e., use different λ values for different variables, to obtain a more accurate overall structure of the function. For this purpose, we can define a more general transformation, called the anisotropic transformation.

DEFINITION 2. *Given a nonlinear function f, the anisotropic transformation $\langle f\rangle_\Lambda$ for f is defined such that for all x,*

$$\langle f\rangle_\Lambda(x) = C_\Lambda \int f(x')\, e^{-\|\frac{x-x'}{\Lambda}\|^2}\, dx' \tag{78}$$

or, equivalently,

$$\langle f\rangle_\Lambda(x) = C_\Lambda \int f(x-x')\, e^{-\|\frac{x'}{\Lambda}\|^2}\, dx', \tag{79}$$

where Λ is a diagonal matrix with positive diagonal elements:

$$\Lambda = \mathbf{diag}([\lambda_1, \lambda_2 \ldots]), \tag{80}$$

and $C_\Lambda = \prod_i c_{\lambda_i}$ with c_{λ_i} determined such that

$$c_{\lambda_i} \int e^{-\|x_i\|^2/\lambda_i^2}\, dx_i = 1 \qquad \forall i. \tag{81}$$

Note also that in this definition,

$$\|x/\Lambda\|^2 = \sum_{i=1}^{n} \|x_i\|^2/\lambda_i^2. \tag{82}$$

From this definition, we see that the anisotropic transformation will be reduced to the isotropic transformation when the diagonal elements of Λ are all identical.

Many of the important properties of the isotropic transformation carry over to the anisotropic case. We state these properties in the following.

First, for the sum of the functions

$$f = \sum_i f_i, \tag{83}$$

we have

$$\langle f\rangle_\Lambda = \sum_i \langle f_i\rangle_\Lambda. \tag{84}$$

Second, for the product of the functions

$$g = \prod_i g_i, \tag{85}$$

where the g_i's do not share common variables, we have

$$\langle g\rangle_\Lambda = \prod_i \langle g_i\rangle_{\Lambda_i}, \tag{86}$$

where Λ_i's are small diagonal matrices. If g_i is a function of j variables $x_1^i, \ldots, x_j^i$, $\Lambda_i = \mathbf{diag}([\lambda_1^i, \ldots, \lambda_j^i])$ for j positive numbers $\lambda_1^i, \ldots, \lambda_j^i$.

Third, for the generalized multilinear functions

$$f = \sum_i \prod_j g_j^i, \tag{87}$$

where the g_j^i's are one-dimensional nonlinear functions, we have

$$\langle f\rangle_\Lambda = \sum_i \prod_j \langle g_j^i\rangle_{\lambda_j}. \tag{88}$$

We can also derive a simple formula to compute the anisotropic transformation for molecular conformation energy functions.

THEOREM 6. *Let f be defined as in* (71). *Then the anisotropic transformation of f can be computed using the formula*

$$\langle f\rangle_\Lambda(x) = \sum_{i=1,j>i}^{n} \langle h_{ij}\rangle_{\sqrt{\lambda_i^2+\lambda_j^2}}(\|r_{ij}\|), \tag{89}$$

where $r_{ij} = x_i - x_j$ and

$$\langle h_{ij}\rangle_{\sqrt{\lambda_i^2+\lambda_j^2}}(\|r_{ij}\|) = c_{\sqrt{\lambda_i^2+\lambda_j^2}} \int h_{ij}(\|r_{ij} - r'_{ij}\|) e^{-\frac{\|r'_{ij}\|^2}{\lambda_i^2+\lambda_j^2}} dr'_{ij}. \tag{90}$$

Proof. By the definition of $\langle f\rangle_\Lambda$, in form (79), for any x,

$$\begin{aligned}
&\langle f\rangle_\Lambda(x)\\
&= \sum_{i=1,j>i}^{n} \langle h_{ij}\rangle_\Lambda(\|x_i - x_j\|)\\
&= \sum_{i=1,j>i}^{n} c_{\lambda_i} c_{\lambda_j} \int h_{ij}(\|(x_i - x'_i) - (x_j - x'_j)\|) e^{-\left(\frac{\|x'_i\|^2}{\lambda_i^2} + \frac{\|x'_j\|^2}{\lambda_j^2}\right)} dx'_i dx'_j, \qquad (91)
\end{aligned}$$

where c_{λ_i} is such that

$$c_{\lambda_i} \int e^{-\|x_i\|^2/\lambda_i^2} dx_i = 1 \qquad \forall i. \tag{92}$$

Make the following variable transformation:

$$r_{ij} = x_i - x_j, \qquad \bar{r}_{ij} = \lambda_j^2 x_i + \lambda_i^2 x_j. \tag{93}$$

Then we have

$$\frac{x_i}{\lambda_i} = \frac{1}{\lambda_i^2+\lambda_j^2}\left(\frac{\bar{r}_{ij}}{\lambda_i} + \lambda_i r_{ij}\right), \tag{94}$$

$$\frac{x_j}{\lambda_j} = \frac{1}{\lambda_i^2+\lambda_j^2}\left(\frac{\bar{r}_{ij}}{\lambda_j} - \lambda_j r_{ij}\right). \tag{95}$$

Using these relations, we can verify that

$$\begin{aligned}
&\langle f\rangle_\Lambda(x)\\
&= \sum_{i=1,j>i}^{n} c_{\lambda_i} c_{\lambda_j} \int h_{ij}(\|(x_i - x'_i) - (x_j - x'_j)\|) e^{-\left(\frac{\|x'_i\|^2}{\lambda_i^2} + \frac{\|x'_j\|^2}{\lambda_j^2}\right)} dx'_i dx'_j\\
&= \sum_{i=1,j>i}^{n} c_{\sqrt{\lambda_i^2+\lambda_j^2}} \int h_{ij}(\|r_{ij} - r'_{ij}\|) e^{-\frac{\|r'_{ij}\|^2}{\lambda_i^2+\lambda_j^2}} dr'_{ij}\\
&= \sum_{i=1,j>i}^{n} \langle h_{ij}\rangle_{\sqrt{\lambda_i^2+\lambda_j^2}}(\|r_{ij}\|), \qquad (96)
\end{aligned}$$

which completes the proof. □

The anisotropic transform determines for any initial solution a unique solution function $x(\lambda)$ for the transformed functions, and therefore can also be used as a continuation process for optimization, more general and powerful than the isotropic transform. We state these results in Theorems 7 and 8. The details for the proof are quite similar to those for Theorems 2 and 3, so we will not present them.

Parallel to Assumptions 1, 2, and 3 for Theorems 2 and 3, we make the following assumptions.

ASSUMPTION 4. *The objective function f is twice continuously differentiable, and transformation* (78) *is well defined for the function as well as its derivatives.*

ASSUMPTION 5. *Let g be the gradient of f and Ψ an operator,*

$$\Psi = \left(\frac{\partial^2}{\partial x_1^2}, \frac{\partial^2}{\partial x_2^2}, \dots \right). \tag{97}$$

Then the operation Ψ can be applied to g, and Ψg is a matrix with

$$(\Psi g)_{ij} = \Psi_j g_i = \frac{\partial^2 g_i}{\partial x_j^2}. \tag{98}$$

Transformation (78) *is well defined for all derivatives involved in Ψg. Also, $\Psi g(x)$ is uniformly bounded and satisfies a Lipschitz condition:*

$$\|\Psi g(x_1) - \Psi g(x_2)\| \leq L \, \|x_1 - x_2\| \qquad \forall x_1, \, x_2. \tag{99}$$

ASSUMPTION 6. *The transformation $\langle \nabla^2 f \rangle_\Lambda(x)$ satisfies a Lipschitz condition:*

$$\|\langle \nabla^2 f \rangle_\Lambda(x_1) - \langle \nabla^2 f \rangle_\Lambda(x_2)\| \leq L \, \|x_1 - x_2\| \qquad \forall x_1, \, x_2, \tag{100}$$

and its inverse is uniformly bounded.

Let S be a vector space, and for a positive vector $\lambda_0 < \infty$, let $\Gamma = \prod_i [0, \lambda_{0i}]$. Then we define function $h(\lambda, x)$ such that $\forall (\lambda, x) \in \Gamma \times S$,

$$h(\lambda, x) = \langle f \rangle_\Lambda(x), \tag{101}$$

where λ is the diagonal vector of Λ, i.e.,

$$\Lambda = \mathbf{diag}(\lambda). \tag{102}$$

THEOREM 7. *Let f be a given function and h be defined as in* (101). *Then under Assumptions* 4 *and* 5, $h''_{x\lambda}(\lambda, x)$ *exists and is uniformly bounded for all $\lambda \in \Gamma$ and $x \in S$, and also satisfies a Lipschitz condition in x:*

$$\|h''_{x\lambda}(\lambda, x_1) - h''_{x\lambda}(\lambda, x_2)\| \leq L \, \|x_1 - x_2\| \qquad \forall x_1, x_2 \in S. \tag{103}$$

In addition,

$$h''_{x\lambda}(\lambda, x)^T = \frac{\Lambda}{2} \langle \Psi g \rangle_\Lambda^T(x). \tag{104}$$

THEOREM 8. *Let f be a function for which Assumptions* 4, 5, *and* 6 *all hold. Then for any stationary point x_0 of $\langle f \rangle_{\Lambda_0}$, $\Lambda_0 = \mathbf{diag}(\lambda_0)$, there is a continuous and differentiable function $x(\lambda)$, $\lambda \in \Gamma$, such that $x_0 = x(\lambda_0)$ and $\forall \lambda \in \Gamma$, $x(\lambda)$ is a*

stationary point of $\langle f\rangle_\Lambda$. *The function* $x(\lambda)$ *is also the unique solution of the initial value problem*

$$x'^{\,T} = -\frac{\Lambda}{2}[\langle\nabla^2 f\rangle_\Lambda^{-1}(x)\langle\Psi g\rangle_\Lambda(x)]^T, \tag{105}$$

$$x_0 = x(\lambda_0). \tag{106}$$

7. Concluding remarks. In this paper, we have discussed a generalization of the effective energy transformation scheme used in [4, 5, 19, 20] for global energy minimization applied to molecular conformation. Instead of applying the transformation to the probability distribution, here we transform the functions directly, generalizing the scheme in [4, 5, 19, 20] to a broader class of functions. A mathematical theory for the transformation as a special continuation approach to global optimization is established. We have established that the proposed method transforms a given nonlinear objective function into a class of gradually deformed, but "smoother" or "easier" functions. A continuation procedure can then be applied to these "smoother" or "easier" functions to trace their solutions back to the original function. Two types of transformation are defined: isotropic and anisotropic. We have demonstrated that both transformation types can be applied to a large subclass of nonlinear partially separable functions and, in particular, the energy functions for molecular conformation. Methods to compute the transformation for these functions are given.

We believe that the proposed method provides a powerful and effective tool for global or robust local optimization. We can see this partially from the work in [4, 5], which can be viewed as a special application of the method. In [4, 5], the transformation method, combined with simulated annealing, was applied to the global energy minimization problem for molecular conformation. Promising results were observed even if only simple algorithms and approximated transformation were implemented.

More numerical work will be done in our future research. We will implement a group of algorithms based on the theory presented in this paper. While the transformation can now be computed with provided formulas, tracing the solution curve can be carried out using advanced numerical methods. There are at least three choices for the implementation of the tracing procedures.

1. Use a general random search procedure to trace the changes of the global solution when the transformed function is gradually changed back to the original function.
2. Apply only local optimization procedures to each transformed function to trace a set of solution curves and choose the best among all solutions obtained.
3. Solve the initial value problems for a set of solution curves and choose the best solution.

The first method is similar to the approach in [4, 5] where a simulated annealing procedure was applied to the transformed functions. This method converges to the global solution with certain probability, but a large number of random trials usually are required to obtain the convergence. The second approach is the most simple and efficient method, but the solution curves to be traced must be selected cleverly, for otherwise the global solution will not be guaranteed. The third method provides a more accurate and reliable way to trace the solution curves. As we have shown in this paper, the curves are solutions to well-defined initial value problems. So standard numerical IVP-methods can be used, e.g., predictor–corrector methods [1].

The implementation of all three tracing procedures and the numerical comparison among them will be of great interest for the further development of the algorithms.

We are especially interested in applying these methods to the global energy minimization problems for molecular conformation, especially protein folding. A set of test problems will be considered, including the Lennard–Jones microcluster conformation problem, the distance geometry problem, and several protein conformation problems.

While searching for native structures of protein molecules is certainly very important, the proposed methods can also provide information about the paths that solutions follow. Such information may contain insights about how protein molecules change from arbitrary configurations to their native structures.

Acknowledgments. The author would like to thank Lizhi Liao, Michael Todd, Lloyd Trefethen, and Wei Yuan for constructive suggestions. He would especially like to thank Thomas Coleman for many discussions relating to this work and for his helpful comments and suggestions on the manuscript, and David Shalloway for many discussions on the protein folding problem as well as the original effective energy transformation ideas.

Finally, the author appreciates the associate editor Robert Schnabel and the anonymous referee for their careful reading of the paper and for their very helpful suggestions.

REFERENCES

[1] E. L. Allgower and K. Georg (1990), *Numerical Continuation Methods*, Springer-Verlag, New York.

[2] C. L. Brooks III, M. Karplus, and B. M. Pettitt (1988), *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics*, John Wiley, New York.

[3] R. H. Byrd, E. Eskow, R. B. Schnabel, and S. L. Smith (1991), *Parallel Global Optimization: Numerical Methods, Dynamic Scheduling Methods, and Application to Molecular Configuration*, Tech. report CU-CS-553-91, Department of Computer Science, University of Colorado at Boulder, Boulder, CO.

[4] T. F. Coleman, D. Shalloway, and Z. Wu (1993), *Isotropic effective energy simulated annealing searches for low energy molecular cluster states*, Comput. Optim. Appl., 2, pp. 145–170.

[5] ——— (1994), *A parallel build-up algorithm for global energy minimizations of molecular clusters using effective energy simulated annealing*, J. Global Optim., 4, pp. 171–185.

[6] G. M. Crippen and T. F. Havel (1988), *Distance Geometry and Molecular Conformation*, John Wiley, New York.

[7] J. E. Dennis, Jr. and R. B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ.

[8] P. E. Gill, W. Murray, and M. H. Wright (1981), *Practical Optimization*, Academic Press, London, U.K.

[9] B. A. Hendrickson (1991), *The Molecular Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, NY.

[10] D. Kincaid and W. Cheney (1991), *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA.

[11] J. Kostrowicki and L. Piela (1991), *Diffusion equation method of global minimization: Performance for standard test functions*, J. Optim. Theory Appl., 69, pp. 269–284.

[12] J. Kostrowicki, L. Piela, B. J. Cherayil, and H. A. Scheraga (1991), *Performance of the diffusion equation method in searches for optimum structures of clusters of Lennard–Jones atoms*, J. Physical Chem., 95, pp. 4113–4119.

[13] Z. Li and H. A. Scheraga (1987), *Monte Carlo approach to the multiple-minima problem in protein folding*, in Proc. National Academy of Science, 84, pp. 15–29.

[14] J. A. Northby (1987), *Structure and binding of Lennard–Jones clusters:* $13 \leq n \leq 147$, J. Chem. Physics, 87, pp. 6166–6178.

[15] L. PIELA, J. KOSTROWICKI, AND H. A. SCHERAGA (1989), *The multiple-minima problem in the conformational analysis of molecules. Deformation of the protein energy hypersurface by the diffusion equation method*, J. Physical Chem., 93, pp. 3339–3346.

[16] E. O. PURISIMA AND H. A. SCHERAGA (1987), *An approach to the multiple-minima problem in protein folding by relaxing dimensionality*, J. Molecular Biology, 196, pp. 697–709.

[17] H. A. SCHERAGA (1993), *ECEPP/3 Empirical Conformation Energy Program for Peptides*, Department of Chemistry, Cornell University, Ithaca, NY.

[18] T. SCHLICK (1987), *Modeling and Minimization Techniques for Predicting Three-Dimensional Structures of Large Biological Molecules*, Ph.D. thesis, Courant Insititute of Mathematical Sciences, New York University, New York.

[19] D. SHALLOWAY (1992), *Packet annealing: A deterministic method for global minimization. Application to molecular conformation*, in Global Optimization, C. Floudas and P. Pardalos, eds., Princeton University Press, Princeton, NJ, pp. 433–477.

[20] ——— (1992), *Application of the renormalization group to deterministic global minimization of molecular conformation energy functions*, Global Optim., 2, pp. 281–311.

[21] L. T. WILLE (1987), *Minimum-energy configuration of atomic clusters: New results obtained by simulated annealing*, Chem. Physics Lett., 133, pp. 405–410.

[22] G.-L. XUE (1992), *Improvement on the Northby Algorithm for Molecular Conformation: Better Solutions*, Army High Performance Computing Research Center, University of Minnesota, Minneapolis, MN.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 769–806, August 1996
© 1996 Society for Industrial and Applied Mathematics
014

VISCOSITY SOLUTIONS OF MINIMIZATION PROBLEMS *

HEDY ATTOUCH†

Abstract. Viscosity methods for minimization problems are revisited from some modern perspectives in variational analysis. Variational convergences for sequences of functions (epi-convergence, Γ-convergence, Mosco-convergence) and for sequences of operators (graph-convergence) provide a flexible tool for such questions. It is proved, in a rather large setting, that the solutions of the approximate problems converge to a "viscosity solution" of the original problem, that is, a solution that is minimal among all the solutions with respect to some viscosity criteria. Various examples coming from mathematical programming, calculus of variations, semicoercive elliptic equations, phase transition theory, Hamilton–Jacobi equations, singular perturbations, and optimal control theory are considered.

Key words. minimization problems, viscosity methods, Tikhonov regularization, elliptic regularization, ill-posed and well-posed problems, singular perturbations, variational convergences, epi-convergence, Γ-convergence, Moreau–Yosida approximation, convex duality, mathematical programming, hierarchical minimization, calculus of variations, phase transitions, semicoercive elliptic equations, optimal control theory, Hamilton–Jacobi equations

AMS subject classifications. 65K05, 65K10, 49J40, 49L25, 58E30, 47H05

Introduction and notation. Viscosity methods provide an efficient approach to a large number of problems coming from different branches of mathematical analysis, such as mathematical programming (for example, Tikhonov regularization and exponential penalty methods for linear programming), variational problems (for example, minimal hypersurfaces, plasticity theory, and phase transition), partial differential equations (for example, entropy solutions of first-order nonlinear hyperbolic systems, viscosity solutions of Hamilton–Jacobi equations, degenerate elliptic and parabolic equations, and semicoercive elliptic equations), control theory, and ill-posed problems. A major feature of these methods is to provide, as a limit of the solutions of the approximate problems, a particular (possibly relaxed or generalized) solution of the original problem, called a *viscosity solution*, which has remarkable properties.

In this paper, we focus our attention on viscosity methods for *global minimization* problems. In a quite general setting, relying on topological and rescaling arguments, we prove that the viscosity solution minimizes, among all the solutions of the original problem, some viscosity criteria. In so doing, we are able to explain, in a unified way, a number of viscosity results; some of them are described in §1.

The first abstract formulations of the properties of the viscosity approximation were given by Tikhonov [58] in 1963 while studying ill-posed problems (for further details, one can consult the monograph of Dontchev and Zolezzi [25, Chap. 1] and its bibliographical notes). Although in a general setting the approximate problems fail to be numerically well conditioned, this method behaves nicely in mathematical linear programming and more generally when considering sharp minimization problems, in which case it is an exact method, as proved by Mangasarian [45], Polyak [49], and Lemaire [38]. Recently, Cominetti and San Martin [21] introduced an exponential penalty method in linear programming that combines advantages of both the exterior and interior penalty methods and can be viewed as a dual viscosity method. The

* Received by the editors December 9, 1993; accepted for publication (in revised form) April 6, 1995.

† Laboratoire d'Analyse Convexe, Département de Mathématiques, Université Montpellier II, 34095 Montpellier Cedex 5, France (attouch@math.univ-montp2.fr).

solutions of the penalized approximate problems converge to a "centroid" of the optimal face.

The viscosity method, also called elliptic regularization, has been successfully applied to various problems originating from calculus of variations. For minimal surface problems one can consult Ekeland and Temam [26, Chap. 5, §3.4] and Giusti [29], and for plasticity theory Temam [56]. The Van der Waals–Cahn–Hilliard theory for phase transitions introduces an additional viscosity term in the classical Gibbs formulation. The asymptotic analysis (when the viscosity coefficient goes to zero) was first proved by Modica [46] in 1987, with the help of the epi-convergence method (Γ-convergence), the viscosity solution being proven to enjoy desirable geometric properties (minimal interface area). Note that, in this situation, one must deal with a sequence of nonconvex variational problems.

The viscosity method also plays a central role in the study of degenerate elliptic and parabolic second-order equations; see Kohn and Nirenberg [34], Lions [41], and Ladyzenskaya and Uralceva [36]. The concept of a viscosity solution for Hamilton–Jacobi equations, which plays a crucial role in control theory, game theory, and partial differential equations, was introduced by Crandall and Lions [22] in 1983. Various applications of the viscosity method can be found in optimal control theory, singular perturbations, and minimal cost problems due, respectively, to Lions [42, 43], Lemaire [37], and Attouch [2, 3] and in stochastic control theory due to Fleming and McEneaney [27].

Let us now make precise the mathematical abstract setting. Given $f\colon X \to \mathbb{R} \cup \{+\infty\}$, an extended real-valued function whose definition may include some constraints, let us consider the minimization problem

$$(\mathcal{P}) \qquad \min\{f(x) : x \in X\}.$$

Given $g\colon X \to \mathbb{R}^+ \cup \{+\infty\}$, called the viscosity function, and $\varepsilon > 0$, a small parameter intended to go to zero, we consider the approximate minimization problem

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) : x \in X\}.$$

The viscosity function g usually enjoys nice properties like strict convexity, continuity, and coerciveness with respect to the norm of a reflexive Banach space that make the approximate minimization problem $(\mathcal{P}_\varepsilon)$ well posed (existence, uniqueness, and stability). So, it is assumed that, for all $\varepsilon > 0$, there exists a solution u_ε of $(\mathcal{P}_\varepsilon)$.

The central question is to study the convergence of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ and the characterization of its limit. To that end, we use *epi-convergence* (also called Γ-*convergence*), a notion of convergence for sequences of functions specially designed to pass to the limit on the corresponding minimization problems. Let us recall the definition of epi-convergence (which for simplicity we state in the context of metrizable topological spaces) and its basic variational properties; for a detailed presentation one can consult Attouch [2], Aubin and Frankowska [10], Dal Maso [23], De Giorgi [24], and Rockafellar and Wets [53]. Let (X, τ) be a topological space and $\{f, f_n\colon X \to \mathbb{R} \cup \{+\infty\}; n \in \mathbb{N}\}$ a sequence of extended real-valued functions. The sequence $\{f_n; n \in \mathbb{N}\}$ *epi-converges* to f and we write $\tau\text{-epi-lim}_{n\to\infty} f_n = f$ if for each $x \in X$ the two following properties (E_1) and (E_2) hold:

$$\begin{cases} (E_1) & \text{There exists } \langle x_n \rangle \ \tau\text{-convergent to } x \text{ for which } f(x) = \lim_{n\to\infty} f_n(x_n). \\ (E_2) & \text{Whenever } \langle x_n \rangle \text{ is } \tau\text{-convergent to } x, \text{then } f(x) \le \liminf_{n\to\infty} f_n(x_n). \end{cases}$$

The terminology comes from the equivalence between the epi-convergence of a sequence $\{f_n; n \in \mathbb{N}\}$ to f and the topological set-convergence of the corresponding sequence of epigraphs $\{\text{epi } f_n; n \in \mathbb{N}\}$ to epi f in $X \times \mathbb{R}$ equipped with the product topology. Without any ambiguity on the topology τ, we omit the prefix τ and write $f = \text{epi-}\lim_{n\to\infty} f_n$.

Let us now explain the epi-convergence approach to the limit on a sequence of minimization problems

$$(\mathcal{P}_n) \qquad \min\{f_n(x) : x \in X\}.$$

As in the direct method in calculus of variations, one first establishes uniform estimates of the solutions u_n of the problems $(\mathcal{P}_n)$, which naturally provides a topology τ making the sequence $\{u_n; n \in \mathbb{N}\}$ relatively compact. So, let us assume that there exists a topology τ on X such that the following topological properties (i) and (ii) are satisfied.

(i) For every $n \in \mathbb{N}$, there exists a solution u_n to $(\mathcal{P}_n)$ and the sequence $\{u_n; n \in \mathbb{N}\}$ is τ-relatively compact.

(ii) $f = \tau\text{-epi-}\lim_{n\to\infty} f_n$.

Then $\lim_{n\to\infty}(\inf_X f_n) = \inf_X f$, and every τ-limit point $\hat{u}$ of the sequence $\{u_n; n \in \mathbb{N}\}$ minimizes f on X.

In general, epi-convergence is not comparable with pointwise convergence. Nevertheless, there is an important situation where the two notions coincide (up to some closure operations). It is the case of monotone sequences (see [2, §2.5] for further details):

$$\text{If } f_1 \le f_2 \le \cdots \le f_n \le \cdots, \text{then } \tau\text{-epi-}\lim_{n\to\infty} f_n = \sup_{n\in\mathbb{N}} cl_\tau f_n.$$

$$\text{If } f_1 \ge f_2 \ge \cdots \ge f_n \ge \cdots, \text{then } \tau\text{-epi-}\lim_{n\to\infty} f_n = \text{cl}_\tau(\inf_{n\in\mathbb{N}} f_n).$$

In the above formulas, $\text{cl}_\tau f$ is the *lower closure* (lower semicontinuous regularization) with respect to the topology τ of f. In some sense, this explains the success of the monotone approximation schemes and the viscosity methods in variational analysis and optimization.

Let us now return to the viscosity approximation method and consider the generalized sequence of functions $f_\varepsilon = f + \varepsilon g$. Let us assume that there exists a topology τ on the space X such that the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ of the corresponding minimizers is τ-relatively compact. In addition, let us suppose that

$$\text{cl}_\tau(f + \delta_{\text{dom } g}) = \text{cl}_\tau f,$$

where δ_C is the *indicator* function of the set C (equal to zero on C and $+\infty$ elsewhere) and $\text{dom } g = \{x \in X: g(x) < +\infty\}$ is the *domain* of g. This means that all the required information for the minimization of f can be recovered just by considering the restriction of f to $\text{dom } g$. It follows easily from the above epi-convergence argument that every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ minimizes $\text{cl}_\tau f$ on the space X and that the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is a minimizing sequence of the original problem $(\mathcal{P})$.

Moreover, and this is the major point, the solution $\hat{u}$ obtained by this method satisfies a *viscosity selection principle* (Theorem 3.3); that is,

$$\text{(VSP)} \qquad \begin{cases} \hat{g}(\hat{u}) \le \hat{g}(v) \quad \text{for every } v \in \text{argmin } \text{cl}_\tau f, \\ \hat{u} \in \text{argmin } \text{cl}_\tau f. \end{cases}$$

The key of the proof of the above viscosity selection principle is a judicious *rescaling* of the approximate problems $(\mathcal{P}_\varepsilon)$. To that end, we introduce some positive real number $\theta(\varepsilon)$ such that

$$\limsup_{\varepsilon\to 0} \frac{1}{\theta(\varepsilon)} \left[\inf_X (f+\varepsilon g) - \inf_X f\right] < +\infty.$$

The viscosity criteria $\hat{g}$ are obtained as an epi-limit, more precisely,

$$\hat{g} = \text{epi-}\lim_{\varepsilon\to 0} \left[\frac{1}{\theta(\varepsilon)} (f - \inf_X f) + \frac{\varepsilon}{\theta(\varepsilon)} g\right].$$

When no singular perturbation occurs, then $\theta(\varepsilon)$ is of order ε and the viscosity criteria $\hat{g}$ can be taken equal to g. This quite elementary situation is examined in §2, where the equality $\hat{g} = g$ is proved (Theorems 2.1 and 2.4), when assuming that g is a finitely valued τ-continuous function. Otherwise, the viscosity criteria $\hat{g}$ may be different from g, as in the phase transition problem, where $\theta\varepsilon) = C\sqrt{\varepsilon}$. Indeed, this rescaling epi-convergence method goes far beyond the viscosity approximation problem. It can be developed in the general topological setting of the epi-convergence; see §3. As a by-product of this method, we obtain an asymptotic expansion, with respect to the parameter ε, of the infimal value of the approximating problems (Theorem 3.3). That is, there exists a function $r(\varepsilon)$ with $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ such that

$$\inf_X (f+\varepsilon g) = \inf_X f + \theta(\varepsilon) \inf\{\hat{g}(x) : x \in \operatorname{argmin} \operatorname{cl}_\tau f\} + \theta(\varepsilon) r(\varepsilon).$$

This is a marginal analysis result. For related results in sensitivity analysis of optimization problems, one may consult Auslender and Cominetti [11], Bonnans [14], Kall [33], Penot [47], Robinson [50], Rockafellar [51, 52], and Seeger [54].

We stress the fact that the above approach provides only a general framework for the analysis of viscosity methods for minimization problems. The computation of $\theta(\varepsilon)$ and the existence of the epi-limit in the definition of $\hat{g}$ are open questions in such an abstract setting. These questions must be solved in each particular situation and may be quite involved, as shown by the phase transition example of §1.3. Theorem 3.5, which states a recent result obtained by Torralba [57], provides a first general answer to these questions. Moreover, this general approach naturally suggests new conjectures for optimization and variational problems involving small parameters (see the examples of §§1.4 and 1.5).

In §4, we show that the method can be iterated and developed as an algorithm in order to provide higher-order information and asymptotic developments. Of particular interest in sensitivity, statistics, and control theory is the situation

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) + \varepsilon^2 h(x) : x \in X\}.$$

In this case, the iterated viscosity selection principle gives rise (Theorem 4.4) to the so-called hierarchical minimization principle.

In §5, we consider *convex* minimization problems, in which case the convergence results can be refined. In Proposition 5.3 we pay particular attention to the differential and geometrical properties of the viscosity trajectories. This is the key for obtaining a dynamical asymptotical viscosity selection principle (Attouch and Cominetti [5]), which is stated in Theorem 5.4.

In §6, we illustrate the viscosity selection on some convex minimization problems. We first consider semicoercive elliptic equations, a class of problems where nonuniqueness is a central issue, and for which the viscosity method selects a particular solution.

We finally describe, thanks to the preceding viscosity results, a new approach to epigraphical regularization (generalized Moreau–Yosida approximation) and obtain some differentiability properties of these approximations with respect to the parameter ε, making the link (Proposition 6.1) with Hamilton–Jacobi equations.

1. Examples. The examples described in this section are mostly intended as illustrations of various situations related to the viscosity method. A complete treatment of each of them is out of the scope of this paper. In §6, we present two other examples (that is, semicoercive elliptic equations and Hamilton–Jacobi-type equations) that will be studied in detail.

1.1. Tikhonov regularization. X is a normal linear space and $\|x\|$ is the norm of an element x of X. Consider the minimization problem

$$(\mathcal{P}) \qquad \min\{f(x)\colon\ x \in X\},$$

which is assumed to be ill posed (that is, lacks existence, or uniqueness, or stability of a solution). Following the standard Tikhonov regularization [58, 59], for every $\varepsilon > 0$, we introduce the approximate problem

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon\|x\|^2\colon\ x \in X\},$$

which is assumed to be well posed, and denote by u_ε the unique solution of $(\mathcal{P}_\varepsilon)$. For example, this is the case if X is a Hilbert space and f is a convex lower semicontinuous proper function. A common feature of this approximation is that

$$u_\varepsilon \to \hat{u}, \quad \text{where } \begin{cases} \hat{u} \in \operatorname{argmin} f, \\ \|\hat{u}\|^2 \le \|v\|^2 \quad \forall\, v \in \operatorname{argmin} f; \end{cases}$$

that is, the sequence u_ε converges to $\hat{u}$, the element of minimal norm of argmin f (when this element is uniquely defined). Moreover, u_ε is a minimizing sequence of the initial problem $(\mathcal{P})$.

As an illustration, let us briefly describe the Tikhonov regularization of ill-posed linear equations. Let X and Y be two Hilbert spaces and $A\colon\ X \to Y$ a linear bounded operator. Given $y \in Y$, the equation $Ax = y$ may be not well posed and the least-squares method consists of solving the convex quadratic minimization problem

$$(\mathcal{P}) \qquad \min\{\|Ax - y\|^2\colon\ x \in X\}.$$

The Euler equation (optimality condition) of this convex minimization problem is $A^*A(x) = A^*y$, whose solution exists if and only if the classical compatibility condition holds; that is, in this situation, $y \in A(X) + A(X)^\perp$. The Tikhonov regularization was introduced to overcome the lack of stability of this equation. For any $\varepsilon > 0$, the approximate problem

$$(\mathcal{P}_\varepsilon) \qquad \min\{\|Ax - y\|^2 + \varepsilon\|x\|^2\colon\ x \in X\}$$

is well posed and has a unique solution u_ε. The sequence $\{u_\varepsilon; \varepsilon \to 0\}$ remains bounded in X iff problem $(\mathcal{P})$ has a solution, in which case this sequence norm converges in X to $\hat{u}$, the unique element of minimal norm of the solution set of $(\mathcal{P})$:

$$\|\hat{u}\|^2 \le \|v\|^2 \quad \forall\, v \in \operatorname{argmin}(\mathcal{P}).$$

A rigorous proof of these statements is given in §5 (Corollary 5.2). For more on this approximation, one can consult Dontchev and Zolezzi [25, Chap. I, §7].

Besides the above selection property of a particular solution, this method has received particular attention because, in a number of practical interesting situations, it is an exact method. If problem $(\mathcal{P})$ has a "sharp" minimum, for example in linear programming, it is not necessary to let ε go to zero. For ε small enough, u_ε is a solution of $(\mathcal{P})$. For more on this property, consult Mangasarian [45], Polyak [49], and Lemaire [38, 39].

1.2. Exponential penalty method in mathematical programming (see Cominetti and San Martin [21]). Let $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ and consider the linear programming problem

$$(\mathcal{P}) \qquad \min\{cx\colon\ Ax \leq b\}.$$

We look for a solution $x \in \mathbb{R}^n$ of the minimization problem $(\mathcal{P})$. The (nonlinear) approximation of $(\mathcal{P})$ by the exponential penalty method is the unconstrained minimization problem

$$(\mathcal{P}_\varepsilon) \qquad \min\left\{cx + \varepsilon \sum_{i=1}^{m} \exp\left[\frac{1}{\varepsilon}\,(A_i x - b_i)\right]\right\}.$$

As ε decreases to zero, the penalization term monotonically increases to $+\infty$ outside of the constraint and monotonically decreases to zero on the constraint set (see Figure 1). This method combines the nice features of the exterior penalty method and the interior barrier methods. The connection between the exponential penalty method and the viscosity method is made clear when considering the dual formulation of the approximated minimization problem

$$(\mathcal{P}_\varepsilon)^* \qquad \min_{\substack{A^t\lambda = -c \\ \lambda \geq 0}} \left\{b^t\lambda + \varepsilon \sum_{i=1}^{m} \lambda_i(\ln \lambda_i - 1)\right\}.$$

Taking X equal to $\mathbb{R}^{m^+}$ and for g the strictly convex continuous function defined by

$$g(\lambda) = \sum_{i=1}^{m} \lambda_i(\ln \lambda_i - 1),$$

the viscosity selection principle does apply. One can prove that the solutions λ_ε of the dual problems $(\mathcal{P}_\varepsilon)^*$ converge to the solution of the initial dual problem $(\mathcal{P})^*$, which is minimal with respect to the viscosity criteria g. More precisely, $\lambda_\varepsilon \to \lambda^*$, where λ^* is the unique solution of the strictly convex minimization problem (where I_0 is the set of active constraints on the optimal solution set of $(\mathcal{P})$)

$$\min\left\{\sum \lambda_i(\ln \lambda_i - 1)\colon\ A^t\lambda = -c; \lambda_i \geq 0, i \in I_0; \lambda_i = 0, i \notin I_0\right\}.$$

More surprisingly, one can prove (see [21, 20]) that the solutions u_ε of $(\mathcal{P}_\varepsilon)$ also converge as

$$u_\varepsilon \to \hat{u} \text{ "centroid" of the optimal face.}$$

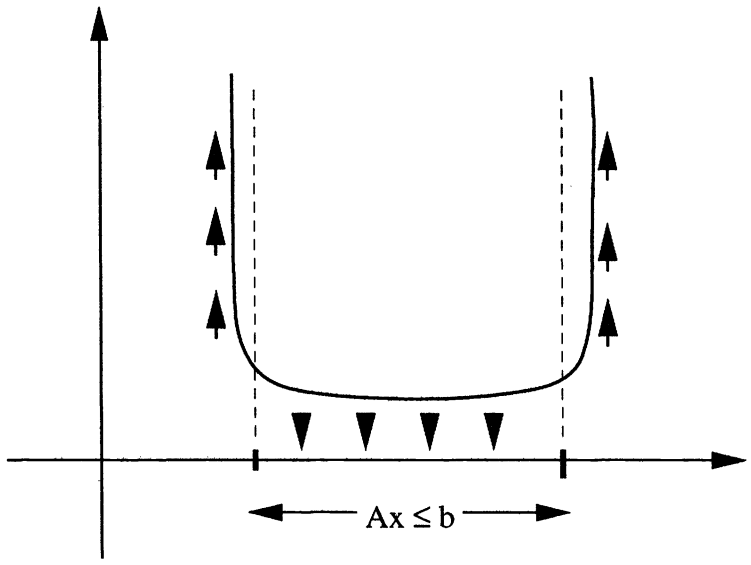


FIG. 1

1.3. Phase transition. The classical Gibbs model for the equilibrium of a fluid with two phases of respective density α and β leads to the minimization problem

$$(\mathcal{P}) \qquad \min\left\{\int_\Omega W(u(x))\,dx:\ \int_\Omega u(x)\,dx = m\right\},$$

where $u(x)$ is the density of the fluid at the point x of Ω, m is the total mass of the fluid, and the internal energy of the fluid W reaches its minimum precisely for the values α and β, as shown by Figure 2. This minimization problem $(\mathcal{P})$ is not convex; it has an infinite number of solutions u that are discontinuous functions taking only the values α and β and such that

$$\alpha \text{ meas } \{u = \alpha\} + \beta \text{ meas } \{u = \beta\} = m.$$

This model is unsatisfying because it does not provide any information about the geometry of the interface between the two phases $\{u = \alpha\}$ and $\{u = \beta\}$. The Van der Waals–Cahn–Hilliard theory introduces an additional viscosity term in the variational formulation:

$$(\mathcal{P}_\varepsilon) \qquad \min\left\{\int_\Omega W(u(x))\,dx + \varepsilon\int_\Omega |Du(x)|^2\,dx:\ \int_\Omega u(x)\,dx = m\right\}.$$

The problem $(\mathcal{P}_\varepsilon)$ has a solution u_ε belonging to $H^1(\Omega)$ and the sequence (u_ε) is relatively compact in $L^1(\Omega)$. The following convergence result was first proved in [46] by Modica in 1987 with the help of the epi-convergence method (Γ-convergence); see also Kohn and Sternberg [35], Bouchitté [15], Anzellotti and Baldo [1], and Torralba [57]. The sequence u_ε is relatively compact in $L^1(\Omega)$ and, if $\hat{u}$ is a limit point of this sequence, satisfies

$$\begin{cases} \hat{u} \in \text{argmin } (\mathcal{P}) = \left\{v \text{ takes only the values } \alpha \text{ and } \beta, \int_\Omega v(x)\,dx = m\right\}, \\ \hat{g}(\hat{u}) \le \hat{g}(v) \quad \text{for all } v \in \text{argmin } (\mathcal{P}), \end{cases}$$

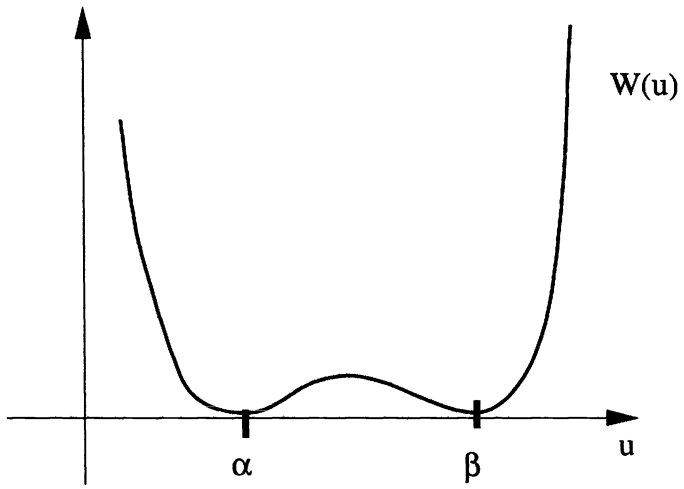


FIG. 2

where

$$\hat{g}(v) := \int_\Omega |Dv(x)|.$$

Note that for v belonging to argmin $(\mathcal{P})$, $\hat{g}(v)$ is equal to the perimeter of the phase $\{v = \alpha\}$. Hence, among all solutions v taking only the values α and β and whose integral is equal to m, $\hat{u}$ has a minimal interface area (perimeter). This is a (relaxed) viscosity selection result.

1.4. Minimal surfaces and Dirichlet problem. Let Ω be an open bounded subset of $\mathbb{R}^n$. The nonparametric minimal surface problem with prescribed Dirichlet boundary data, "$u = \varphi$ on $\partial\Omega$," can be formulated as

$$(\mathcal{P}) \qquad \min\left\{\int_\Omega \sqrt{1+|Du(x)|^2}\,dx : u = \varphi \text{ on } \partial\Omega\right\}$$

(see Aubin and Ekeland [9] and Ekeland and Temam [26]). In general, there is no solution of problem $(\mathcal{P})$ in the Sobolev space $W^{1,1}(\Omega)$ because the minimizing sequences for problem $(\mathcal{P})$ are bounded only in $W^{1,1}(\Omega)$, which is a nonreflexive Banach space. So, the direct method of the calculus of variations leads naturally to a generalized solution in $\mathrm{BV}(\Omega)$, the space of functions whose first derivatives are bounded measures on Ω. The above analysis and the corresponding relaxation formulation of $(\mathcal{P})$ can be attacked with the help of the viscosity approximation

$$(\mathcal{P}_\varepsilon) \qquad \min\left\{\int_\Omega \sqrt{1+|Du(x)|^2}\,dx + \varepsilon\int_\Omega |Du(x)|^2\,dx : u = \varphi \text{ on } \partial\Omega\right\}.$$

For each fixed $\varepsilon > 0$, problem $(\mathcal{P}_\varepsilon)$ admits a unique solution u_ε in $H^1(\Omega)$ and the sequence (u_ε) is bounded in $W^{1,1}(\Omega)$ and, hence, relatively compact in $L^1(\Omega)$. Letting $u_{\varepsilon_\nu} \to \hat{u}$ in $L^1(\Omega)$, it can be proved (see [26, 29]) that $\hat{u}$ is a solution of the relaxed problem

$$(\mathcal{P}_{\mathrm{rel}}) \qquad \min\left\{\int_\Omega \sqrt{1+|D_a u(x)|^2}\,dx + \int_\Omega |D_s u| + \int_{\partial\Omega} |u-\varphi|\,d\sigma:\ u \in \mathrm{BV}(\Omega)\right\}.$$

In the above formula, $D_a u$ is the density of the absolutely continuous part of Du with respect to the Lebesgue measure on $\mathbb{R}^n$ and $D_s u$ is the singular part of Du.

Note that the Dirichlet boundary condition "$u = \varphi$ on $\partial\Omega$" has been relaxed into the integral form $\int_{\partial\Omega} |u-\varphi|\, d\sigma$. In general, for a nonconvex open set Ω, the solution of $(\mathcal{P}_{\text{rel}})$ is uniquely defined up to an additive constant (see [9, 26, 29]). The convergence of the whole sequence (u_ε) to a precise solution (viscosity solution) of $(\mathcal{P}_{\text{rel}})$ is an open question; one may *conjecture* that the whole sequence (u_ε) converges to some function $\hat{u}$ that minimizes, among all solutions, some relaxed viscosity criteria. The functional

$$\hat{g}(v) := \int_{\partial\Omega} |v - \varphi|\, d\sigma$$

looks like a natural candidate as a relaxed viscosity criteria. This type of conjecture, naturally suggested by the general viscosity selection principle, is even more interesting when considering the viscosity approximation of variational problems from plasticity theory, since in that case the nonuniqueness of the solution of the relaxed problem is even more evident.

1.5. Optimal control and singular perturbations. Let us describe an example of a singular perturbation problem in optimal control theory. We consider the state equation

$$\begin{cases} -\varepsilon \Delta y_\varepsilon(v) + \beta(y_\varepsilon(v)) = v & \text{on } \Omega, \\ y_\varepsilon(v) = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\beta\colon \mathbb{R} \to \mathbb{R}$ is a monotone continuous function. The control is v, given as a real-valued function on the open set Ω, and the state $y_\varepsilon(v)$ is the corresponding solution of the above equation (note that, for $\varepsilon > 0$ fixed and v in the space $L^2(\Omega)$, this is a well-posed equation). Then take as a criterion (cost functional) the functional J_ε given by the formula

$$J_\varepsilon(v) := \int_\Omega |y_\varepsilon(v) - z_d|^p\, dx + N \int_\Omega v^2\, dx.$$

The optimal control problem can be formulated as

$$\min\{J_\varepsilon(v) : v \in U_{\text{ad}}\}$$

where U_{ad}, the set of admissible controls, is a closed convex subset of $L^2(\Omega)$.

The natural question that arises in this kind of situation is the convergence, as ε goes to zero, of the corresponding sequence of minimization problems, as well as the existence of an asymptotic expansion, with respect to ε, of the solution v_ε of the above problem. This may be an involved problem, especially when there are constraints on the control v. This type of question was intensively studied around 1983; see Lions [42, 43], Haraux and Murat [31, 32], Buttazzo and Dal Maso [18, 19], and Attouch [2, Ex. 2.51].

Let us return to the above problem and take for U_{ad} the whole space $L^2(\Omega)$. By using the state equation, the minimization of J_ε on $L^2(\Omega)$ is equivalent to the minimization of the functional

$$F_\varepsilon(y) := \int_\Omega \left(|y - z_d|^p + N(\beta(y) - \varepsilon\Delta y)^2\right) dx.$$

By integration by parts,

$$-\int_\Omega \beta(y)\,\Delta y = \int_\Omega \beta'(y)\,|Dy|^2,$$

so we can rewrite the functional F_ε as

$$\begin{aligned} F_\varepsilon(y) &= \int_\Omega (|y - z_d|^p + N\beta(y)^2) - 2\varepsilon N\beta(y)\Delta y + N\varepsilon^2(\Delta y)^2 \\ &= \int_\Omega (|y - z_d|^p + N\beta(y)^2) + \varepsilon \int_\Omega 2N\beta'(y)\,|Dy|^2 + \varepsilon^2 \int_\Omega N(\Delta y)^2 \\ &= F_0(y) + \varepsilon F_1(y) + \varepsilon^2 F_2(y) \end{aligned}$$

with $F_1 \geq 0, F_2 \geq 0$ (viscosity functionals). This type of situation will be examined in §4. It gives rise to a so-called hierarchical viscosity selection principle.

2. The case $\hat{g} = g$. Let us first examine the case where the viscosity function g is finitely valued. In this quite elementary situation (no singular perturbation occurs) one can give a simple direct proof of the viscosity selection principle. To prepare the reader for the general case (in the next paragraph), we give a proof relying on an epi-convergence argument and which will serve as a guideline in the general case.

THEOREM 2.1. *Let $f\colon X \to \mathbb{R} \cup \{+\infty\}$ be an extended real-valued function and*

$$(\mathcal{P}) \qquad \min\{f(x) : x \in X\}$$

be the associated minimization problem. Let us assume that f is proper and bounded from below. Let $g\colon X \to \mathbb{R}^+$ be a nonnegative real-valued function, and for any $\varepsilon > 0$ let us consider the approximate minimization problem

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) : x \in X\}.$$

Let us assume that there exists a topology τ on X such that the following topological properties are satisfied.

(i) *For every $\varepsilon > 0$, there exists a solution u_ε to $(\mathcal{P}_\varepsilon)$ and the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is τ-relatively compact.*

(ii) *f and g are τ-lower semicontinuous.*

Then every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ minimizes the function f on X and satisfies the viscosity selection principle

$$(\text{VSP}) \qquad \begin{cases} g(\hat{u}) \leq g(v) \quad \text{for every } v \in \operatorname{argmin} f, \\ \hat{u} \in \operatorname{argmin} f. \end{cases}$$

In other words, $\hat{u}$ minimizes g over the nonempty set argmin f. *The sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is a minimizing sequence of the original problem $(\mathcal{P})$, more precisely*

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\,[f(u_\varepsilon) - \inf_X f] = 0$$

and

$$\lim_{\varepsilon \to 0} g(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} f\}.$$

Proof. a) Let us denote $f_\varepsilon := f + \varepsilon g$. Since g is a nonnegative finite-valued function, the sequence $\{f_\varepsilon; \varepsilon \to 0\}$ is pointwise convergent and *monotonically decreasing* to f (as ε decreases to zero). From the general properties of the epi-convergence, it is τ-epiconvergent to $\mathrm{cl}_\tau f$, the τ-lower semicontinuous regularization of f; see [2, Thm. 2.46]. Since, by assumption (ii), f is τ-lower semicontinuous, we have $\mathrm{cl}_\tau f = f$, and τ-epi-lim $f_\varepsilon = f$. By assumption (i), the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ satisfies $u_\varepsilon \in \operatorname{argmin} f_\varepsilon$ and is τ-relatively compact, which, from the variational properties of the epi-convergence [2, Thm. 2.11] implies that every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ minimizes f on X. Let us denote $\hat{u} = \lim_\nu u_{\varepsilon(\nu)}$. As a consequence, the set argmin f is nonvoid.

b) To prove that $\hat{u}$ satisfies the viscosity selection principle (VSP), we rescale the minimization problem $(\mathcal{P}_\varepsilon)$. In so doing, we obtain an enlarged and sharper view of what is happening around the minimal value and the solution set. The basic and quite elementary idea is that the set of solutions of a minimization problem is not affected by adding some constant and/or multiplying by a positive constant the function that is to be minimized. If we subtract the real number $\inf_X f$ from f_ε and then multiply the resulting function by ε^{-1}, we obtain that, for each $\varepsilon > 0$, u_ε is the solution of the minimization problem (rescaled problem)

$$(\mathcal{RP}_\varepsilon) \qquad \min\left\{\frac{1}{\varepsilon}\,[f(x) - \inf_X f] + g(x) : x \in X\right\}.$$

Let us denote

$$h_\varepsilon := \frac{1}{\varepsilon}\,(f - \inf_X f) + g.$$

Since $f - \inf_X f$ is a nonnegative function, the sequence $\{h_\varepsilon; \varepsilon \to 0\}$ is *monotonically increasing* to $h = g + \delta_S$, where δ_S is the indicator function of the nonempty solution set $S = \operatorname{argmin} f$. Using once more the epi-convergence property of monotone sequences of functions, but now for increasing sequences (see [2, Thm. 2.40]), and noticing that h_ε is τ-lower semicontinuous, we obtain

$$\tau\text{-epi-lim } h_\varepsilon = g + \delta_S.$$

As in the preceding argument, we conclude, thanks to the variational properties of the epi-convergence, that $\hat{u}$ minimizes $g + \delta_S$ on X; that is, $\hat{u}$ minimizes g on S.

c) Let us make precise the convergence of the optimal values. Since u_ε minimizes h_ε on X, for all v in X we have

$$\frac{1}{\varepsilon}\,(f(u_\varepsilon) - \inf_X f) + g(u_\varepsilon) \le \frac{1}{\varepsilon}\,(f(v) - \inf_X f) + g(v). \tag{2.1}$$

Hence, for each v belonging to argmin f,

$$g(u_\varepsilon) \le g(v).$$

It follows that

$$\limsup_{\varepsilon \to 0} g(u_\varepsilon) \le g(\hat{u}) = \min\{g(v) : v \in \operatorname{argmin} f\}. \tag{2.2}$$

On the other hand, from the τ-lower semicontinuity of g

$$g(\hat{u}) \le \liminf_\nu g(u_{\varepsilon(\nu)}) \le \limsup_\nu g(u_{\varepsilon(\nu)}) \le \limsup_{\varepsilon \to 0} g(u_\varepsilon). \tag{2.3}$$

Combining (2.2) and (2.3), we obtain

$$\lim_{\nu} g(u_{\varepsilon(\nu)}) = \min\{g(v) : v \in \operatorname{argmin} f\}.$$

This being valid for any extracted subsequence, the whole sequence $\{g(u_\varepsilon); \varepsilon \to 0\}$ converges:

$$\lim_{\varepsilon \to 0} g(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} f\}. \tag{2.4}$$

Let us now return to (2.1); by using (2.4) and by taking $v = \hat{u}$, we obtain

$$\limsup_{\varepsilon \to 0} \frac{1}{\varepsilon} (f(u_\varepsilon) - \inf_X f) + g(\hat{u}) \leq g(\hat{u}),$$

that is,

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon} (f(u_\varepsilon) - \inf_X f) = 0, \tag{2.5}$$

which completes the proof of Theorem 2.1. □

We are led to introduce the following definition.

DEFINITION 2.2. *We say that a solution $\hat{u}$ of the minimization problem*

$$\min\{f(x) : x \in X\} \tag{$\mathcal{P}$}$$

is a viscosity solution corresponding to the viscosity function g: $X \to \mathbb{R}^+ \cup \{+\infty\}$ if $\hat{u}$ minimizes g over argmin f.

We can reformulate Theorem 2.1 by saying that any limit point of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$, where u_ε is a solution of the approximate problem $(\mathcal{P}_\varepsilon)$, is a viscosity solution of $(\mathcal{P})$. Note that the notion of viscosity solution depends on the choice of the viscosity function g. However, it is not affected when replacing g by its composition $\alpha o g$ with a strictly increasing real-valued function α.

Remark 2.3. The "rescaled problem"

$$\min\left\{\frac{1}{\varepsilon}[f(x) - \inf_X f] + g(x) : x \in X\right\} \tag{$\mathcal{RP}_\varepsilon$}$$

can be viewed as the penalized approximation of the minimization problem

$$\min\{g(x) : x \in \operatorname{argmin} f\}.$$

In this case, $P_f(\cdot) = f(\cdot) - \inf_X f$ is a penalty function for the constraint set argmin f. As a consequence, the conclusions of Theorem 2.1 can be derived from general penalty results; see Luenberger [44]. The close connection between viscosity methods and penalty methods was already observed by Bensoussan and Kenneth [13] and Lemaire [37]. But this penalty argument will not permit us to derive the general viscosity selection principle (Theorem 3.3). It is by an adaptation of the epi-convergence and scaling method that we shall obtain this result.

When the function f to be minimized is not τ-lower semicontinuous, some relaxation phenomena may occur, as shown by the following theorem, which is a direct adaptation of Theorem 2.1 and is stated in a complete form for the convenience of the reader.

THEOREM 2.4 (viscosity and relaxation). *With the same notation as in Theorem* 2.1, *let*

$$(\mathcal{P}) \qquad \min\{f(x) : x \in X\}$$

be a given minimization problem, and for any $\varepsilon > 0$ *let us consider the approximate minimization problem with a nonnegative function* g:

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) : x \in X\}.$$

Let us assume that there exists a topology τ *on* X *such that the following conditions are satisfied.*

(i) *For every* $\varepsilon > 0$, *there exists a solution* u_ε *to* $(\mathcal{P}_\varepsilon)$ *and the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *is* τ*-relatively compact.*

(ii) g *is* τ*-continuous.*

Then every τ*-limit point* $\hat{u}$ *of the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *minimizes* $\mathrm{cl}_\tau f$ *on* X *and satisfies*

$$\text{(VSP)} \qquad \begin{cases} g(\hat{u}) \le g(v) & \text{for every } v \in \operatorname{argmin} \mathrm{cl}_\tau f, \\ \hat{u} \in \operatorname{argmin} \mathrm{cl}_\tau f. \end{cases}$$

Moreover, the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ *is a minimizing sequence of problem* $(\mathcal{P})$. *More precisely,*

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon} [f(u_\varepsilon) - \inf_X f] = 0$$

and

$$\lim_{\varepsilon \to 0} g(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} \mathrm{cl}_\tau f\}.$$

Proof of Theorem 2.4. As already mentioned, Theorem 2.4 can be obtained by a direct adaptation of the proof of Theorem 2.1. To that end, let us notice that a given function h and its τ-lower closure, $\mathrm{cl}_\tau h$, have the same infimum. Moreover, if u minimizes h, then it minimizes $\mathrm{cl}_\tau h$, that is, $\operatorname{argmin} h \subset \operatorname{argmin} \mathrm{cl}_\tau h$. Using the same notation as in the proof of Theorem 2.1, we have that u_ε minimizes over X the function $f_\varepsilon = f + \varepsilon g$. By the preceding argument, u_ε minimizes over X the function $\mathrm{cl}_\tau f_\varepsilon$, which by assumption (ii) on the τ-continuity of g is equal to $\mathrm{cl}_\tau f + \varepsilon g$. Similarly, in the proof of the viscosity selection principle, u_ε minimizes over X the function $\mathrm{cl}_\tau h_\varepsilon$, which again by the τ-continuity assumption on g is equal to $\frac{1}{\varepsilon}(\mathrm{cl}_\tau f - \inf_X f) + g$. The epi-convergence arguments of sections a) and b) of the proof of Theorem 2.1, as well as part c), can now be developed in a similar way, just replacing f by $\mathrm{cl}_\tau f$. □

THEOREM 2.5 (asymptotic development). *Under the assumptions of Theorem* 2.1, *the optimal values of the approximate problems satisfy the following asymptotic development. There exists some real-valued function* $\varepsilon \to r(\varepsilon)$ *satisfying* $\lim_{\varepsilon \to 0} r(\varepsilon) = 0$ *such that*

$$\inf_X (f + \varepsilon g) = \inf_X f + \varepsilon[\inf g(x) : x \in \operatorname{argmin} f] + \varepsilon r(\varepsilon).$$

Similarly, under the assumptions of Theorem 2.4,

$$\inf_X (f + \varepsilon g) = \inf_X f + \varepsilon[\inf g(x) : x \in \operatorname{argmin} \mathrm{cl}_\tau f] + \varepsilon r(\varepsilon).$$

Proof. With the notation of Theorem 2.1, we have

$$\begin{aligned}\inf_X(f+\varepsilon g) &= f(u_\varepsilon)+\varepsilon g(u_\varepsilon)\\ &= \inf_X f + [f(u_\varepsilon)-\inf_X f]+\varepsilon g(u_\varepsilon)\\ &= \inf_X f + \varepsilon \inf[g(x):\ x\in \operatorname{argmin} f] + \varepsilon[g(u_\varepsilon)-\inf\{g(x)\ :\ x\in \operatorname{argmin} f\}]\\ &\quad + \varepsilon\left[\frac{1}{\varepsilon}\,(f(u_\varepsilon)-\inf_X f)\right].\end{aligned}$$

The result follows from equalities (2.4) and (2.5). In the case of Theorem 2.4, one just needs to replace f by $\mathrm{cl}_\tau f$. □

In the preceding situations, the viscosity function is everywhere defined and takes only finite values. In a large number of problems of, for example, calculus of variations and partial differential equations, the viscosity function g takes infinite values and its domain is strictly contained in the space X where we expect to find a solution. In that case, the viscosity criteria $\hat{g}$ may be different from g. We shall consider this general situation in the next section. In the following statement, we examine a case where g takes infinite values and where one can still exhibit a viscosity selection principle with $\hat{g}=g$. To do so, we need to adapt the preceding argument and assume that one can recover all the required information just from the knowledge of f on the domain of g.

THEOREM 2.6. *With the same notation as in Theorem* 2.1, *let*

$$(\mathcal{P}) \qquad \min\{f(x): x\in X\}$$

be a given minimization problem, and for any $\varepsilon>0$ *let*

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x)+\varepsilon g(x): x\in X\}$$

be the approximate minimization problem with $g\colon X\to \mathbb{R}^+\cup\{+\infty\}$. *Let us assume that there exists a topology* τ *on* X *such that the following topological properties are satisfied.*

(i) *for every* $\varepsilon>0$, *there exists a solution* u_ε *to* $(\mathcal{P}_\varepsilon)$ *and the sequence* $\{u_\varepsilon;\varepsilon\to 0\}$ *is* τ*-relatively compact.*

(ii) $\mathrm{cl}_\tau(f+\delta_{\mathrm{dom}\, g})=\mathrm{cl}_\tau f$.

Then every τ*-limit point* $\hat{u}$ *of the sequence* $\{u_\varepsilon;\varepsilon\to 0\}$ *minimizes the function* $\mathrm{cl}_\tau f$ *on* X *and*

$$\lim_{\varepsilon\to 0} f(u_\varepsilon)=\inf_X f.$$

Assuming, moreover, that the following hypotheses hold:

(iii) g *is* τ*-lower semicontinuous,*

(iv) $\mathrm{cl}_\tau[f+\varepsilon g]=\mathrm{cl}_\tau f+\varepsilon g$ *for every* $\varepsilon>0$,

(v) $\mathrm{dom}\, g\cap \operatorname{argmin}\mathrm{cl}_\tau f\neq\varnothing$,

$\hat{u}$ *satisfies*

$$(\mathrm{VSP}) \qquad \begin{cases} g(\hat{u})\le g(v) \quad \text{for every } v\in \operatorname{argmin}\mathrm{cl}_\tau f,\\ \hat{u}\in \operatorname{argmin}\mathrm{cl}_\tau f.\end{cases}$$

Moreover,

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon} [f(u_\varepsilon) - \inf_X f] = 0,$$

$$\lim_{\varepsilon \to 0} g(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} \operatorname{cl}_\tau f\}.$$

Proof. We just need to adapt the proof of Theorem 2.1, using the fact that the function g is no longer everywhere defined and may take infinite values.

a) Let us still denote $f_\varepsilon = f + \varepsilon g$. Since g is a nonnegative extended real-valued function, the sequence $\{f_\varepsilon; \varepsilon \to 0\}$ is pointwise convergent and monotonically decreasing to $f + \delta_{\operatorname{dom} g}$ (as ε decreases to zero). From the general properties of the epi-convergence, it is τ-epiconvergent to $\operatorname{cl}_\tau(f + \delta_{\operatorname{dom} g})$, which by assumption (ii) is equal to $\operatorname{cl}_\tau f$. Using that $u_\varepsilon \in \operatorname{argmin} f_\varepsilon$, τ-epi-lim $f_\varepsilon = \operatorname{cl}_\tau f$, and the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is τ-relatively compact, it follows from the variational properties of the epi-convergence that every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ minimizes $\operatorname{cl}_\tau f$ on X. Let us denote $u_{\varepsilon(\nu)} \to \hat{u}$. As a consequence, the set $\operatorname{argmin} \operatorname{cl}_\tau f$ is nonvoid.

b) To prove that $\hat{u}$ satisfies the viscosity selection principle (VSP), we rescale the minimization problem $(\mathcal{P}_\varepsilon)$ and obtain that, for each $\varepsilon > 0$, u_ε is a solution of the minimization problem (rescaled problem)

$$(\mathcal{RP}_\varepsilon) \qquad \min \left\{ \frac{1}{\varepsilon} [f(x) - \inf_X \operatorname{cl}_\tau f] + g(x) : x \in X \right\}.$$

It follows immediately from the classical properties of the lower semicontinuous regularization that u_ε minimizes over X the function

$$h_\varepsilon = \operatorname{cl}_\tau \left[\frac{1}{\varepsilon} (f - \inf_X \operatorname{cl}_\tau f) + g \right],$$

which, by assumption (iv), can be reexpressed as

$$h_\varepsilon = \frac{1}{\varepsilon} [\operatorname{cl}_\tau f - \inf_X \operatorname{cl}_\tau f] + g.$$

Since $[\operatorname{cl}_\tau f - \inf_X \operatorname{cl}_\tau f]$ is a nonnegative function, the sequence $\{h_\varepsilon; \varepsilon \to 0\}$ is monotonically increasing to $h = g + \delta_S$, where δ_S is the indicator function of the nonvoid solution set $S = \operatorname{argmin} \operatorname{cl}_\tau f$. Using once more the epi-convergence property of monotone sequences of functions, but now for increasing sequences, we obtain

$$\tau\text{-epi-lim } h_\varepsilon = g + \delta_S.$$

As in the preceding argument, we conclude, thanks to the variational properties of the epi-convergence, that $\hat{u}$ does minimize $g + \delta_S$ on X, that is, $\hat{u}$ minimizes g on S. Clearly, this minimization property brings some additional information on $\hat{u}$ only when the function $g + \delta_S$ is proper, that is, when the domain of g intersects the solution set S, which is precisely assumption (v).

c) Let us make precise the convergence of the optimal values. Since u_ε minimizes h_ε over X, for all v in X we have

$$\frac{1}{\varepsilon} [\operatorname{cl}_\tau f(u_\varepsilon) - \inf_X \operatorname{cl}_\tau f] + g(u_\varepsilon) \leq \frac{1}{\varepsilon} [\operatorname{cl}_\tau f(v) - \inf_X \operatorname{cl}_\tau f] + g(v).$$

Hence, for every v belonging to $\operatorname{argmin} \operatorname{cl}_\tau f$, we have $g(u_\varepsilon) \leq g(v)$.

It follows that

$$\limsup_{\varepsilon\to 0} g(u_\varepsilon) \leq g(\hat{u}) = \min\{g(v) : v \in \operatorname{argmin} \operatorname{cl}_\tau f\}.$$

On the other hand, from the τ-lower semicontinuity of g

$$g(\hat{u}) \leq \liminf_{\nu} g(u_{\varepsilon(\nu)}) \leq \limsup_{\varepsilon\to 0} g(u_\varepsilon).$$

It follows that

$$\lim_{\nu} g(u_{\varepsilon(\nu)}) = \min\{g(v) : v \in \operatorname{argmin} \operatorname{cl}_\tau f\}.$$

The argument being valid for any extracted subsequence, we obtain that the whole sequence $g(u_\varepsilon)$ converges and

$$\lim_{\varepsilon\to 0} g(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} \operatorname{cl}_\tau f\}.$$

In a similar way as in the proof of Theorem 2.1, we obtain

$$\limsup_{\varepsilon\to 0} \frac{1}{\varepsilon} (\operatorname{cl}_\tau f(u_\varepsilon) - \inf_X \operatorname{cl}_\tau f) + g(\hat{u}) \leq g(\hat{u}),$$

that is,

$$\lim_{\varepsilon\to 0} \frac{1}{\varepsilon} (\operatorname{cl}_\tau f(u_\varepsilon) - \inf_X \operatorname{cl}_\tau f) = 0.$$

We conclude thanks to the equality $\operatorname{cl}_\tau f(u_\varepsilon) = f(u_\varepsilon)$, which follows from the equality between the infimal value of the function $f + \varepsilon g$, that is, $f(u_\varepsilon) + \varepsilon g(u_\varepsilon)$, and the infimal value of its closure $\operatorname{cl}_\tau(f + \varepsilon g)$, that is, $\operatorname{cl}_\tau f(u_\varepsilon) + \varepsilon g(u_\varepsilon)$. Note that to obtain this last equality, we have once more applied assumption (iv). □

Examples. Assumptions (i), (ii), (iii), (iv) are not really restrictive and are satisfied in most practical situations. The crucial assumption is (v), which may not be satisfied typically when g acts as a singular perturbation on f (this will be examined in the next section). Theorems 2.1 and 2.4 can be directly applied to obtain the convergence of the viscosity method and the viscosity selection principle in the classical Tikhonov approximation (§1.1) as well as in the dual exponential penalty method (§1.2). Theorem 2.6 is well fitted to the barrier approximation methods; see [2, Ex. 2.49] and [8] for the epi-convergence analysis of the barrier methods.

3. The general case. Epi-convergence and scaling method. As already mentioned, in a large number of infinite-dimensional problems, the viscosity term acts as a *singular perturbation*, and it may happen that the domain of the viscosity function g does not even meet the solution set of the original problem, argmin f. In that case, we are going to prove that the viscosity approximation still selects a particular solution of the original problem and there is a selection viscosity principle, but for a relaxed viscosity criteria $\hat{g}$ that may be different from g. It is an appropriate scaling on the values of the approximating problems that will allow us to exhibit such a viscosity selection principle. Indeed, this method goes far beyond the viscosity approximation problem and can be developed in the general topological setting of epi-convergence. The basic idea is summarized in the following theorem, which somewhat improves (it makes precise the order of the scaling) a related result from Anzellotti and Baldo [1, Thm. 1.2]. For the convenience of the reader, in the first part of the statement, we

repeat the main lines of the classical epi-convergence approach to the limit of sequences of minimization problems.

THEOREM 3.1. *Given* $\{f, f_\varepsilon\colon X \to \mathbb{R} \cup \{+\infty\}; \varepsilon \to 0\}$ *a sequence of extended real-valued functions, let us consider the corresponding sequence of minimization problems*:

$$(\mathcal{P}_\varepsilon) \qquad \min\{f_\varepsilon(x) : x \in X\}.$$

Let us assume that there exists a topology τ *on* X *such that the following topological properties are satisfied.*

(i) *For every* $\varepsilon > 0$, *there exists a solution* u_ε *of* $(\mathcal{P}_\varepsilon)$ *and the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *is* τ*-relatively compact.*

(ii) $f = \tau\text{-epi-lim}_{\varepsilon\to 0} f_\varepsilon$.

Then $\lim_{\varepsilon\to 0}(\inf_X f_\varepsilon) = \inf_X f$ *and every* τ*-limit point* $\hat{u}$ *of the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *minimizes* f *on* X.

Let us go further and introduce some positive real number $\theta(\varepsilon)$ *that tends to zero with* ε *and such that*

(iii) $\limsup_{\varepsilon\to 0} \frac{1}{\theta(\varepsilon)}[\inf_X f_\varepsilon - \inf_X f] < +\infty$.

Let us consider the sequence of rescaled functions $h_\varepsilon(x) := \frac{1}{\theta(\varepsilon)}[f_\varepsilon(x) - \inf_X f]$ *and assume that*

(iv) $h := \tau\text{-epi-lim}_{\varepsilon\to 0} h_\varepsilon$ *exists with* $h\colon X \to \mathbb{R} \cup \{+\infty\}$.

Then h *is a proper function that is identically equal to* $+\infty$ *outside of* argmin f *and* $\hat{u}$ *minimizes* h *on the set* argmin f:

$$\begin{cases} h(\hat{u}) \le h(v) & \text{for every } v \in \operatorname{argmin} f, \\ \hat{u} \in \operatorname{argmin} f. \end{cases}$$

Moreover, there exists some function $r(\varepsilon)$ *with* $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ *such that*

$$\inf_X f_\varepsilon = \inf_X f + \theta(\varepsilon) \inf\{h(x) : x \in \operatorname{argmin} f\} + \theta(\varepsilon) r(\varepsilon).$$

Proof of Theorem 3.1. Since $\theta(\varepsilon)$ is positive, the rescaling $h_\varepsilon = \frac{1}{\theta(\varepsilon)}[f_\varepsilon - \inf_X f]$ preserves the solution set and u_ε still minimizes the function h_ε. From the variational properties of epi-convergence (recall that the sequence u_ε is τ-relatively compact), it follows that $\hat{u}$ minimizes the function $h := \tau\text{-epi-lim}_{\varepsilon\to 0} h_\varepsilon$ on the space X and that

$$\lim_{\varepsilon\to 0} \inf_X h_\varepsilon = \inf_X h.$$

Since

$$\inf_X h_\varepsilon = \frac{1}{\theta(\varepsilon)} [\inf_X f_\varepsilon - \inf_X f],$$

it follows from assumption (iii) that

$$\inf_X h < +\infty. \tag{3.1}$$

Hence, $-\infty < h(\hat{u}) = \inf_X h < +\infty$ and h is a proper function. Let us now verify that h is identically equal to $+\infty$ outside of argmin f. Let us fix some point v in X such that $v \notin \operatorname{argmin} f$; that is,

$$f(v) > \inf_X f. \tag{3.2}$$

By assumption, $h := \tau\text{-epi-lim}_{\varepsilon\to 0} h_\varepsilon$, which, by definition of epi-convergence, yields the existence of some τ-converging sequence $v_\varepsilon \to v$ such that

$$h(v) = \lim_{\varepsilon\to 0} h_\varepsilon(v_\varepsilon).$$

Equivalently,

$$h(v) = \lim_{\varepsilon\to 0} \frac{1}{\theta(\varepsilon)} [f_\varepsilon(v_\varepsilon) - \inf_X f]. \tag{3.3}$$

On the other hand, again using the assumption $f = \tau\text{-epi-lim}_{\varepsilon\to 0} f_\varepsilon$, we have

$$\liminf_{\varepsilon\to 0} f_\varepsilon(v_\varepsilon) \geq f(v),$$

which, from (3.2), implies the existence of some positive number $m > 0$, such that for ε sufficiently small,

$$f_\varepsilon(v_\varepsilon) - \inf_X f > m.$$

This implies that, in formula (3.3), $h(v) = +\infty$. Returning to (3.1) we obtain

$$\inf_X h = \inf\{h(x) : x \in \operatorname{argmin} f\} < +\infty,$$

which, combined with the convergence of $\inf_X h_\varepsilon$ to $\inf_X h$, yields

$$\lim_{\varepsilon\to 0} \frac{\inf_X f_\varepsilon - \inf_X f}{\theta(\varepsilon)} = \inf\{h(x) : x \in \operatorname{argmin} f\}.$$

Equivalently, there exists some function $r(\varepsilon)$ with $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ such that

$$\inf_X f_\varepsilon = \inf_X f + \theta(\varepsilon) \inf\{h(x) : x \in \operatorname{argmin} f\} + \theta(\varepsilon) r(\varepsilon). \qquad \square$$

Remark 3.2. a) We stress the fact that the appropriate scaling is obtained by dividing $(f_\varepsilon - \inf_X f)$ by some $\theta(\varepsilon)$, which is of order less than or equal to $\inf_X f_\varepsilon - \inf_X f$. Indeed, we do not need to know exactly the quantity $\inf_X f_\varepsilon - \inf_X f$, but what we need to know precisely is the order of this infinitesimal quantity.

b) The assumption $\theta(\varepsilon) > 0$ is not really restrictive; one can always reduce to that case when $\inf_X f_\varepsilon - \inf_X f$ is different from zero. Let us consider the viscosity approximation method when $\inf_X(f + \varepsilon g) - \inf_X f = 0$. Taking u_ε as a solution of $(\mathcal{P}_\varepsilon)$,

$$\inf_X f \leq f(u_\varepsilon) \leq f(u_\varepsilon) + \varepsilon g(u_\varepsilon) \leq \inf_X f_\varepsilon = \inf_X f;$$

that is, u_ε is a solution of $(\mathcal{P})$ and $g(u_\varepsilon) = 0$. In other words, in that case the approximation method is exact and provides a solution that still minimizes the viscosity criteria g.

This combined scaling and epi-convergence method is illustrated in Figure 3.

A straight adaptation of Theorem 3.1 to the viscosity method yields the following result.

THEOREM 3.3 (general viscosity selection principle). *Let $f\colon X \to \mathbb{R} \cup \{+\infty\}$ be an extended real-valued function that is proper and bounded from below and*

$$\min\{f(x) : x \in X\} \tag{$\mathcal{P}$}$$

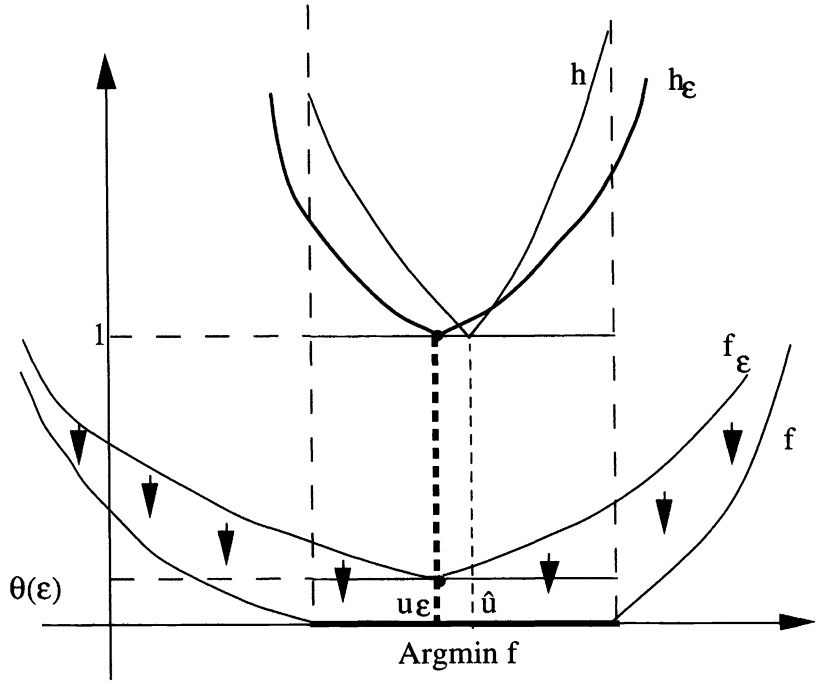


FIG. 3

be the corresponding minimization problem. Let g: $X \to \mathbb{R}^+ \cup \{+\infty\}$ *be a nonnegative extended real-valued function, and for any* $\varepsilon > 0$ *let us consider the approximate minimization problem*

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) : x \in X\}.$$

Let us assume that there exists a topology τ *on* X *such that the following topological properties are satisfied.*

(i) *For every* $\varepsilon > 0$, *there exists a solution* u_ε *of* $(\mathcal{P}_\varepsilon)$ *and the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *is* τ*-relatively compact.*

(ii) $\mathrm{cl}_\tau(f + \delta_{\mathrm{dom}\, g}) = \mathrm{cl}_\tau f$.

Then $\lim_{\varepsilon \to 0} \inf_X (f + \varepsilon g) = \inf_X f$ *and every* τ*-limit point* $\hat{u}$ *of the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *minimizes the function* $\mathrm{cl}_\tau f$ *on* X. *Let us introduce some positive real number* $\theta(\varepsilon)$ *that tends to zero with* ε *and such that*

$$\limsup_{\varepsilon \to 0} \frac{1}{\theta(\varepsilon)} \left[\inf_X (f + \varepsilon g) - \inf_X f\right] < +\infty.$$

Then consider the sequence of rescaled functions $h_\varepsilon(x) := \frac{1}{\theta(\varepsilon)}[(f + \varepsilon g)(x) - \inf_X f]$ *and assume that the following epi-limit exists:*

$$(3.4) \qquad \text{(iii)} \quad \hat{g} := \text{epi-lim}_{\varepsilon \to 0} \left[\frac{1}{\theta(\varepsilon)} (f - \inf_X f) + \frac{\varepsilon}{\theta(\varepsilon)}\, g\right].$$

Then $\hat{g}$ *is a proper function that is equal to* $+\infty$ *outside of* $\mathrm{argmin}\, \mathrm{cl}_\tau f$ *and* $\hat{u}$ *satisfies*

$$(\mathrm{VSP}) \qquad \begin{cases} \hat{g}(\hat{u}) \le \hat{g}(v) \quad \textit{for every } v \in \mathrm{argmin}\, \mathrm{cl}_\tau f, \\ \hat{u} \in \mathrm{argmin}\, \mathrm{cl}_\tau f. \end{cases}$$

Moreover, there exists some function $r(\varepsilon)$ *with* $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ *such that*

$$\inf_X(f+\varepsilon g) = \inf_X f + \theta(\varepsilon)\inf\{\hat{g}(x) : x \in \operatorname{argmin} \mathrm{cl}_\tau f\} + \theta(\varepsilon)r(\varepsilon).$$

Proof of Theorem 3.3. The first part of the proof is similar to Theorem 2.6. The second part of the proof is a straight adaptation of Theorem 3.1, taking $h = \hat{g}$. □

Example 3.4. a) In the viscosity method with a finite-valued continuous viscosity function, it results from Theorem 2.5 (this is an "a posteriori" result) that $\inf_X(f + \varepsilon g) - \inf_X f$ is an infinitesimal term of order greater than or equal to ε. Indeed, this can be a priori verified since

$$\varepsilon g(u_\varepsilon) \le \inf_X(f+\varepsilon g) - \inf_X f \le \varepsilon g(\hat{u}),$$

where u_ε is a solution of $(\mathcal{P}_\varepsilon)$ and $\hat{u}$ is an arbitrary point in argmin f. This justifies the choice $\theta(\varepsilon) = \varepsilon$ when rescaling the problem. Formula (3.4) then becomes

$$\begin{aligned}\hat{g} &= \text{epi-lim}_{\varepsilon\to 0}\left[\frac{1}{\varepsilon}(f - \inf_X f) + g\right]\\ &= g + \delta_S\end{aligned}$$

where $S = \operatorname{argmin} f$.

b) In the phase transition problem, the right scaling is obtained by dividing by $\sqrt{\varepsilon}$, which corresponds to the order of $\inf_X f_\varepsilon - \inf_X f$. Noticing that $\inf_X f = 0$, formula (3.4) in this case becomes

$$\hat{g}(u) = \text{epi-lim}_{\varepsilon\to 0}\left[\frac{1}{\sqrt{\varepsilon}}\int_\Omega W(u(x))\,dx + \sqrt{\varepsilon}\int_\Omega |Du(x)|^2\,dx\right],$$

the space $X = L^1(\Omega)$ being equipped with its norm topology and the functions u satisfying the prescribed constraint $\int_\Omega u(x)\,dx = m$. Note that, in this epi-limit process, the two terms $\frac{1}{\sqrt{\varepsilon}}\int_\Omega W(u(x))\,dx$ and $\sqrt{\varepsilon}\int_\Omega |Du(x)|^2\,dx$ cannot be treated separately, there being an interaction between them that makes the limiting process quite involved; see [46].

c) At this point, it is interesting to understand the effect of an arbitrary scaling. Let us illustrate this on the viscosity problem with a continuous finite-valued function g. Given $t \ge 0$, let us denote

$$h_{t,\varepsilon}(x) := \frac{1}{\varepsilon^t}\left[(\mathrm{cl}_\tau f(x) - \inf_X f) + \varepsilon g(x)\right]$$

and

$$h_t := \tau\text{-epi-lim}_{\varepsilon\to 0} h_{t,\varepsilon}.$$

An elementary computation yields

$$\begin{aligned}h_0 &= \mathrm{cl}_\tau f - \inf_X f,\\ h_t &= \delta_{\operatorname{argmin}\mathrm{cl}\, f} \quad \text{if } 0 < t < 1,\\ h_1 &= g + \delta_{\operatorname{argmin}\mathrm{cl}\, f}\end{aligned}$$

and, when $t > 1$, h_t is identically equal to $+\infty$ if $\inf[g(x) : x \in \operatorname{argmin} \mathrm{cl}_\tau f]$ is different from zero and equal to the indicator function of $\operatorname{argmin}[g + \delta_{\operatorname{argmin}\mathrm{cl}\, f}]$ otherwise. The

viscosity selection principle appears precisely for the value $t = 1$. Clearly the sequence of functions $\{h_t\}$ increases with t, and simultaneously the solution set argmin h_t decreases. So, it is for the largest value of the parameter t for which h_t is proper (not identically equal to $+\infty$) that we obtain the best localization of the limits $\hat{u}$ of the approximate solutions. Here it is for $t = 1$ that we obtain the viscosity selection principle. As a general rule, the most interesting viscosity selection law is obtained by taking $\theta(\varepsilon)$ of the same order as $\inf_X(f + \varepsilon g) - \inf_X f$.

Computation of $\theta(\varepsilon)$ and $\hat{g}$. Let us now discuss, from a general point of view, the two key points of Theorem 3.3, which are the determination of $\theta(\varepsilon)$ and the computation, when it exists, of

$$\hat{g} := \text{epi-lim}_{\varepsilon \to 0} \left[\frac{1}{\theta(\varepsilon)} (f - \inf_X f) + \frac{\varepsilon}{\theta(\varepsilon)} g \right].$$

As a general rule, these two questions must be treated simultaneously and it is the existence as a limit of a proper function $\hat{g}$ that tells us we have taken the right scaling $\theta(\varepsilon)$. One can distinguish two cases.

a) dom $g \cap$ argmin $\text{cl}_\tau f \neq \varnothing$, in which case one can take $\theta(\varepsilon) = \varepsilon$ and $\hat{g} = g$ by applying Theorem 2.6.

b) dom $g \cap$ argmin $\text{cl}_\tau f = \varnothing$, in which case one must take $\theta(\varepsilon) > \varepsilon$; otherwise the above epi-limit is identically $+\infty$.

As a model example of such a situation, let us take $\theta(\varepsilon) = \varepsilon^t$ with $0 < t < 1$. Then we must consider the sequence of functions $\{h_{t,\varepsilon}; \varepsilon \to 0\}$ with

$$h_{t,\varepsilon} := \frac{1}{\varepsilon^t} (f - \inf_X f) + \varepsilon^{1-t} g.$$

THEOREM 3.5 (see Torralba [57]). *Let X be a metrizable topological space. Let us consider f, g: $X \to \mathbb{R}^+ \cup \{+\infty\}$ two nonnegative extended real-valued functions that are supposed to be lower semicontinuous. For each $0 < t < 1$, let us define $h_{t,\varepsilon} := \frac{1}{\varepsilon^t}(f - \inf_X f) + \varepsilon^{1-t} g$. Then*

$$\text{epi-lim} \inf_{\varepsilon \to 0} h_{t,\varepsilon}(x) = \begin{cases} +\infty & \text{if } x \notin \text{argmin } f, \\ k_t \text{cl}_\tau(f^{1-t} g^t)(x) & \text{if } x \in \text{argmin } f, \end{cases}$$

where $k_t = \frac{1}{t}(\frac{1}{t} - 1)^{-(1-t)}$.

Note that the previous theorem does not tell us if the epi-limit exists (one also must compute the epi-lim sup). This is still an open problem in such a general setting. Indeed, it tells us, and this is an important point, what is the value of the epi-limit (that is, $\hat{g}$) when it exists.

4. Higher order epi-graphical analysis. Hierarchical minimization. The epi-convergence and scaling argument that has been developed in the preceding section can be iterated in a natural way. Indeed, in a number of practical situations, the most interesting results can be obtained by doing a second-order asymptotical analysis. Let us return to the general abstract setting where $\{f_\varepsilon\colon X \to \mathbb{R} \cup \{+\infty\}; \varepsilon \to 0\}$ is a sequence of extended real-valued functions, $(\mathcal{P}_\varepsilon)$ are the corresponding minimization problems

$$(\mathcal{P}_\varepsilon) \qquad \min\{f_\varepsilon(x) : x \in X\},$$

and u_ε is a solution of $(\mathcal{P}_\varepsilon)$. It is assumed that the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is τ-relatively compact for some topology τ on X, and we want to characterize the τ-limit points of the sequence u_ε and so let $u_{\varepsilon(\nu)} \to \hat{u}$. The epi-convergence and scaling method can be developed as an algorithm whose zero and first-order steps have been precisely considered in the preceding section and which is described below.

ALGORITHM ES (epiconvergence and scaling).

Step 0. $f = \tau\text{-epi-lim}_{\varepsilon\to 0} f_\varepsilon$.

Step 1. $\limsup_{\varepsilon\to 0} \dfrac{1}{\theta_1(\varepsilon)} [\inf_X f_\varepsilon - \inf_X f] < +\infty, \theta_1(\varepsilon) > 0$

$$h_{1,\varepsilon}(x) := \frac{1}{\theta_1(\varepsilon)} [f_\varepsilon(x) - \inf_X f].$$

$$h_1 := \tau\text{-epi-lim}_{\varepsilon\to 0} h_{1,\varepsilon}$$

Step 2. $\limsup_{\varepsilon\to 0} \dfrac{1}{\theta_2(\varepsilon)} [\inf_X h_{1,\varepsilon} - \inf_X h_1] < +\infty, \theta_2(\varepsilon) > 0$

$$h_{2,\varepsilon}(x) := \frac{1}{\theta_2(\varepsilon)} [h_{1,\varepsilon}(x) - \inf_X h_1].$$

$$h_2 := \tau\text{-epi-lim}_{\varepsilon\to 0} h_{2,\varepsilon}$$

...

Step i. $\limsup_{\varepsilon\to 0} \dfrac{1}{\theta_i(\varepsilon)} [\inf_X h_{i-1,\varepsilon} - \inf_X h_{i-1}] < +\infty, \theta_i(\varepsilon) > 0$

$$h_{i,\varepsilon}(x) := \frac{1}{\theta_i(\varepsilon)} [h_{i-1,\varepsilon}(x) - \inf_X h_{i-1}].$$

$$h_i := \tau\text{-epi-lim}_{\varepsilon\to 0} h_{i,\varepsilon}.$$

The above method can be iterated as long as $h_i := \tau\text{-epi-lim}_{\varepsilon\to 0} h_{i,\varepsilon}$ exists, and its main properties are summarized in the following statement whose proof is similar to that of Theorem 3.1.

PROPOSITION 4.1. *Let us assume that Algorithm* ES *has been developed up to Step* i. *Then for each* $j = 0, 1, 2, \ldots, i$ *the function* h_j *is a proper function that is equal to* $+\infty$ *outside of* argmin h_{j-1} *and we have*

$$\operatorname{argmin} f \supset \operatorname{argmin} h_1 \supset \operatorname{argmin} h_2 \supset \cdots \supset \operatorname{argmin} h_i \ni \hat{u}.$$

Moreover, there exists some function $r(\varepsilon)$ *with* $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ *such that*

$$\inf_X f_\varepsilon = \inf_X f + \theta_1(\varepsilon) \inf_X h_1 + \cdots + \theta_1(\varepsilon)\theta_2(\varepsilon)\ldots\theta_i(\varepsilon) \inf_X h_i + \theta_1(\varepsilon)\theta_2(\varepsilon)\ldots\theta_i(\varepsilon) r(\varepsilon).$$

As a consequence, the precision of the localization of $\hat{u}$ increases with the order of the iteration. In some sense, this method can be viewed as an epi-graphical asymptotic development with respect to ε of the sequence f_ε around the set of minimizers argmin f.

As an illustration of the above iterated scaling method, let us study the hierarchical minimization

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) + \varepsilon^2 h(x) + \cdots : x \in X\}$$

and show on this example how the above method permits us to detect the influence of the higher order terms $\varepsilon g(x), \varepsilon^2 h(x) \ldots$; on the way the approximate solutions u_ε are driven to a particular solution $\hat{u}$ of problem $(\mathcal{P})$. In this situation we shall be able, under some additional assumptions, to compute the successive iteration terms of Algorithm ES. The terminology suggests that the criteria to minimize involves different terms whose comparative importance are scaled by the corresponding coefficients $\varepsilon, \varepsilon^2, \ldots$. This type of problem naturally occurs in, for example, control theory (§1.5), statistics (see Lemaire-Misonne [40]), and sensitivity analysis. At this point, it is worth noticing that $(\mathcal{P}_\varepsilon)$ can be rewritten in the general form

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g_\varepsilon(x) : x \in X\},$$

where the sequence $\{g_\varepsilon; \varepsilon \to 0\}$ is converging, in a sense that we must make precise, to g. The viscosity selection principle as stated in Theorem 2.1 is extended in Proposition 4.3 to the above situation and will provide the zero and first-order steps of Algorithm ES. Its proof relies on the following elementary result concerning the epi-convergence of a sum of two epi-convergent sequences of functions.

LEMMA 4.2 (epi-convergence of a sum). *Let* $\{\varphi_\varepsilon\colon X \to \mathbb{R} \cup \{+\infty\}; \varepsilon \to 0\}$ *and* $\{\phi_\varepsilon\colon X \to \mathbb{R} \cup \{+\infty\}; \varepsilon \to 0\}$ *be two sequences of functions that are converging both in the pointwise and the* τ*-epiconvergence sense, respectively, to some functions* φ *and* ϕ*. Then the sequence* $\{\varphi_\varepsilon + \phi_\varepsilon; \varepsilon \to 0\}$ τ*-epiconverges to* $\varphi + \phi$ *(and pointwise converges too!).*

Proof. By the definition of epi-convergence, we just need to verify the following.

a) For every x in X, there exists some τ-converging sequence $x_\varepsilon \to x$ such that $(\varphi + \phi)(x) = \lim_{\varepsilon \to 0}(\varphi_\varepsilon + \phi_\varepsilon)(x_\varepsilon)$. Since the two sequences φ_ε and ϕ_ε are pointwise convergent, respectively, to φ and ϕ, this property is clearly satisfied with $x_\varepsilon = x$.

b) Given an arbitrary τ-converging sequence $x_\varepsilon \to x$, we have that

$$\begin{aligned} \liminf_{\varepsilon \to 0}(\varphi_\varepsilon + \phi_\varepsilon)(x_\varepsilon) &\geq \liminf_{\varepsilon \to 0} \varphi_\varepsilon(x_\varepsilon) + \liminf_{\varepsilon \to 0} \phi_\varepsilon(x_\varepsilon) \\ &\geq \varphi(x) + \phi(x), \end{aligned}$$

this last inequality being true because of the epi-convergence of the sequences φ_ε and ϕ_ε, respectively, to φ and ϕ. □

We can now state the following extension of Theorem 2.1 to the above situation, that is, when g is replaced by some converging sequence $\{g_\varepsilon; \varepsilon \to 0\}$.

PROPOSITION 4.3. *Given* $f\colon X \to \mathbb{R} \cup \{+\infty\}$ *an extended real-valued function that is assumed to be proper and bounded from below, let us consider the minimization problem*

$$(\mathcal{P}) \qquad \min\{f(x) : x \in X\}.$$

Given $\{g, g_\varepsilon\colon X \to \mathbb{R}^+; \varepsilon \to 0\}$ *a sequence of nonnegative finite-valued functions, for any* $\varepsilon > 0$*, let us consider the approximate minimization problem*

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g_\varepsilon(x) : x \in X\}.$$

Let us assume that there exists a topology τ *on* X *such that the following topological properties are satisfied.*

(i) *For every* $\varepsilon > 0$*, there exists a solution* u_ε *to* $(\mathcal{P}_\varepsilon)$ *and the sequence* $\{u_\varepsilon; \varepsilon \to 0\}$ *is* τ*-relatively compact.*

(ii) *f is τ-lower semicontinuous.*

(iii) *The sequence $\{g_\varepsilon : X \to \mathbb{R}^+; \varepsilon \to 0\}$ converges to g both in the pointwise convergence sense and in the τ-epiconvergence sense.*

Then every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ minimizes f on X and satisfies

$$\text{(VSP)} \qquad g(\hat{u}) \leq g(v) \quad \text{for every } v \in \operatorname{argmin} f.$$

Moreover, the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is a minimizing sequence of the original problem $(\mathcal{P})$:

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon} [f(u_\varepsilon) - \inf_X f] = 0,$$

$$\lim_{\varepsilon \to 0} g_\varepsilon(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} f\};$$

and there exists some real-valued function $\varepsilon \to r(\varepsilon)$ satisfying $\lim_{\varepsilon \to 0} r(\varepsilon) = 0$ such that

$$\inf_X (f + \varepsilon g_\varepsilon) = \inf_X f + \varepsilon[\inf g(v) : v \in \operatorname{argmin} f] + \varepsilon r(\varepsilon).$$

Proof of Proposition 4.3. The proof is a straight adaptation of the proof of Theorem 2.1. We just need to apply successively Lemma 4.2 to the sequences $\{f + \varepsilon g_\varepsilon; \varepsilon \to 0\}$ and $\{\frac{1}{\varepsilon}(f - \inf_X f) + g_\varepsilon; \varepsilon \to 0\}$. Note that the sequence $\{\varepsilon g_\varepsilon \colon X \to \mathbb{R}^+; \varepsilon \to 0\}$ converges both in the pointwise and the τ-epiconvergence sense to zero and that the monotonically increasing sequence $\{\frac{1}{\varepsilon}(f - \inf_X f); \varepsilon \to 0\}$ converges both in the pointwise and the τ-epiconvergence sense to δ_S, the indicator function of the nonempty solution set $S = \operatorname{argmin} f$. □

By the same argument as that which was used when passing from Theorem 2.1 to Theorem 2.4 (which formally consists of replacing f by $\operatorname{cl}_\tau f$), we can obtain a similar statement just assuming that f is proper and bounded from below and that $\{g, g_\varepsilon \colon X \to \mathbb{R}^+; \varepsilon \to 0\}$ is a sequence of nonnegative finite-valued functions, which are τ-continuous.

We can now state the main result of this section.

THEOREM 4.4 (hierarchical minimization). *Let $f_0 \colon X \to \mathbb{R} \cup \{+\infty\}$ be an extended real-valued function that is proper and bounded from below, and let*

$$(\mathcal{P}) \qquad \min\{f_0(x) : x \in X\}$$

be the corresponding minimization problem. Take $f_1, f_2 \colon X \to \mathbb{R}^+$ as two nonnegative finite-valued functions, and consider for any $\varepsilon > 0$ the approximate minimization problem

$$(\mathcal{P}_\varepsilon) \qquad \min\{f_0(x) + \varepsilon f_1(x) + \varepsilon^2 f_2(x) \colon x \in X\}.$$

Let us denote $S_0 = \operatorname{argmin} f_0, S_1 = \operatorname{argmin}\{f_1(x); x \in S_0\}, S_2 = \operatorname{argmin}\{f_2(x); x \in S_1\}$. Let us assume that there exists a topology τ on X such that the following conditions are satisfied.

(i) *For every $\varepsilon > 0$, there exists a solution u_ε to $(\mathcal{P}_\varepsilon)$ and the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ is τ-relatively compact.*

(ii) *f_0, f_1, and f_2 are τ-lower semicontinuous.*

a) *Without any additional assumption, we have only* first-order *information. Every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ belongs to $S_1, \lim_{\varepsilon \to 0} f_0(u_\varepsilon) =$*

$\inf_X f_0$, *and* $\lim_{\varepsilon\to 0} f_1(u_\varepsilon) = \inf_{S_0} f_1$. *Moreover, there exists some real-valued function* $\varepsilon \to r(\varepsilon)$ *satisfying* $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ *such that the following first-order asymptotic expansion holds*:

$$\inf_X [f_0 + \varepsilon f_1 + \varepsilon^2 f_2] = \inf_X f_0 + \varepsilon \inf_{S_0} f_1 + \varepsilon r(\varepsilon).$$

b) *To obtain* second-order *asymptotic information, we make the assumption*

$$(H_1) \qquad \liminf_{\varepsilon\to 0} \frac{1}{\varepsilon} (f_1(u_\varepsilon) - \inf_{S_0} f_1) \geq 0.$$

Then $\hat{u}$ belongs to S_2 *and we have the following second-order asymptotic expansion*:

$$\inf_X [f_0 + \varepsilon f_1 + \varepsilon^2 f_2] = \inf_X f_0 + \varepsilon \inf_{S_0} f_1 + \varepsilon^2 \inf_{S_1} f_2 + \varepsilon^2 r(\varepsilon).$$

Proof of Theorem 4.4. a) We first notice that $f_0 + \varepsilon f_1 + \varepsilon^2 f_2 = f_0 + \varepsilon(f_1 + \varepsilon f_2)$. Take $g_\varepsilon := f_1 + \varepsilon f_2$, which clearly both epi-converges and pointwise converges to f_1. The assumptions of Proposition 4.3 are satisfied. Therefore, the sets S_0 and S_1 are nonempty and every τ-limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$, letting $u_{\varepsilon(\nu)} \to \hat{u}$, belongs to the solution set S_1. Moreover,

$$\lim_{\varepsilon\to 0} f_0(u_\varepsilon) = \inf_X f_0$$

and

$$\lim_{\varepsilon\to 0} f_1(u_\varepsilon) = \inf_{S_0} f_1.$$

b) To obtain second-order information and prove that $\hat{u}$ belongs to S_2, let us apply Step 2 of Algorithm ES. With the notation of ES, we have

$$f_\varepsilon = f_0 + \varepsilon f_1 + \varepsilon^2 f_2 \quad \text{and} \quad \tau\text{-epi-lim } f_\varepsilon = f_0,$$

$$h_{1,\varepsilon} = \frac{1}{\varepsilon} (f_\varepsilon - \inf_X f_0) \quad \text{and} \quad \tau\text{-epi-lim } h_{1,\varepsilon} = h_1 = f_1 + \delta_{S_0}.$$

At the next step, taking $\theta_2(\varepsilon) = \varepsilon$ (this choice will be a posteriori justified), we introduce

$$\begin{aligned} h_{2,\varepsilon} &= \frac{1}{\varepsilon} (h_{1,\varepsilon} - \inf_X h_1) \\ &= \frac{1}{\varepsilon} (h_{1,\varepsilon} - \inf_{S_0} f_1) \\ &= \frac{1}{\varepsilon} \left[\frac{1}{\varepsilon} (f_\varepsilon - \inf_X f_0) - \inf_{S_0} f_1\right] \\ &= \frac{1}{\varepsilon^2} (f_0 - \inf_X f_0) + \frac{1}{\varepsilon} (f_1 - \inf_{S_0} f_1) + f_2. \end{aligned}$$

So, we must study the epi-convergence of the sequence $h_{2,\varepsilon}$, which we briefly denote

$$h_\varepsilon := \frac{1}{\varepsilon^2} (f_0 - \inf_X f_0) + \frac{1}{\varepsilon} (f_1 - \inf_{S_0} f_1) + f_2.$$

A major difference from the situations previously encountered is that this sequence is no longer monotonically increasing because of the term $\frac{1}{\varepsilon}(f_1 - \inf_{S_0} f_1)$, which is increasing on the set $\{f_1 \geq \inf_{S_0} f_1\}$ and decreasing on its complement. So, we must focus our attention on the behavior of the sequence h_ε on the set $S_1 = \operatorname{argmin}\{f_1(x); x \in S_0\}$. We recall that u_ε minimizes the function h_ε on the whole space X, and in particular for every v in S_1 we have

$$h_\varepsilon(u_\varepsilon) \leq h_\varepsilon(v).$$

Noticing that on the set S_1 the functions $(f_0 - \inf_X f_0)$ and $(f_1 - \inf_{S_0} f_1)$ are identically equal to zero, it follows from the definition of h_ε that for every v in S_1

$$\frac{1}{\varepsilon^2}(f_0(u_\varepsilon) - \inf_X f_0) + \frac{1}{\varepsilon}(f_1(u_\varepsilon) - \inf_{S_0} f_1) + f_2(u_\varepsilon) \leq f_2(v). \tag{4.1}$$

Since $(f_0(u_\varepsilon) - \inf_X f_0)$ is nonnegative, this implies that for every v in S_1

$$\frac{1}{\varepsilon}(f_1(u_\varepsilon) - \inf_{S_0} f_1) + f_2(u_\varepsilon) \leq f_2(v).$$

Hence,

$$\liminf_{\varepsilon \to 0}\left[\frac{1}{\varepsilon}(f_1(u_\varepsilon) - \inf_{S_0} f_1)\right] + \limsup_{\varepsilon \to 0} f_2(u_\varepsilon) \leq \inf_{S_1} f_2.$$

Using assumption (H_1), we obtain

$$\limsup_{\varepsilon \to 0} f_2(u_\varepsilon) \leq \inf_{S_1} f_2. \tag{4.2}$$

On the other hand, from the τ-lower semicontinuity of f_2,

$$f_2(\hat{u}) \leq \liminf_{\nu} f_2(u_{\varepsilon(\nu)}) \leq \limsup_{\nu} f_2(u_{\varepsilon(\nu)}) \leq \limsup_{\varepsilon \to 0} f_2(u_\varepsilon). \tag{4.3}$$

Combining (4.2), (4.3) and the fact that $\hat{u}$ belongs to the solution set S_1, we obtain

$$\lim_{\nu} f_2(u_{\varepsilon(\nu)}) = f_2(\hat{u}) = \inf_{S_1} f_2.$$

Since the argument is valid for any extracted subsequence, it follows that the whole sequence $\{f_2(u_\varepsilon); \varepsilon \to 0\}$ is convergent and

$$\lim_{\varepsilon \to 0} f_2(u_\varepsilon) = f_2(\hat{u}) = \min_{S_1} f_2, \tag{4.4}$$

that is, $\hat{u}$ belongs to S_2. We now return to (4.1) and derive that

$$\limsup_{\varepsilon \to 0} \frac{1}{\varepsilon^2}(f_0(u_\varepsilon) - \inf_X f_0) + \liminf_{\varepsilon \to 0} \frac{1}{\varepsilon}(f_1(u_\varepsilon) - \inf_{S_0} f_1) + \lim_{\varepsilon \to 0} f_2(u_\varepsilon) \leq \min_{S_1} f_2,$$

which, combined with (4.4) and assumption (H_1), yields

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon^2}(f_0(u_\varepsilon) - \inf_X f_0) = 0. \tag{4.5}$$

Let us finally return to (4.1) and derive from (4.4) and (4.5) that

$$\limsup_{\varepsilon\to 0}\frac{1}{\varepsilon}\,(f_1(u_\varepsilon)-\inf_{S_0} f_1)\le 0;$$

that is, because of assumption (H_1),

$$\lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\,(f_1(u_\varepsilon)-\inf_{S_0} f_1)=0. \tag{4.6}$$

We can now combine (4.4), (4.5), and (4.6) to obtain

$$\begin{aligned}
&\inf_X[f_0+\varepsilon f_1+\varepsilon^2 f_2]\\
&\quad=f_0(u_\varepsilon)+\varepsilon f_1(u_\varepsilon)+\varepsilon^2 f_2(u_\varepsilon)\\
&\quad=\inf_X f_0+\varepsilon\inf_{S_0} f_1+\varepsilon^2\inf_{S_1} f_2\\
&\qquad+\varepsilon^2\left[\frac{1}{\varepsilon^2}\,(f_0(u_\varepsilon)-\inf_X f_0)+\frac{1}{\varepsilon}\,(f_1(u_\varepsilon)-\inf_{S_0} f_1)+f_2(u_\varepsilon)-\min_{S_1} f_2\right]\\
&\quad=\inf_X f_0+\varepsilon\inf_{S_0} f_1+\varepsilon^2\inf_{S_1} f_2+\varepsilon^2 r(\varepsilon)
\end{aligned}$$

for some real-valued function $\varepsilon\to r(\varepsilon)$ satisfying $\lim_{\varepsilon\to 0} r(\varepsilon)=0$. □

Remark 4.5. a) It follows from the above proof that, without any assumption, the inequality

$$\limsup_{\varepsilon\to 0}\frac{1}{\varepsilon}\,(f_1(u_\varepsilon)-\inf_{S_0} f_1)\le 0$$

is always satisfied. Hence, assumption (H_1) is equivalent to

$$\lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\,(f_1(u_\varepsilon)-\inf_{S_0} f_1)=0.$$

The reason why the first-order argument cannot be directly extended to the second order, and hence to an arbitrary order $i\in\mathbb{N}$, is that we must know that

$$\liminf_{\varepsilon\to 0}\frac{1}{\varepsilon^k}\,(f_i(u_\varepsilon)-\inf_{S_{i-1}} f_i)\ge 0,\qquad k=1,2,\dots,i,$$

and, precisely, it is only for $i=0$, in which case it reduces to $f_0(u_\varepsilon)\ge\inf_X f_0$, that this inequality is automatically satisfied (we denote $S_{-1}=X$).

b) Note that assumption (H_1) is satisfied when, for ε small enough, the approximate solutions u_ε belong to the solution set S_0. When u_ε does not belong to the solution set S_0, we must examine the infinitesimal quantity $(f_1(u_\varepsilon)-\inf_{S_0} f_1)$. If, for ε small enough, it is nonnegative, then clearly assumption (H_1) is satisfied. Otherwise, we must know the order of this infinitesimal quantity, and assumption (H_1) fails to be satisfied when $f_1(u_\varepsilon)$ tends to $\inf_{S_0} f_1$ from below with $f_1(u_\varepsilon)-\inf_{S_0} f_1\le -k\varepsilon$ for some $k>0$. Indeed, one can construct quite elementary examples showing that the second-order hierarchical selection may fail without some additional assumption like (H_1).

Example 4.6. The second- (and higher-) order epi-convergence and rescaling method can be applied to various situations.

a) In the exponential penalization method in linear programming [21], one can obtain the convergence of the approximate solutions of the primal problem to a centroid of the optimal face.

b) To pass from the three-dimensional elasticity model to a two-dimensional elasticity model, one must rescale the stored energy functionals (taking as a parameter ε the thickness of the material)

$$F_\varepsilon(v) = \frac{1}{\varepsilon^\lambda} \int_{\Sigma_\varepsilon} W(x, e(v))\, dx$$

and compute their epi-limit when ε goes to zero. Here Σ is a smooth two-dimensional manifold and Σ_ε and ε-enlargement of Σ. This was achieved in [17] by Buttazzo, a striking result being that, depending on the order or the scaling, one obtains different epi-limit functionals. Assuming that W has a quadratic growth, taking $\lambda = 1$, one obtains at the limit the two-dimensional elasticity. In that case, the epi-limit functional involves only first-order tangential derivatives of v on Σ. By contrast, when $\lambda = 3$, the limit functional describes the stored energy of a shell and depends on second-order derivatives of v!

c) Let us make the connection between the epi-convergence and scaling method and obtaining optimality conditions by the way of *first- and second-order epi-derivatives*; see Rockafellar [52] and Penot [47]. Suppose that x_0 minimizes f on X. Taking $t > 0$ as a parameter, this is equivalent to saying that zero is a minimizer of $f_t(\xi) := f(x_0 + t\xi)$. Assuming that the function f is lower semicontinuous, the sequence $\{f_t; t \to 0\}$ epi-converges to the constant function $f_0 = f(x_0)$. With the notation of Algorithm ES, we are led to consider the rescaled functions

$$h_{1,t}(\xi) := \frac{1}{t}\,[f_t(\xi) - \inf_X f_0] = \frac{1}{t}\,[f(x_0 + t\xi) - f(x_0)],$$
$$h_1 := \tau\text{-epi-lim}_{t\to 0} h_{1,t},$$

which is precisely the first-order directional epi-derivative f'_{x_0} of f at x_0. Noticing that the infimum of h_1 is still equal to zero, at the next step we obtain

$$h_{2,t}(\xi) := \frac{1}{t}\,[h_{1,t}(\xi) - \inf_X h_1]$$

$$= \frac{1}{t}\left[\frac{1}{t}\,[f(x_0 + t\xi) - f(x_0)]\right]$$

$$= \frac{1}{t^2}\,[f(x_0 + t\xi) - f(x_0)],$$

$$h_2 := \tau\text{-epi-lim}_{t\to 0} h_{2,t}.$$

This is precisely the second-order epi-derivative $f''_{x_0,0}$. As a corollary, we obtain that necessary conditions for x_0 to minimize f are that f'_{x_0} and $f''_{x_0,0}$ are nonnegative.

5. Convex minimization and the viscosity method. In this section, unless otherwise specified, X is a reflexive Banach space and $\|x\|$ is the norm of an element x of X. We denote by X^* the topological dual of X and by $\langle \cdot, \cdot \rangle$ the duality pairing

between X and X^*. The space X is said to be Kadec if, for any sequence in X, the weak convergence and the convergence of the norms imply the strong convergence. We now recall some standard notions from convex analysis. Given $f\colon X \to \mathbb{R} \cup \{+\infty\}$ a convex lower semicontinuous proper function, its *Fenchel conjugate* $f^*\colon X^* \to \mathbb{R} \cup \{+\infty\}$ is given by the formula

$$f^*(y) = \sup\{\langle x, y\rangle - f(x)\colon x \in X\}.$$

The *subdifferential* of f at a point $x_0 \in \operatorname{dom} f$ is the possibly empty set

$$\partial f(x) = \{y \in X^*\colon f(x) \geq f(x_0) + \langle x - x_0, y\rangle \text{ for each } x \in X\}.$$

A function $g\colon X \to \mathbb{R}$ is said to be *coercive* if $\lim_{\|x\|\to+\infty} g(x) = +\infty$.

THEOREM 5.1. *Let us consider the minimization problem*

$$(\mathcal{P}) \qquad \min\{f(x) : x \in X\},$$

where X is a reflexive Banach space and $f\colon X \to \mathbb{R} \cup \{+\infty\}$ is a convex lower semicontinuous proper function that is bounded from below. Let $g\colon X \to \mathbb{R}^+$ be a convex, continuous, coercive function, and for any $\varepsilon > 0$ let us consider the approximate minimization problem

$$(\mathcal{P}_\varepsilon) \qquad \min\{f(x) + \varepsilon g(x) : x \in X\}.$$

Then for each $\varepsilon > 0$ there exists a solution u_ε of $(\mathcal{P}_\varepsilon)$. The sequence $\{u_\varepsilon; \varepsilon \to 0\}$ remains bounded if and only if the solution set argmin f *is nonvoid. In that case, every weak limit point $\hat{u}$ of the sequence $\{u_\varepsilon; \varepsilon \to 0\}$ minimizes f on X and satisfies*

$$\text{(VSP)} \qquad g(\hat{u}) \leq g(v) \quad \textit{for every } v \in \operatorname{argmin} f.$$

Moreover, the corresponding optimal values converge. More precisely,

$$\lim_{\varepsilon\to 0} \frac{1}{\varepsilon}\,[f(u_\varepsilon) - \inf_X f] = 0,$$

$$\lim_{\varepsilon\to 0} g(u_\varepsilon) = \min\{g(v) : v \in \operatorname{argmin} f\},$$

and there exists some real-valued function $\varepsilon \to r(\varepsilon)$ satisfying $\lim_{\varepsilon\to 0} r(\varepsilon) = 0$ such that

$$\inf_X(f + \varepsilon g) = \inf_X f + \varepsilon \inf[g(v) : v \in \operatorname{argmin} f] + \varepsilon r(\varepsilon).$$

Assuming, moreover, that g is strictly convex and satisfies

$$[u_\varepsilon \textit{ weakly converges to } u \textit{ and } g(u_\varepsilon) \to g(u)] \Rightarrow [u_\varepsilon \textit{ strongly converges to } u],$$

the whole sequence (u_ε) strongly converges *to the unique element $\hat{u}$ that minimizes g over the set* argmin f.

Proof of Theorem 5.1. The conclusions of Theorem 5.1 follow directly from Theorem 2.1 by taking the topology τ equal to the weak topology of the space X. For each $\varepsilon > 0$, the function $f_\varepsilon = f + \varepsilon g$ is convex lower semicontinuous and coercive on the reflexive Banach space X. Hence, there exists at least a solution u_ε of the minimization problem $(\mathcal{P}_\varepsilon)$. If the sequence u_ε remains bounded in X, then every weak limit point

$\hat{u}$ of the sequence u_ε minimizes f on X, and the set argmin f is nonvoid. Conversely, if the set argmin f is nonvoid, by using (2.4),

$$g(u_\varepsilon) \leq g(v) \quad \text{for all } v \in \operatorname{argmin} f.$$

Since g is a finite-valued function, the sequence $g(u_\varepsilon)$ remains bounded and, from the coerciveness assumption on g, so is the sequence (u_ε). So, the sequence (u_ε) is weakly relatively compact in X. Since f and g are convex lower semicontinuous and hence weakly lower semicontinuous, one can directly apply the conclusions of Theorem 2.1. □

In a large number of applications, one must have some more geometrical and differential information on the viscosity trajectory. For the sake of simplicity, we examine this question in Hilbert spaces (from now on, throughout this section, we assume that X is a Hilbert space) and take $g(x) = \frac{1}{2}\|x\|^2$. Let us first reformulate the above result in this setting with the help of the corresponding Euler equations.

COROLLARY 5.2. *Let X be a Hilbert space and $f\colon X \to \mathbb{R} \cup \{+\infty\}$ be a convex lower semicontinuous proper function. Given $z \in X$, for each $\varepsilon > 0$ there exists a unique solution u_ε of the equation*

$$\varepsilon u_\varepsilon + \partial f(u_\varepsilon) \ni z.$$

The sequence $\{u_\varepsilon; \varepsilon \to 0\}$ remains bounded if and only if the set $S = (\partial f)^{-1} z$ is nonvoid. In this case, the whole sequence (u_ε) strongly converges to $\hat{u}$, the unique element of minimal norm of S

$$\lim_{\varepsilon \to 0} u_\varepsilon = \operatorname{proj}_{(\partial f)^{-1} z} 0 = \hat{u}.$$

PROPOSITION 5.3. *Let X be a Hilbert space and $f\colon X \to \mathbb{R} \cup \{+\infty\}$ be a convex lower semicontinuous proper function such that* argmin $f = (\partial f)^{-1} 0$ *is nonempty. Let us consider the mapping u: $(0, +\infty) \to X$, which to each $\varepsilon > 0$ associates $u(\varepsilon)$, the unique solution of*

$$\varepsilon u_\varepsilon + \partial f(u_\varepsilon) \ni 0.$$

The mapping u is Lipschitz continuous on $(\delta, +\infty)$ for each $\delta > 0$. For almost every $\varepsilon > 0$ the following inequalities hold:

$$\varepsilon \left\| \frac{du}{d\varepsilon}(\varepsilon) \right\|^2 + \left\langle u(\varepsilon), \frac{du}{d\varepsilon}(\varepsilon) \right\rangle \leq 0$$

and

$$\left\| \frac{du}{d\varepsilon}(\varepsilon) \right\| \leq \frac{C}{\varepsilon} \quad \text{for some } C \geq 0.$$

In addition,

$$\int_0^{+\infty} \varepsilon \left\| \frac{du}{d\varepsilon}(\varepsilon) \right\|^2 < +\infty.$$

Moreover, as ε decreases to zero, $\|u(\varepsilon)\|$ increases to $\operatorname{dist}(0, (\partial f)^{-1} 0)$ *and $f(u(\varepsilon))$ decreases to* $\inf_X f$.

Proof of Proposition 5.3. Let $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ be arbitrary. By definition of $u(\varepsilon)$, we have $-\varepsilon_1 u(\varepsilon_1) \in \partial f(u(\varepsilon_1))$ and $-\varepsilon_2 u(\varepsilon_2) \in \partial f(u(\varepsilon_2))$. By the monotonicity of the operator ∂f,

$$\langle -\varepsilon_1 u(\varepsilon_1) + \varepsilon_2 u(\varepsilon_2), u(\varepsilon_1) - u(\varepsilon_2) \rangle \geq 0.$$

Equivalently,

$$\langle -\varepsilon_1 u(\varepsilon_1) + \varepsilon_1 u(\varepsilon_2), u(\varepsilon_1) - u(\varepsilon_2)\rangle + \langle (\varepsilon_2 - \varepsilon_1)u(\varepsilon_2), u(\varepsilon_1) - u(\varepsilon_2)\rangle \geq 0,$$

that is,

$$\varepsilon_1 \|u(\varepsilon_1) - u(\varepsilon_2)\|^2 + (\varepsilon_2 - \varepsilon_1)\langle u(\varepsilon_2), u(\varepsilon_2) - u(\varepsilon_1)\rangle \leq 0. \tag{5.1}$$

Therefore,

$$\|u(\varepsilon_1) - u(\varepsilon_2)\| \leq \frac{|\varepsilon_2 - \varepsilon_1|}{\varepsilon_1} \|u(\varepsilon_2)\|.$$

Note that the trajectory $u(\cdot)$ remains bounded since we have assumed that argmin $f = (\partial f)^{-1}0$ is nonempty. Hence, the mapping u is Lipschitz continuous on $(\delta, +\infty)$ for each $\delta > 0$. By using (5.1), we obtain that for almost every $\varepsilon > 0$

$$\varepsilon \left\| \frac{du}{d\varepsilon}(\varepsilon) \right\|^2 + \left\langle u(\varepsilon), \frac{du}{d\varepsilon}(\varepsilon) \right\rangle \leq 0. \tag{5.2}$$

This implies

$$\left\| \frac{du}{d\varepsilon}(\varepsilon) \right\| \leq \frac{\|u(\varepsilon)\|}{\varepsilon} \leq \frac{C}{\varepsilon}. \tag{5.3}$$

Moreover, by integrating (5.2) and noticing that $2\langle u(\varepsilon), \frac{du}{d\varepsilon}(\varepsilon)\rangle = \frac{d}{d\varepsilon}\|u(\varepsilon)\|^2$, we obtain

$$\int_0^{+\infty} \varepsilon \left\| \frac{du}{d\varepsilon}(\varepsilon) \right\|^2 < +\infty. \tag{5.4}$$

It follows from (5.2) that

$$\frac{d}{d\varepsilon}\|u(\varepsilon)\|^2 = 2\left\langle u(\varepsilon), \frac{du}{d\varepsilon}(\varepsilon) \right\rangle \leq 0,$$

and hence $\|u(\varepsilon)\|$ increases to $\operatorname{dist}(0, (\partial f)^{-1}0)$ as ε decreases to zero. Moreover, from

$$\begin{aligned} \frac{d}{d\varepsilon} f(u(\varepsilon)) &= \left\langle \partial f(u(\varepsilon)), \frac{du}{d\varepsilon}(\varepsilon) \right\rangle \\ &= \left\langle -\varepsilon u(\varepsilon)), \frac{du}{d\varepsilon}(\varepsilon) \right\rangle \end{aligned}$$

we obtain by using (5.2) that this quantity is greater than or equal to zero, and hence $f(u(\varepsilon))$ decreases to $\inf_X f$ as ε decreases to zero. □

The preceding estimations on the viscosity trajectory play a central role in the following dynamical asymptotical result, which combines the generalized steepest descent method and the viscosity approximation.

THEOREM 5.4. *Let X be a Hilbert space and $f\colon X \to \mathbb{R} \cup \{+\infty\}$ be a convex lower semicontinuous proper function such that* argmin $f = (\partial f)^{-1}0$ *is nonempty. For a given u_0 belonging to the domain of f, let us consider the Cauchy problem*

$$\begin{cases} \dfrac{du}{dt} + \partial f(u(t)) + \dfrac{1}{t^\lambda} u(t) \ni 0, \\ u(0) = u_0, \end{cases}$$

where $\lambda > 0$ is a real nonnegative parameter. Then

a) *if $0 < \lambda \leq 1, u(t)$ strongly converges to the element of minimal norm of* argmin $f = (\partial f)^{-1}0$ *as t goes to $+\infty$;*

b) *if $\lambda > 1, u(t)$ weakly converges to some element u_∞ belonging to* argmin f *as t goes to $+\infty$.*

Proof of Theorem 5.4. When λ is strictly greater than one, the viscosity term is small and the trajectory behaves like the classical steepest descent method for convex minimization problems in Hilbert spaces. This type of situation was analyzed in a more general context by Furuya, Miyashiba, and Kenmochi in [28]. When λ is less than or equal to one, which is the interesting case, then there is a viscosity effect. One can prove (see Attouch and Cominetti [5] for the case $\lambda < 1$ and Guillaume [30] when $\lambda = 1$) that the trajectory of the above dynamical system asymptotically behaves like the viscosity trajectory. As we already claimed, the proof of this result relies on estimations (5.3) and (5.4), which are related to the length of the viscosity trajectory. □

6. Applications. In this section, we discuss some other situations where the results of the previous sections can be directly applied. Let us first consider a model example for semicoercive variational problems.

6.1. Elliptic semicoercive equations. Convex minimization problems with some lack of coerciveness, often called semicoercive variational problems, have been intensively studied because of their practical importance. One of the most famous examples is the (nonlinear) Neumann problem

$$\text{(E)} \qquad \begin{cases} -\Delta u + \beta(u) \ni h & \text{on } \Omega, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega, \end{cases}$$

where β is a maximal monotone graph in $\mathbb{R} \times \mathbb{R}$. Recent progress concerning the existence of solutions to noncoercive (not necessarily convex) minimization problems were obtained by using the concept of recession function; see Baiocchi, Buttazzo, Gastaldi, and Tomarelli [12]. A closely related approach based on the concept of recession operator for maximal monotone operators and that can be applied to equations that do not necessarily derive from a minimization problem was developed by Attouch, Chbani, and Moudafi [4]. The key step for obtaining such existence results is to prove that the sequence (u_ε) of the approximate viscosity problems remains bounded, as is claimed in Corollary 5.2. A sufficient condition for obtaining such a uniform bound, besides some compactness assumptions, is that the kernel of the associated recession operator is a subspace [4, Thm. 3.1]. This gives rise to the so-called compatibility conditions.

Let us return to the above example and show how the viscosity method works in this situation. Take $X = L^2(\Omega)$, where Ω is an open regular bounded subset of $\mathbb{R}^N, N \geq 1$, and assume that $h \in L^2(\Omega)$. Let β be a maximal monotone graph in $\mathbb{R} \times \mathbb{R}$ and j_β a convex function whose subdifferential is equal to β, that is, $\partial j_\beta = \beta$. Classically, equation (E) is the Euler equation of the convex minimization problem

$$\min\{F(v) - \langle h, v\rangle; v \in L^2(\Omega)\},$$

where

$$F(v) = \frac{1}{2}\int_\Omega |Dv|^2\,dx + \int_\Omega j_\beta(v)\,dx \quad \text{if } v \in H^1(\Omega),$$

$$+\infty \quad \text{otherwise.}$$

The viscosity approximate minimization problem is

$$\min\left\{F(v) - \langle h, v\rangle + \frac{\varepsilon}{2}\int_\Omega v(x)^2\,dx; v \in L^2(\Omega)\right\}.$$

The Euler equation of the viscosity approximate problem

$$(\mathrm{E}_\varepsilon) \qquad \begin{cases} \varepsilon u - \Delta u + \beta(u) \ni h & \text{on } \Omega, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega \end{cases}$$

has a unique solution that is denoted by u_ε. A sufficient condition (see, for example, [4, 12]) for the sequence (u_ε) to be bounded in $L^2(\Omega)$, and hence for the existence of a solution to problem (E), is that the following compatibility condition holds:

$$\frac{1}{|\Omega|}\int_\Omega h \in \text{interior of the range of } \beta.$$

Then the sequence (u_ε) norm converges in $L^2(\Omega)$ to $\hat{u}$, the unique element of minimal L^2 norm of the solution set of equation (E). This combined with the convergence of the optimal values

$$\lim_{\varepsilon\to 0} F(u_\varepsilon) = \inf_X F = F(\hat{u})$$

yields by a standard argument (use Fatou's lemma and the fact that the weak convergence and the convergence of the norms in H^1 imply the norm convergence) that the sequence (u_ε) norm converges in $H^1(\Omega)$.

When $\beta = 0$, the structure of the solution set of (E) is easy to describe and the solution of (E) is defined up to an arbitrary additive constant. Then the viscosity selection principle yields that for any constant C, the following equality holds:

$$\langle 0 - \hat{u}, C\rangle = 0\cdot$$

that is, the integral of $\hat{u}$ must be equal to zero. In that case, the viscosity solution $\hat{u}$ is the unique solution of the system

$$(\mathrm{VSP}) \qquad \begin{cases} -\Delta u = h & \text{on } \Omega, \\ \displaystyle\int_\Omega u(x)\,dx = 0, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases}$$

6.2. Generalized Moreau–Yosida approximation and Hamilton–Jacobi equations.

PROPOSITION 6.1. *Let X be a reflexive Banach space. Let ϕ: $X \to \mathbb{R}^+$ be a potential, that is, a continuously differentiable, strictly convex, and coercive function with $\phi(0) = 0$, such that*

(6.1) $[y_\varepsilon$ *weakly converges to* y *and* $\phi^*(y_\varepsilon) \to \phi^*(y)] \Rightarrow [y_\varepsilon$ *strongly converges to* $y]$.

Given f: $X \to \mathbb{R} \cup \{+\infty\}$ a convex lower semicontinuous proper function, for any $\varepsilon > 0$ let us define

$$f_\varepsilon(x) = \min\left\{ f(v) + \varepsilon\phi\left(\frac{x-v}{\varepsilon}\right) : v \in X \right\}.$$

Then f_ε is a continuously differentiable convex function. The sequence f_ε increases monotonically to f as ε decreases to zero. The following dual formulation for f_ε holds:

$$f_\varepsilon(x) = -\min\{f^*(y) - \langle x, y\rangle + \varepsilon\phi^*(y) : y \in X^*\}.$$

It follows from the above "viscosity" formulation that

$$\lim_{\varepsilon \to 0} Df_\varepsilon(x) = \operatorname{argmin}\{\phi^*(y) : y \in \partial f(x)\}$$

when this last set is reduced to a singleton, and

$$\frac{d}{d\varepsilon} f_\varepsilon(x)|_{\varepsilon=0} = -\min\{\phi^*(y) : y \in \partial f(x)\}.$$

Moreover, the mapping $(t, x) \to f_t(x)$ is the (viscosity) solution of the Hamilton–Jacobi equation

$$\frac{\partial F}{\partial t} + \phi^*\left(\frac{\partial F}{\partial x}\right) = 0, \qquad \lim_{t \to 0} F(t, x) = f(x).$$

Proof of Proposition 6.1. The continuous differentiability property of the generalized Moreau–Yosida approximation was analyzed in the Hilbert case in [7, Prop. 3.4]; for the classical case see [16]. We use here the recent version given in Plazanet's thesis [48, §C, Thm. 13]. Since f_ε is a convex continuous (and hence lower semicontinuous) function, we have

$$f_\varepsilon = (f_\varepsilon)^{**}. \tag{6.2}$$

We notice that f_ε is the inf-convolution (epi-graphical sum) of f and of the kernel $\varepsilon\phi(\frac{\cdot}{\varepsilon})$. It is a classical result that the dual operation of the inf-convolution is the ordinary sum of the conjugate functions (see [7, 8]). Moreover, an elementary computation yields that

$$\left[\varepsilon\phi\left(\frac{\cdot}{\varepsilon}\right)\right]^* = \varepsilon\phi^*.$$

It follows from (6.2) that

$$f_\varepsilon = (f^* + \varepsilon\phi^*)^*,$$

that is,

$$f_\varepsilon(x) = -\min\{f^*(y) - \langle x, y\rangle + \varepsilon\phi^*(y) : y \in X^*\}. \tag{6.3}$$

Note that, because of the continuity of ϕ, the conjugate function ϕ^* is coercive and the above infimum is actually a minimum. Let us now compute the gradient of f_ε, which we denote by $y_\varepsilon = Df_\varepsilon(x)$. A similar duality argument as above yields that

$$\begin{aligned} y_\varepsilon = Df_\varepsilon(x) &\Leftrightarrow x \in \partial(f_\varepsilon)^*(y_\varepsilon) \\ &\Leftrightarrow x \in \partial(f^* + \varepsilon\phi^*)(y_\varepsilon) \\ &\Leftrightarrow 0 \in \partial(f^* - \langle x, \,.\,\rangle + \varepsilon\phi^*)(y_\varepsilon), \end{aligned}$$

which expresses that y_ε is a solution of the minimization problem (6.3). We are now in the situation analyzed in Theorem 5.1. The viscosity function $g(y) := \phi^*(y)$ is convex continuous (because ϕ is coercive), coercive, and by assumption (6.1) satisfies

$$[y_\varepsilon \text{ weakly converges to } y \text{ and } g(y_\varepsilon) \to g(y)] \Rightarrow [y_\varepsilon \text{ strongly converges to } y].$$

Noticing that $\operatorname{argmin}\{f^*(\cdot) - \langle x, \cdot\rangle\} = \{y \in X^*; y \in \partial f(x)\}$, we obtain that for all x in the domain of the subdifferential operator ∂f

$$\lim_{\varepsilon\to 0} Df_\varepsilon(x) = \operatorname{argmin}\{\phi^*(y)\colon\ y \in \partial f(x)\}$$

and

$$\frac{d}{d\varepsilon} f_\varepsilon(x)|_{\varepsilon=0} = -\min\{\phi^*(y)\colon\ y \in \partial f(x)\}.$$

The differentiability property of $(t, x) \to f_t(x)$ and the corresponding relations between the partial derivatives are easily extended from $t = 0$ to an arbitrary $t > 0$ by using the semigroup property $(f_t)_s = f_{t+s}$. We finally obtain the Hamilton–Jacobi equation

$$\frac{\partial f_\varepsilon}{\partial \varepsilon} + \phi^*\left(\frac{\partial f_\varepsilon}{\partial x}\right) = 0, \qquad \lim_{\varepsilon\to 0} f_\varepsilon(x) = f(x).$$

Replacing ε by t, we obtain the desired result. □

COROLLARY 6.2 (Moreau–Yosida approximation). *Assume that X is a reflexive Banach space such that both X and X^* are strictly convex and Kadec. For any convex lower semicontinuous and proper function $f\colon X \to \mathbb{R} \cup \{+\infty\}$, and for any $\varepsilon > 0$, the Moreau–Yosida approximation of index $\varepsilon > 0$ of f is defined by*

$$f_\varepsilon(x) := \min\left\{f(v) + \frac{1}{2\varepsilon}\,\|x - v\|^2 \colon v \in X\right\}.$$

Then f_ε is a continuously differentiable convex function. The sequence f_ε is increasingly convergent to f as ε decreases to zero. Moreover, the following dual formulation for f_ε holds:

$$f_\varepsilon(x) = -\min\left\{f^*(y) - \langle x, y\rangle + \frac{\varepsilon}{2}\,\|y\|^2 \colon y \in X^*\right\}.$$

It follows from the above viscosity formulation that

$$\lim_{\varepsilon\to 0} Df_\varepsilon(x) = \operatorname{argmin}\{\|y\|^2\colon\ y \in \partial f(x)\};$$

that is, for all x belonging to dom ∂f,

$$\lim_{\varepsilon\to 0} (\partial f)_\varepsilon(x) = (\partial f)^0(x),$$

where $(\partial f)^0(x)$ *is the element of minimal norm of the closed convex nonempty set* $(\partial f)(x)$. *Moreover,*

$$\frac{d}{d\varepsilon} f_\varepsilon(x)|_{\varepsilon=0} = -\min\left\{\frac{1}{2}\|y\|^2 \colon y \in \partial f(x)\right\},$$

and the mapping $(t,x) \to f_t(x)$ *is the (viscosity) solution of the Hamilton–Jacobi equation*

$$\frac{\partial F}{\partial t} + \frac{1}{2}\left\|\frac{\partial F}{\partial x}\right\|^2 = 0, \qquad \lim_{t\to 0} F(t,x) = f(x).$$

Acknowledgments. We are particularly grateful to Professors B. Lemaire, A. Seeger, D. Torralba, and C. Zalinescu and to the anonymous referees whose comments and suggestions have done much to bring this article to its present form.

REFERENCES

[1] G. Anzellotti and S. Baldo, *Asymptotic development by* Γ*-convergence*, Appl. Math. Optim., 27 (1993), pp. 105–123.

[2] H. Attouch, *Variational convergence for functions and operators*, Appl. Math., Pitman, London, 1984.

[3] ———, *Convergence of viscosity type approximation methods to particular solutions of variational problems*, in Proc. Variational Methods, Nonlinear Analysis and Differential Equations, E.C.I.G.-Genova, M. Chicco, P. Oppezzi, and T. Zolezzi, eds., 1994, pp. 13–43.

[4] H. Attouch, Z. Chbani, and A. Moudafi, *Recession operators and solvability of variational problems in reflexive Banach spaces*, in Calculus of Variations, Homogenization and Continuum Mechanics, G. Bouchitte, G. Buttazzo, and P. Suquet, eds., Ser. Adv. Math. Appl. Sci. 18, World Scientific, Teaneck, NJ, 1993, pp. 51–67.

[5] H. Attouch and R. Cominetti, *A dynamical approach to convex minimization coupling approximation with the steepest descent method*, J. Differential Equations, to appear.

[6] H. Attouch and M. Théra, *A general duality principle for the sum of two operators*, J. Convex Anal., 3, 1, 1996.

[7] H. Attouch and R. Wets, *Epigraphical analysis*, in Analyse non linéaire, H. Attouch, J. P. Aubin, F. Clarke, and I. Ekeland, eds., Gauthier-Villars, Paris, 1989, pp. 73–100.

[8] ———, *Approximation and convergence in nonlinear optimization*, in Nonlinear Programming, Vol. 4, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1981.

[9] J. P. Aubin and I. Ekeland, *Applied Nonlinear Analysis*, J. Wiley, New York, 1984.

[10] J. P. Aubin and H. Frankowska, *Set-Valued Analysis*, Birkhäuser, Boston, 1990.

[11] A. Auslender and R. Cominetti, *First and second order sensitivity analysis of nonlinear programs under directional constraint qualification condition*, Optimization, 21 (1990), pp. 351–363.

[12] C. Baiocchi, G. Buttazzo, F. Gastaldi, and F. Tomarelli, *General existence theorems for unilateral problems in continuum mechanics*, Arch. Rational Mech. Anal., 100 (1988), pp. 149–189.

[13] A. Bensoussan and P. Kenneth, *Sur l'analogie entre les méthodes de régularisation et de pénalisation*, Rev. Inform. Recherche Opér., 13 (1969), pp. 13–26.

[14] J. F. Bonnans, *Directional derivatives of optimal solutions in nonlinear programming*, J. Optim. Theory Appl., 73 (1992), pp. 27–45.

[15] G. Bouchitté, *Singular perturbations of variational problems arising from a two phases transition model*, Appl. Math. Optim., 21 (1990), pp. 289–314.

[16] H. Brezis, *Opérateurs Maximaux Monotones et Semi Groupes de Contractions dans les Espaces de Hilbert*, North-Holland, Amsterdam, 1973.

[17] G. Buttazzo, *Thin Structures in Elasticity: The Variational Method*, preprint no. 117, Università di Ferrara, 1989.

[18] G. Buttazzo and G. Dal Maso, Γ*-convergence and optimal control problems*, J. Optim. Theory Appl., 38 (1982), pp. 385–407.

[19] ———, *Γ-convergence et problèmes de perturbations singulières*, C. R. Acad. Sci. Paris Ser. I Math., 296 (1983), pp. 649–651.

[20] R. COMINETTI AND J.-P. DUSSAULT, *A Stable Exponential-Penalty Algorithm with Superlinear Convergence*, Publicaciones Tecnicas Universidad de Chile, Santiago, Chile, MA-91-B-3374, 1991.

[21] R. COMINETTI AND J. SAN MARTIN, *Trajectory Analysis for the Exponential Penalty Method in Linear Programming*, Publicaciones Tecnicas Universidad de Chile, Santiago, Chile, MA-92-B-395, 1992.

[22] M. G. CRANDALL AND P. L. LIONS, *Viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., 277 (1983), pp. 1–42.

[23] G. DAL MASO, *Γ-Convergence in Calculus of Variations*, Birkhäuser, Boston, 1993.

[24] E. DE GIORGI, *Convergence problems for functionals and operators*, in Proc. Int. Meeting on *Recent Methods in Nonlinear Analysis*, Rome 1978, E. De Giorgi, E. Magenes, and U. Mosco, eds., Pitagora editrice, Bologne, 1979, pp. 131–188.

[25] A. L. DONTCHEV AND T. ZOLEZZI, *Well-posed optimization problems*, Lecture Notes in Mathematics 1543, Springer-Verlag, New York, 1993.

[26] I. EKELAND ET R. TEMAM, *Analyse Convexe et Problèmes Variationnels*, Dunod & Gauthiers-Villars, Paris, 1974.

[27] W. H. FLEMING AND W. M. MCENEANEY, *Risk Sensitive Control on an Infinite Time Horizon*, working paper, 1993.

[28] H. FURUYA, K. MIYASHIBA, AND N. KENMOCHI, *Asymptotic behavior of solutions to a class of nonlinear evolution equations*, J. Differential Equations, 62 (1986), pp. 73–94.

[29] E. GIUSTI, *Minimal Surfaces and Functions of Bounded Variations*, Birkhäuser, Boston, 1984.

[30] S. GUILLAUME, personal communication.

[31] A. HARAUX ET F. MURAT, *Perturbations singulières et problèmes de contrôle optimal: deux cas bien posés*, C. R. Acad. Sci. Paris Sér. I Math., 297 (1983), pp. 21–24.

[32] ———, *Perturbations singulières et problèmes de contrôle optimal: un cas mal posé*, C. R. Acad. Sci. Paris Sér. I Math., 297 (1983), pp. 93–96.

[33] P. KALL, *Approximation to optimization problems: an elementary review*, Math. Oper. Res., 11 (1986), pp. 9–18.

[34] R. V. KOHN AND L. NIRENBERG, *Degenerate elliptic parabolic equations of second order*, Comm. Pure Appl. Math., 20 (1967), pp. 797–892.

[35] R. V. KOHN AND P. STERNBERG, *Local minimizers and singular perturbations*, Proc. Roy. Soc. Edinburgh, to appear.

[36] O. A. LADYZENSKAYA AND N. N. URALCEVA, *Local estimates for gradients of solutions of non uniformly elliptic and parabolic equations*, Comm. Pure Appl. Math., 18 (1970), pp. 677–703.

[37] B. LEMAIRE, *Régularisation et pénalisation en optimisation convexe*, Sém. Anal. Convexe Montpellier, Vol. 1, exposé 17, 1971.

[38] ———, *Quelques résultats récents sur l'algorithme proximal*, Sém. Anal. Numér. Toulouse, 1989.

[39] ———, *About the convergence of the proximal method*, in Advances in Optimization, Proceedings Lambrecht 1991, *Lecture Notes in Econom. and Math. Systems* 382, Springer-Verlag, New York, 1992, pp. 39–51.

[40] C. LEMAIRE-MISONNE, *Optimisation à plusieurs critères hiérarchisés: méthode d'approximation*, Preprint Université des Sciences et Techniques du Languedoc, Colloque d'Analyse Numérique d'Epinal, Montpellier, France, 1972.

[41] J. L. LIONS, *Quelques Méthodes de Résolution des Problèmes aux Limites Non Linéaires*, Dunod & Gauthiers-Villars, Paris, 1969.

[42] ———, *Perturbations singulières dans les problèmes aux limites et en contrôle optimal*, Lecture Notes in Mathematics 323, Springer-Verlag, New York, 1973.

[43] ———, *Contrôle des Systèmes Distribués Singuliers*, Gauthiers-Villars, Paris, 1983.

[44] D. G. LUENBERGER, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, MA, 1984.

[45] O. L. MANGASARIAN, *Iterative solution of linear programs*, SIAM J. Numer. Anal., 18 (1981), pp. 606–614.

[46] L. MODICA, *Gradient theory of phase transitions and minimal interface criteria*, Arch. Rational Mech. Anal., 98 (1987), pp. 123–142.

[47] J. P. PENOT, *Miscellaneous incidences of convergence theories in optimization*, Set-Valued Analysis, 2 (1994), pp. 259–274.

[48] P. PLAZANET, *Contributions à l'Analyse des Fonctions Convexes*, Thèse Université P. Sabatier, Toulouse, France, 1990.

[49] B. T. POLYAK, *Sharp minimum*, International workshop on augmented Lagrangians, Vienna, 1979.

[50] S. M. ROBINSON, *Local Epi-Continuity and Local Optimization*, Math. Programming, 37 (1987), pp. 208–222.

[51] R. T. ROCKAFELLAR, *Directional differentiability of the optimal value in a nonlinear programming problem*, Math. Programming Stud., 21 (1984), pp. 87–111.

[52] ———, *First and second order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 307 (1988), pp. 75–108.

[53] R. T. ROCKAFELLAR AND R. WETS, Set-valued analysis and subdifferential calculus, in preparation.

[54] A. SEEGER, *Analyse du Second Ordre de Problèmes Non Différentiables*, Thèse Université P. Sabatier, Toulouse, France, 1986.

[55] P. STERNBERG, *The Effect of a Singular Perturbation on Nonconvex Variational Problems*, Ph.D. Thesis, New York University, 1986.

[56] R. TEMAM, *Problèmes Mathématiques en Plasticité*, Gauthiers-Villars, Paris, 1983.

[57] D. TORRALBA, *Développements asymptotiques pour les méthodes d'approximation par viscosité*, C. R. Acad. Sci. Paris Sér. I Math., 322 (1996), pp. 123–128.

[58] A. N. TIKHONOV, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl. 4 (1963), pp. 1035–1038.

[59] A. TIKHONOV AND V. ARSÉNINE, *Méthodes de Résolution de Problèmes Mal Posés*, Traduction française, MIR, 1976.

[60] C. ZALINESCU, *A note on d-stability of convex programs and limiting Lagrangians*, Math. Programming, 53 (1992), pp. 267–277.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 807–822, August 1996
© 1996 Society for Industrial and Applied Mathematics
015

INCREMENTAL LEAST SQUARES METHODS AND THE EXTENDED KALMAN FILTER*

DIMITRI P. BERTSEKAS†

Abstract. In this paper we propose and analyze nonlinear least squares methods which process the data incrementally, one data block at a time. Such methods are well suited for large data sets and real time operation and have received much attention in the context of neural network training problems. We focus on the extended Kalman filter, which may be viewed as an incremental version of the Gauss–Newton method. We provide a nonstochastic analysis of its convergence properties, and we discuss variants aimed at accelerating its convergence.

Key words. optimization, least squares, Kalman filter

AMS subject classifications. 93-11, 90C30, 65K10

1. Introduction.

We consider least squares problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) = \|g(x)\|^2 = \sum_{i=1}^{m} \|g_i(x)\|^2 \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{1}$$

where g is a continuously differentiable function with component functions $g_1, \ldots, g_m$, where $g_i : \Re^n \to \Re^{r_i}$. Here we write $\|z\|$ for the usual Euclidean norm of a vector z, that is, $\|z\| = \sqrt{z'z}$, where the prime denotes transposition. We also write ∇g_i for the $n \times r_i$ gradient matrix of g_i and ∇g for the $n \times (r_1 + \cdots + r_m)$ gradient matrix of g. Least squares problems very often arise in contexts where the functions g_i correspond to measurements that we are trying to fit with a model parameterized by x. Motivated by this context, we refer to each component g_i as a *data block*, and we refer to the entire function $g = (g_1, \ldots, g_m)$ as the *data set*.

One of the most common iterative methods for solving least squares problems is the Gauss–Newton method, given by

$$x^{k+1} = x^k - \alpha^k \big(\nabla g(x^k) \nabla g(x^k)'\big)^{-1} \nabla g(x^k) g(x^k), \tag{2}$$

where α^k is a positive stepsize, and we assume that the $n \times n$ matrix $\nabla g(x^k) \nabla g(x^k)'$ is invertible. The case $\alpha^k = 1$ corresponds to the pure form of the method, where x^{k+1} is obtained by linearizing g at the current iterate x^k and minimizing the norm of the linearized function, that is,

$$x^{k+1} = \arg\min_{x \in \Re^n} \|g(x^k) + \nabla g(x^k)'(x - x^k)\|^2 \quad \text{if } \alpha^k = 1. \tag{3}$$

In problems where there are many data blocks, the Gauss–Newton method may be ineffective because the size of the data set makes each iteration very costly. For such problems it may be much better to use an *incremental method* that does not

*Received by the editors May 27, 1994; accepted for publication (in revised form) April 4, 1995. This research was supported by NSF grant 9300494-DMI.

†Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 (dimitrib@mit.edu).

wait to process the entire data set before updating x, as discussed in [Ber95]. Instead, the method cycles through the data blocks in sequence and updates the estimate of x after each data block is processed. A further advantage is that estimates of x become available as data is accumulated, making the approach suitable for real time operation. Such methods include the Widrow–Hoff least-mean-square (LMS) algorithm [WiH60], [WiS85] for the case where the data blocks are linear, and other steepest-descent-like methods for nonlinear data blocks that have been used extensively for the training of neural networks under the generic name of *backpropagation methods*. A cycle through the data set of a typical example of such a method starts with a vector x^k and generates x^{k+1} according to

$$x^{k+1} = \psi_m,$$

where ψ_m is obtained at the last step of the recursion

$$\psi_i = \psi_{i-1} - \alpha^k \nabla g_i(\psi_{i-1}) g_i(\psi_{i-1}), \qquad i = 1, \ldots, m,$$

α^k is a positive stepsize, and $\psi_0 = x^k$.

Backpropagation methods are often effective, and they are supported by stochastic [PoT73], [Lju77], [KuC78], [Pol87], [BeT89], [Whi89a], [Whi89b], [Gai93], [BeT96] as well as deterministic convergence analyses [Luo91], [Gri93], [LuT93], [MaS94], [Man93], [BeT96]. The main difference between stochastic and deterministic methods of analysis is that the former apply to an infinite data set (one with an infinite number of data blocks) satisfying some statistical assumptions, while the latter apply to a finite data set. There are also parallel asynchronous versions of backpropagation methods and corresponding stochastic [Tsi84], [TBA86], [BeT89], [Gai93] as well as deterministic convergence results [Tsi84], [TBA86], [BeT89], [MaS94]. However, backpropagation methods typically have a slow convergence rate not only because they are first-order steepest-descent-like methods, but also because they require a diminishing stepsize $\alpha^k = O(1/k)$ for convergence. If α^k is instead taken to be a small constant, an oscillation within each data cycle typically arises, as shown by [Luo91].

In this paper we focus on methods that combine the advantages of backpropagation methods for large data sets with the often superior convergence rate of the Gauss–Newton method. We thus consider an incremental version of the Gauss–Newton method, which operates in cycles through the data blocks. The $(k+1)$st cycle starts with a vector x^k and a positive semidefinite matrix H^k to be defined later, then updates x via a Gauss–Newton-like iteration aimed at minimizing

$$\lambda(x - x^k)' H^k (x - x^k) + \|g_1(x)\|^2,$$

where λ is a scalar with

$$0 < \lambda \leq 1,$$

then updates x via a Gauss–Newton-like iteration aimed at minimizing

$$\lambda^2(x - x^k)' H^k (x - x^k) + \lambda\|g_1(x)\|^2 + \|g_2(x)\|^2,$$

and similarly continues, with the ith step consisting of a Gauss–Newton-like iteration aimed at minimizing the weighted partial sum

$$\lambda^i(x - x^k)' H^k (x - x^k) + \sum_{j=1}^{i} \lambda^{i-j}\|g_j(x)\|^2.$$

In particular, given x^k, the $(k+1)$st cycle sequentially generates the vectors

(4)
$$\psi_i = \arg\min_{x\in\Re^n}\left\{\lambda^i(x-x^k)'H^k(x-x^k)+\sum_{j=1}^{i}\lambda^{i-j}\|\tilde{g}_j(x,\psi_{j-1})\|^2\right\}, \qquad i=1,\ldots,m,$$

and sets

(5)
$$x^{k+1}=\psi_m,$$

where $\tilde{g}_j(x,\psi_{j-1})$ are the linearized functions

(6)
$$\tilde{g}_j(x,\psi_{j-1})=g_j(\psi_{j-1})+\nabla g_j(\psi_{j-1})'(x-\psi_{j-1})$$

and ψ_0 is the estimate of x at the end of the kth cycle:

(7)
$$x^k=\psi_0.$$

As will be seen later, the quadratic minimizations above can be efficiently implemented using the recursive Kalman filter formulas.

The most common version of the preceding algorithm is obtained when the matrices H^k are updated by the recursion

(8)
$$H^{k+1}=\lambda^m H^k+\sum_{j=1}^{m}\lambda^{m-j}\nabla g_j(\psi_{j-1})\nabla g_j(\psi_{j-1})'.$$

Then for $\lambda=1$ and $H^0=0$, the method becomes the well-known extended Kalman filter (EKF for short) specialized to the case where the state of the underlying dynamical system stays constant and the measurement equation is nonlinear. The EKF was originally conceived as a method for estimating parameters from nonlinear measurements that are generated in real time. The basic idea of the method is to linearize each new measurement around the current value of the estimate and treat the measurement as if it were linear (cf. eq. (4)). The estimate is then corrected to account for the new (linearized) measurement using the convenient Kalman filter formulas (see Lemma 1). The algorithm considered here cycles repeatedly through the data set and is sometimes called the *iterated extended Kalman filter*. For the problem of estimating the state of a dynamic system, a cycle through the data set involves solving a problem of smoothing the estimate of the state trajectory before starting a new cycle (see, e.g., [Bel94]). The matrix H^k has the meaning of the inverse of an approximate error covariance of the estimate x^k. In the case $\lambda<1$, the effect of old data blocks is discounted, and successive estimates produced by the method tend to change more rapidly. In this way one may obtain a faster rate of progress of the method, and this is the main motivation for considering $\lambda<1$.

The EKF has been used extensively in a variety of control and estimation applications (see, e.g., [AWT69], [Jaz70], [Meh71], [THS77], [AnM79], [WeM80]) and has also been suggested for the training of neural networks (see, e.g., [WaT90] and [RRK92]). The version of the algorithm (4)–(8) with $\lambda<1$ has also been proposed by Davidon [Dav76]. Unaware of the earlier work in the control and estimation literature, Davidon described the qualitative behavior of the method together with favorable computational experience for problems with large data sets, but gave no convergence

analysis. The first convergence analysis of the EKF was given by Ljung [Lju79], who assuming $\lambda = 1$ used a stochastic formulation (i.e., an infinite data set) and the ODE approach of [Lju77] to prove satisfactory convergence properties for a version of the EKF that is closely related to the one considered here (Theorem 6.1 of [Lju79], which assumes a stationary measurement equation and additive noise). Ljung also showed that the EKF, when applied to more complex models where the underlying dynamic system is linear but its dynamics depend on x, exhibits complex behavior, including the possible convergence to biased estimates. For such models he suggested the use of a different formulation of the least squares problem involving the innovations process (see also [Urs80]). The algorithms and analysis of the present paper apply to any type of deterministic least squares problem, and thus also apply to Ljung's innovations-based formulation.

A deterministic analysis of the EKF method (4)–(8), where $\lambda < 1$, was given in Pappas's Master's thesis [Pap82]. He considered only the special case where $\min_x \|g(x)\| = 0$ and showed that the EKF converges locally to a nonsingular solution of the system $g(x) = 0$ at a rate that is linear with convergence ratio λ^m. He also argued by example that when $\lambda < 1$ and $\min_x \|g(x)\| > 0$, the iterates ψ_i produced by the EKF within each cycle generally oscillate with a "size" of oscillation that diminishes as λ approaches 1.

The purpose of this paper is to provide a deterministic analysis of the convergence properties of the EKF for the general case where $\min_x \|g(x)\|$ is not necessarily zero. Our analysis is complicated by the lack of an explicit stepsize in the algorithm. In the case where $\lambda = 1$ we show that the limit points of the generated sequence $\{x^k\}$ by the EKF are stationary points of the least squares problem. The idea of the proof is to show that the method involves an implicit stepsize of order $O(1/k)$ and then to apply arguments similar to those used by Tsitsiklis [Tsi84] and Tsitsiklis, Bertsekas, and Athans [TBA86] in their analyses of asynchronous distributed gradient methods, and by Mangasarian and Solodov [MaS94] in their convergence proof of an asynchronous parallel backpropagation method. To improve the rate of convergence of the method, which is sublinear and typically slow, we suggest a convergent and empirically faster variant where λ is initially less than 1 and is progressively increased toward 1.

In addition to dealing more naturally with the case of a finite data set, a nice aspect of the deterministic analysis is that it decouples the stochastic modeling of the data generation process from the algorithmic solution of the least squares problem. In other words, the EKF discussed here will (typically) find a least squares solution even if the least squares formulation is inappropriate for the corresponding real parameter estimation problem. This is a valuable insight because it is sometimes thought that convergence of the EKF depends on the validity of the underlying stochastic model assumptions.

2. The EKF. When the data blocks are linear functions, it takes a single pure Gauss–Newton iteration to find the least squares estimate. This iteration can be implemented as an incremental algorithm, the *Kalman filter*, which we now describe. Assume that the functions g_i are linear and of the form

$$g_i(x) = z_i - C_i x, \tag{9}$$

where $z_i \in \Re^{r_i}$ are given vectors and C_i are given $r_i \times n$ matrices. Let us consider the incremental method that generates the vectors

$$\psi_i = \arg\min_{x \in \Re^n} \sum_{j=1}^{i} \lambda^{i-j} \|z_j - C_j x\|^2, \qquad i = 1, \ldots, m. \tag{10}$$

Then the method can be recursively implemented, as shown by the following well-known proposition (see, e.g., [AnM79]).

PROPOSITION 1 (Kalman filter). *Assuming that the matrix $C_1'C_1$ is positive definite, the least squares estimates*

$$\psi_i = \arg\min_{x\in\Re^n} \sum_{j=1}^{i} \lambda^{i-j}\|z_j - C_j x\|^2, \qquad i = 1,\dots,m,$$

can be generated by the algorithm

$$\psi_i = \psi_{i-1} + H_i^{-1}C_i'(z_i - C_i\psi_{i-1}), \qquad i = 1,\dots,m, \tag{11}$$

where ψ_0 is an arbitrary vector, and the positive-definite matrices H_i are generated by

$$H_i = \lambda H_{i-1} + C_i'C_i, \qquad i = 1,\dots,m, \tag{12}$$

with

$$H_0 = 0.$$

More generally, for all $\bar{i} < i$ we have

$$\psi_i = \psi_{\bar{i}} + H_i^{-1} \sum_{j=\bar{i}+1}^{i} \lambda^{i-j}C_j'(z_j - C_j\psi_{\bar{i}}), \qquad i = 1,\dots,m. \tag{13}$$

The proof of Proposition 1 is obtained by using the following lemma involving two data blocks, the straightforward proof of which is omitted.

LEMMA 1. *Let ζ_1, ζ_2 be given vectors and Γ_1, Γ_2 be given matrices such that $\Gamma_1'\Gamma_1$ is positive definite. Then the vectors*

$$\psi_1 = \arg\min_{x\in\Re^n} \|\zeta_1 - \Gamma_1 x\|^2 \tag{14}$$

and

$$\psi_2 = \arg\min_{x\in\Re^n} \left\{\|\zeta_1 - \Gamma_1 x\|^2 + \|\zeta_2 - \Gamma_2 x\|^2\right\} \tag{15}$$

are also given by

$$\psi_1 = \psi_0 + (\Gamma_1'\Gamma_1)^{-1}\Gamma_1'(\zeta_1 - \Gamma_1\psi_0) \tag{16}$$

and

$$\psi_2 = \psi_1 + (\Gamma_1'\Gamma_1 + \Gamma_2'\Gamma_2)^{-1}\Gamma_2'(\zeta_2 - \Gamma_2\psi_1), \tag{17}$$

where ψ_0 is an arbitrary vector.

The proof of eqs. (12) and (13) of Proposition 1 follows by applying Lemma 1 with the correspondences $\psi_0 \sim \psi_0$, $\psi_1 \sim \psi_{\bar{i}}$, $\psi_2 \sim \psi_i$, and

$$\zeta_1 \sim \begin{pmatrix} \sqrt{\lambda^{i-1}}z_1 \\ \vdots \\ \sqrt{\lambda^{i-\bar{i}}}z_{\bar{i}} \end{pmatrix}, \qquad \Gamma_1 \sim \begin{pmatrix} \sqrt{\lambda^{i-1}}C_1 \\ \vdots \\ \sqrt{\lambda^{i-\bar{i}}}C_{\bar{i}} \end{pmatrix}, \tag{18}$$

$$\zeta_2 \sim \begin{pmatrix} \sqrt{\lambda^{i-\bar{i}-1}} z_{\bar{i}+1} \\ \vdots \\ z_i \end{pmatrix}, \qquad \Gamma_2 \sim \begin{pmatrix} \sqrt{\lambda^{i-\bar{i}-1}} C_{\bar{i}+1} \\ \vdots \\ C_i \end{pmatrix},$$

and by carrying out the straightforward algebra.

Note that the positive definiteness assumption on $C_1'C_1$ in Proposition 1 is needed to guarantee that the first matrix H_1 is positive definite and hence invertible; then the positive definiteness of the subsequent matrices $H_2, \ldots, H_m$ follows from eq. (12). As a practical matter, it is possible to guarantee the positive definiteness of $C_1'C_1$ by lumping a sufficient number of measurements into the first data block (C_1 should contain n linearly independent columns). An alternative is to redefine ψ_i as

$$\psi_i = \arg\min_{x \in \Re^n} \left\{ \delta\lambda^i \|x - \psi_0\|^2 + \sum_{j=1}^{i} \lambda^{i-j} \|z_j - C_j x\|^2 \right\}, \qquad i = 1, \ldots, m, \tag{19}$$

where δ is a small positive scalar. Then it can be seen that ψ_i is generated by the same equations (11) and (12), except that the initial condition $H_0 = 0$ is replaced by

$$H_0 = \delta I, \tag{20}$$

so that $H_1 = \delta I + C_1'C_1$ is positive definite even if $C_1'C_1$ is not. Note, however, that in this case the last estimate ψ_m is only approximately equal to the least squares estimate x^*, even if $\lambda = 1$ (the approximation error depends on the size of δ).

Now consider the general case where the data blocks g_i are nonlinear. Then the EKF can be used, and its first cycle can be implemented by means of the Kalman filter equations of Proposition 1. Using formulas (11) and (12) with the identifications

$$z_i = g_i(\psi_{i-1}) - \nabla g_i(\psi_{i-1})'\psi_{i-1}, \qquad C_i = -\nabla g_i(\psi_{i-1})',$$

the kth cycle of the EKF can be written in the incremental form

$$\psi_i = \psi_{i-1} - H_i^{-1} \nabla g_i(\psi_{i-1}) g_i(\psi_{i-1}), \qquad i = 1, \ldots, m, \tag{21}$$

where the matrices H_i are generated by

$$H_i = \lambda H_{i-1} + \nabla g_i(\psi_{i-1}) \nabla g_i(\psi_{i-1})', \qquad i = 1, \ldots, m, \tag{22}$$

with

$$H_0 = 0. \tag{23}$$

To contrast the EKF with the pure form of the Gauss–Newton method (unit stepsize), note that a single iteration of the latter can be written as

$$x^{k+1} = \arg\min_{x \in \Re^n} \sum_{i=1}^{m} \|\tilde{g}_i(x, x^k)\|^2. \tag{24}$$

Using the formulas of Proposition 1 with the identifications

$$z_i = g_i(x^k) - \nabla g_i(x^k)'x^k, \qquad C_i = -\nabla g_i(x^k)',$$

we can generate x^{k+1} by an incremental algorithm as

$$x^{k+1} = \overline{\psi}_m,$$

where

$$\overline{\psi}_i = \overline{\psi}_{i-1} - \overline{H}_i^{-1} \nabla g_i(x^k)\big(g_i(x^k) + \nabla g_i(x^k)'(\overline{\psi}_{i-1} - x^k)\big), \qquad i = 1, \ldots, m, \tag{25}$$

$\overline{\psi}_0 = x^k$, and the matrices $\overline{H}_i$ are generated by

$$\overline{H}_i = \overline{H}_{i-1} + \nabla g_i(x^k) \nabla g_i(x^k)', \qquad i = 1, \ldots, m, \tag{26}$$

with

$$\overline{H}_0 = 0. \tag{27}$$

Thus, by comparing eqs. (21)–(23) with eqs. (25)–(27), we see that, if $\lambda = 1$, a cycle of the EKF through the data set differs from a pure Gauss–Newton iteration only in that the linearization of the data blocks g_i is done at the corresponding current estimates ψ_{i-1} rather than at the estimate x^k available at the start of the cycle. If the data blocks are linear, the EKF with $\lambda = 1$ and the Gauss–Newton iteration coincide and solve the problem in a single cycle through the data set.

3. Convergence of the EKF. We have considered so far a single cycle of the EKF. To obtain an algorithm that cycles through the data set multiple times, there are two basic approaches. The first approach is to reset the matrix H to some fixed matrix H_0 at the start of each cycle. Unfortunately, the convergence properties of the resulting algorithm are questionable, and one can construct examples where the method diverges, basically because the increments $\psi_i - \psi_{i-1}$ produced by the method (cf. eq. (21)) may be too large. One may attempt to correct this behavior by selecting H_0 to be a sufficiently large multiple of the identity matrix, but this leads to large asymptotic convergence errors (biased estimates), as can be seen through simple examples where the data blocks are linear.

The second approach, which is followed in this paper, is to create a larger data set by concatenating multiple copies of the original data set, that is, by forming what we refer to as *the extended data set*

$$(g_1, g_2, \ldots, g_m, g_1, g_2, \ldots, g_m, g_1, g_2, \ldots). \tag{28}$$

The EKF is then applied to the extended data set and takes the form given in the introduction (eqs. (4)–(8)). The algorithm has the form

$$H_{km+i} = \lambda H_{km+i-1} + \nabla g_i(\psi_{km+i-1}) \nabla g_i(\psi_{km+i-1})', \qquad i = 1, \ldots, m,$$

$$\psi_{km+i} = \psi_{km+i-1} - H_{km+i}^{-1} \nabla g_i(\psi_{km+i-1}) g_i(\psi_{km+i-1}), \qquad i = 1, \ldots, m,$$

where $H_0 = 0$, $\psi_0 = x^0$ is an arbitrary vector, and k indexes the current cycle through the data set. Note that while in the above equations λ is written as a constant, we will later consider the possibility of changing λ in the course of the algorithm. Also, we assume that the matrix $\nabla g_1(x^0) \nabla g_1(x^0)'$ is invertible, so that H_1^{-1} is well defined. However, it can be shown that the convergence result to be given shortly

also holds when H_0 is any positive-definite matrix, in which case the invertibility of $\nabla g_1(x^0)\nabla g_1(x^0)'$ is unnecessary.

We will show that when $\lambda = 1$, the EKF version just described typically converges to stationary points of the least squares problem. The basic reason is that the EKF asymptotically resembles a gradient method with diminishing stepsize of order $O(1/k)$. To get a sense of this, assume that the EKF is applied to the extended data set (28) with $\lambda = 1$. Let us denote by x^k the iterate at the end of the kth cycle through the data set, that is,

$$x^k = \psi_{km}, \qquad k = 1, 2, \ldots.$$

Then, by using eq. (13) with $i = (k+1)m$ and $\bar{i} = km$, we obtain

$$x^{k+1} = x^k - H^{-1}_{(k+1)m} \sum_{i=1}^{m} \nabla g_i(\psi_{km+i-1}) g_i(\psi_{km+i-1}). \tag{29}$$

Now $H_{(k+1)m}$ grows roughly in proportion to $k+1$ because, by eq. (12), we have

$$H_{(k+1)m} = \sum_{j=0}^{k} \sum_{i=1}^{m} \nabla g_i(\psi_{jm+i-1}) \nabla g_i(\psi_{jm+i-1})'. \tag{30}$$

It is therefore reasonable to expect that the method tends to make slow progress when k is large, which means that the vectors ψ_{km+i-1} in eq. (29) are roughly equal to x^k. Thus, for large k, the sum in the right-hand side of eq. (29) is roughly equal to the gradient $\nabla g(x^k) g(x^k)$, while from eq. (30), $H_{(k+1)m}$ is roughly equal to $(k+1)\nabla g(x^k)\nabla g(x^k)'$, where $g = (g_1, g_2, \ldots, g_m)$ is the original data set. It follows that for large k, the EKF iteration (29) can be written approximately as

$$x^{k+1} \approx x^k - \frac{1}{k+1} \big(\nabla g(x^k) \nabla g(x^k)'\big)^{-1} \nabla g(x^k) g(x^k), \tag{31}$$

that is, as an approximate Gauss–Newton iteration with diminishing stepsize. Thus, based on generic properties of gradient methods with diminishing stepsize (see, e.g., [Pol87]), we can expect convergence to stationary points of the least squares problem and a sublinear convergence rate.

When $\lambda < 1$, the matrix H_i^{-1} generated by the EKF recursion (22) will typically not diminish to zero, and $\{x^k\}$ may not converge to a stationary point of $\sum_{i=1}^{m} \lambda^{m-i} \|g_i(x)\|^2$. Furthermore, as the following example shows, the sequences $\{\psi_{km+i}\}$ produced by the EKF using eq. (21) may converge to different limits for different i.

Example 1. Consider the case where there are two data blocks, $g_1(x) = x - c_1$ and $g_2(x) = x - c_2$, where c_1 and c_2 are given scalars. Each cycle of the EKF consists of two steps. At the second step of the kth cycle, we minimize

$$\sum_{i=1}^{k} \left(\lambda^{2i-1}(x - c_1)^2 + \lambda^{2i-2}(x - c_2)^2\right),$$

which is equal to the following scalar multiple of $\lambda(x - c_1)^2 + (x - c_2)^2$:

$$(1 + \lambda^2 + \cdots + \lambda^{2k-2})\big(\lambda(x - c_1)^2 + (x - c_2)^2\big).$$

Thus, at the second step, we obtain the minimizer of $\lambda(x-c_1)^2+(x-c_2)^2$:

$$\psi_{2k}=\frac{\lambda c_1+c_2}{\lambda+1}.$$

At the first step of the kth cycle we minimize

$$(x-c_1)^2+\lambda\sum_{i=1}^{k-1}\left(\lambda^{2i-1}(x-c_1)^2+\lambda^{2i-2}(x-c_2)^2\right),$$

which is equal to the following scalar multiple of $(x-c_1)^2+\lambda(x-c_2)^2$:

$$(1+\lambda^2+\cdots+\lambda^{2k-4})\big((x-c_1)^2+\lambda(x-c_2)^2\big),$$

plus the diminishing term $\lambda^{2k-2}(x-c_1)^2$. Thus, at the first step, we obtain approximately (for large k) the minimizer of $(x-c_1)^2+\lambda(x-c_2)^2$:

$$\psi_{2k-1}\approx\frac{c_1+\lambda c_2}{1+\lambda}.$$

We see, therefore, that within each cycle there is an oscillation around the minimizer $(c_1+c_2)/2$ of $(x-c_1)^2+(x-c_2)^2$. The size of the oscillation diminishes as λ approaches 1.

The preceding example suggests that each sequence $\{\psi_{km+i}\}$, where $i=1,\ldots,m$, may converge to a stationary point of the function

$$f_i(x)=\sum_{j=1}^{m}\lambda^{m-j}\|g_{j+i}(x)\|^2,\qquad i=1,\ldots,m,$$

where we use the definition

$$g_j(x)=g_{j\bmod(m)+1}(x)\quad\text{if } j>m.$$

This is readily shown when the data blocks g_i are linear in view of the definition of ψ_{km+i} as the minimizer of

$$\sum_{j=1}^{km+i}\lambda^{km+i-j}\|g_j(x)\|^2,$$

which can also be written as

$$\sum_{j=1}^{i}\lambda^{km+i-j}\|g_j(x)\|^2+\big(1+\lambda^2+\cdots+\lambda^{(k-1)m}\big)f_i(x).$$

Since the leftmost summation above vanishes as $k\to\infty$, ψ_{km+i} minimizes $f_i(x)$ asymptotically. In the case of nonlinear data blocks, a related but more complex analysis of the cyclic convergence behavior described above is possible, but this analysis will not be attempted in this paper.

Generally, for a nonlinear least squares problem, the convergence rate tends to be faster when $\lambda<1$ than when $\lambda=1$, essentially because the implicit stepsize does not diminish to zero as in the case $\lambda=1$. For this reason, a hybrid method that uses a

different value of λ within each cycle may work best in practice. One may start with a relatively small λ to attain a fast initial rate of convergence, and then progressively increase λ toward 1 to attain high solution accuracy. The following proposition shows convergence for the case where λ tends to 1 at a sufficiently fast rate.

PROPOSITION 2. *Assume that $\nabla g_i(x)$ has full rank for all x and $i = 1, \ldots, m$, and that for some $L > 0$ we have*

$$\|\nabla g_i(x) g_i(x) - \nabla g_i(y) g_i(y)\| \leq L\|x - y\| \quad \forall\, x, y \in \Re^n,\ i = 1, \ldots, m. \tag{32}$$

Assume also that there is a constant $c > 0$ such that the scalar λ used in the updating formula (22) within the kth cycle, call it λ_k, satisfies

$$0 \leq 1 - \lambda_k^m \leq \frac{c}{k} \quad \forall\, k = 1, 2, \ldots. \tag{33}$$

Then if the EKF applied to the extended data set (28) generates a bounded sequence of vectors ψ_i, the sequence $\{f(x^k)\}$ converges and each of the limit points of $\{x^k\}$ is a stationary point of the least squares problem.

We develop the proof of Proposition 2 through a series of lemmas, all of which implicitly assume the conditions of Proposition 2.

LEMMA 2. *There exist positive scalars c_1 and c_2 such that for all k, the eigenvalues of the matrices H_{km} lie within the interval $[c_1 k, c_2 k]$.*

Proof. We have, using the update formula (22), that

$$H_{(k+1)m} = \lambda_{k+1}^m H_{km} + \sum_{i=1}^{m} \lambda_{k+1}^{m-i} \nabla g_i(\psi_{km+i-1}) \nabla g_i(\psi_{km+i-1})'. \tag{34}$$

Let X be a compact set containing all vectors ψ_i generated by the algorithm, and let B and b be an upper bound and a lower bound, respectively, for the eigenvalues of $\nabla g_i(x) \nabla g_i(x)'$ as x ranges over X. From eq. (34), it is seen by induction that all eigenvalues of H_{km} are less than or equal to $c_2 k$ with $c_2 = mB$. Furthermore, if v_k is the smallest eigenvalue of H_{km}, then from eqs. (33) and (34) it is seen by induction that

$$v_{k+1} \geq \left(1 - \frac{c}{k+1}\right) v_k + \left(1 - \frac{c}{k+1}\right) mb \quad \forall\, k \geq 1. \tag{35}$$

Using this relation, we will prove that $v_k \geq k\beta$ for a sufficiently small but positive value of β. Indeed, let $\bar{k}$ be the minimal positive integer k such that $c/k < 1$, and let β be any positive scalar such that

$$\beta \leq \frac{(\bar{k} + 1 - c)mb}{\bar{k} + 1 + \bar{k}c}.$$

From eq. (35), it is seen that if $v_{\bar{k}} \geq \beta \bar{k}$, then

$$\begin{aligned} v_{\bar{k}+1} &\geq \left(1 - \frac{c}{\bar{k}+1}\right) \beta \bar{k} + \left(1 - \frac{c}{\bar{k}+1}\right) mb \\ &= \beta(\bar{k} + 1) + \frac{(\bar{k} + 1 - c)mb}{\bar{k} + 1} - \frac{(\bar{k} + 1 + \bar{k}c)\beta}{\bar{k} + 1} \\ &\geq \beta(\bar{k} + 1). \end{aligned}$$

Similarly, it is shown that $v_k \geq \beta k$ for all $k \geq \bar{k}$. Thus, by taking β equal to the scalar c_1 given below,

$$c_1 = \min\left\{\frac{(\bar{k}+1-c)mb}{\bar{k}+1+\bar{k}c}, \min_{k=1,\ldots,\bar{k}} \frac{v_k}{k}\right\},$$

we see that $v_k \geq c_1 k$ for all $k \geq 1$. □

We will use the notation

$$f(x) = \frac{1}{2}\sum_{i=1}^{m} \|g_i(x)\|^2, \tag{36}$$

from which we have

$$\nabla f(x) = \sum_{i=1}^{m} \nabla g_i(x) g_i(x). \tag{37}$$

The next lemma shows that the vector that is multiplied by $H_{(k+1)m}^{-1}$ to obtain the direction used by the EKF (cf. eqs. (13) and (29)) differs from the gradient $\nabla f(x^k)$ by a relatively small amount.

LEMMA 3. *Let*

$$e^k = \nabla f(x^k) - \sum_{i=1}^{m} \lambda_{k+1}^{m-i} \nabla g_i(\psi_{km+i-1}) g_i(\psi_{km+i-1}). \tag{38}$$

Then there exists a scalar γ such that for all k

$$\|e^k\| \leq \frac{\gamma}{k+1}. \tag{39}$$

Proof. We have, using eqs. (37) and (38),

$$\begin{aligned} e^k = &\sum_{i=1}^{m} (1-\lambda_{k+1}^{m-i}) \nabla g_i(x^k) g_i(x^k) \\ &+ \sum_{i=1}^{m} \lambda_{k+1}^{m-i} \left(\nabla g_i(x^k) g_i(x^k) - \nabla g_i(\psi_{km+i-1}) g_i(\psi_{km+i-1})\right), \end{aligned}$$

so from assumptions (32) and (33) we obtain

$$\begin{aligned} \|e^k\| &\leq (1-\lambda_{k+1}^{m}) \sum_{i=1}^{m} \|\nabla g_i(x^k) g_i(x^k)\| + L \sum_{i=1}^{m} \|x^k - \psi_{km+i-1}\| \\ &\leq \frac{cM}{k+1} + L \sum_{i=1}^{m} \|x^k - \psi_{km+i-1}\|, \end{aligned} \tag{40}$$

where M is a bound for $\sum_{i=1}^{m} \|\nabla g_i(x^k) g_i(x^k)\|$. We also have, using eq. (21), for all k and $i \geq 2$,

$$\|x^k - \psi_{km+i-1}\| \leq \|H_{km}^{-1}\| \sum_{j=1}^{i-1} \|\nabla g_i(\psi_{km+j}) g_i(\psi_{km+j})\|.$$

Using the boundedness of ψ_i and Lemma 2, we see that for all i and some $\delta > 0$ we have for $k \geq 1$

$$\|x^k - \psi_{km+i-1}\| \leq \frac{\delta}{k} \leq \frac{2\delta}{k+1}.$$

Combining this relation with eq. (40), we obtain the desired relation (39). □

Assumption (32) together with eq. (37) implies that

$$\|\nabla f(x) - \nabla f(y)\| \leq mL\|x - y\| \quad \forall\, x, y \in \Re^n. \tag{41}$$

The next lemma is a well-known consequence of this relation. We include the proof for completeness.

LEMMA 4. *For all x and y, there holds*

$$f(x+y) \leq f(x) + y'\nabla f(x) + \frac{mL}{2}\|y\|^2. \tag{42}$$

Proof. Let t be a scalar parameter and let $F(t) = f(x+ty)$. Using eq. (41), we have

$$\begin{aligned} f(x+y) - f(x) = F(1) - F(0) &= \int_0^1 \frac{dF}{dt}(t)\,dt = \int_0^1 y'\nabla f(x+ty)\,dt \\ &\leq \int_0^1 y'\nabla f(x)\,dt + \left| \int_0^1 y'\big(\nabla f(x+ty) - \nabla f(x)\big)\,dt \right| \\ &\leq \int_0^1 y'\nabla f(x)\,dt + \int_0^1 \|y\| \cdot \|\nabla f(x+ty) - \nabla f(x)\| dt \\ &\leq y'\nabla f(x) + \|y\| \int_0^1 Lt\|y\|\,dt \\ &= y'\nabla f(x) + \frac{mL}{2}\|y\|^2. \qquad \square \end{aligned}$$

We are now ready to prove Proposition 2.

Proof of Proposition 2. We have, using the Kalman filter recursion (13) and the definition (38) of e^k,

$$x^{k+1} = x^k - H_{(k+1)m}^{-1}\left(\sum_{i=1}^m \lambda_{k+1}^{m-i}\nabla g_i(\psi_{km+i-1})g_i(\psi_{km+i-1})\right) = x^k + d^k,$$

where

$$d^k = -H_{(k+1)m}^{-1}\big(\nabla f(x^k) - e^k\big). \tag{43}$$

Using Lemmas 2 and 3 and the fact that

$$\frac{1}{c_2(k+1)} \leq \|H_{(k+1)m}^{-1}\| \leq \frac{1}{c_1(k+1)},$$

which follows from Lemma 2, it is seen that

$$\begin{aligned} d^{k\prime}\nabla f(x^k) &= -\nabla f(x^k)' H_{(k+1)m}^{-1}\nabla f(x^k) + e^{k\prime} H_{(k+1)m}^{-1}\nabla f(x^k) \\ &\leq -\frac{\|\nabla f(x^k)\|^2}{c_2(k+1)} + \frac{\|e^k\|\,\|\nabla f(x^k)\|}{c_1(k+1)} \\ &\leq -\frac{\|\nabla f(x^k)\|^2}{c_2(k+1)} + O\left(\frac{1}{(k+1)^2}\right)\|\nabla f(x^k)\| \end{aligned}$$

and

(44)

$$\begin{aligned}\|d^k\|^2 &\le \|H^{-1}_{(k+1)m}\|^2\left(\|\nabla f(x^k)\| + \|e^k\|\right)^2 \\ &= O\left(\frac{1}{(k+1)^2}\right)\left(\|\nabla f(x^k)\| + O\left(\frac{1}{k+1}\right)\right)^2 \\ &= O\left(\frac{1}{(k+1)^2}\right)\|\nabla f(x^k)\|^2 + O\left(\frac{1}{(k+1)^3}\right)\|\nabla f(x^k)\| + O\left(\frac{1}{(k+1)^4}\right).\end{aligned}$$

Using these relations in eq. (42), we obtain

$$\begin{aligned}f(x^{k+1}) &\le f(x^k) + d^{k\prime}\nabla f(x^k) + \frac{mL}{2}\|d^k\|^2 \\ &\le f(x^k) - \left(\frac{1}{c_2(k+1)} + O\left(\frac{1}{(k+1)^2}\right)\right)\|\nabla f(x^k)\|^2 \\ &\quad + O\left(\frac{1}{(k+1)^2}\right)\|\nabla f(x^k)\| + O\left(\frac{1}{(k+1)^4}\right).\end{aligned}$$

Thus, since $\|\nabla f(x^k)\|$ is bounded, there exist constants $\beta_1 > 0$ and $\beta_2 > 0$ and a positive integer $\bar{k}$ such that

$$f(x^{k+1}) \le f(x^k) - \frac{\beta_1}{k}\|\nabla f(x^k)\|^2 + \frac{\beta_2}{k^2} \quad \forall\, k \ge \bar{k}. \tag{45}$$

It is well known that if $\{u^k\}$ and $\{d^k\}$ are nonnegative sequences such that $u^{k+1} \le u^k + \delta^k$ for all k and $\sum_{k=1}^{\infty}\delta^k < \infty$, then $\{u^k\}$ converges; this is a special case of the supermartingale convergence theorem (see, e.g., [Pol87, p. 49] or [BeT89, p. 677]). Since $f(x) \ge 0$ for all x, it follows from eq. (45) that $\{f(x^k)\}$ converges.

From eq. (45) we have for all $k \ge \bar{k}$

$$f(x^{k+1}) \le f(x^{\bar{k}}) - \sum_{i=\bar{k}}^{k}\frac{\beta_1}{i}\|\nabla f(x^i)\|^2 + \sum_{i=\bar{k}}^{k}\frac{\beta_2}{i^2}. \tag{46}$$

Since $\sum_{i=\bar{k}}^{\infty} 1/i = \infty$ and $\sum_{i=\bar{k}}^{\infty} 1/i^2 < \infty$, we see also that there cannot exist an $\epsilon > 0$ such that $\|\nabla f(x^k)\|^2 > \epsilon\ \forall k \ge \bar{k}$. Therefore, we must have $\liminf_{k\to\infty}\|\nabla f(x^k)\| = 0$.

We will now show that $\|\nabla f(x^k)\| \to 0$. Indeed, assume the contrary, that is, there exists an $\epsilon > 0$ such that $\|\nabla f(x^k)\| > \epsilon$ for all k in an infinite subset of integers $\mathcal{K}$. For each $k \in \mathcal{K}$, let $i(k)$ be the first index i such that $i > k$ and $\|\nabla f(x^i)\| < \epsilon/2$, so that

(47)

$$\frac{\epsilon}{2} \le \|\nabla f(x^k)\| - \|\nabla f(x^{i(k)})\| \le \|\nabla f(x^k) - \nabla f(x^{i(k)})\| \le L\|x^k - x^{i(k)}\| \le L\sum_{i=k}^{i(k)-1}\|d^i\|.$$

Since from eq. (44) we have $\|d^k\| = O(1/k)$, eq. (47) implies that for some constant $B_1 > 0$,

$$\frac{\epsilon}{2} \le B_1\sum_{i=k}^{i(k)-1}\frac{1}{i} \quad \forall\, k \in \mathcal{K}.$$

From eq. (46) we see that

$$f\left(x^{i(k)}\right) \leq f(x^k) - \beta_1 \left(\frac{\epsilon}{2}\right)^2 \sum_{i=k}^{i(k)-1} \frac{1}{i} + \sum_{i=k}^{i(k)-1} \frac{\beta_2}{i^2} \quad \forall\, k \in \mathcal{K}.$$

Since $\{f(x^k)\}$ converges and $\lim_{k\to\infty} \sum_{i=k}^{i(k)-1} \beta_2/i^2 = 0$, it follows that

$$\lim_{k\to\infty,\, k\in\mathcal{K}} \sum_{i=k}^{i(k)-1} \frac{1}{i} = 0,$$

contradicting the earlier conclusion that $\frac{\epsilon}{2} \leq B_1 \sum_{i=k}^{i(k)-1} 1/i$ for all $k \in \mathcal{K}$. Therefore, $\|\nabla f(x^k)\| \to 0$, and it follows that every limit point of $\{x^k\}$ is a stationary point of f. □

Note that the proof of Lemma 2 carries through even if the initial matrix H_0 is any positive-definite matrix rather than $H_0 = 0$. As a result, Proposition 2 also holds when H_0 is some positive-definite matrix, in which case it is unnecessary to assume that the matrices $\nabla g_i(x)\nabla g_i(x)'$ have full rank, as long as enough alternative assumptions are imposed to guarantee the validity of the crucial Lemma 2. More generally, our method of proof shows that the convergence characteristics of the method are maintained by any scheme that varies λ and/or H in a way that Lemma 2 holds.

The boundedness assumption on the sequence of vectors ψ_i is a substantial weakness of Proposition 2. It is not easy to remove this assumption because the algorithm does not have an explicit stepsize mechanism to control the magnitude of the initial iterates. On the other hand, one can employ the device of projecting the iterates ψ_i on a compact set that is known to contain an optimal solution and use a projection version of the EKF of the type introduced in [Ber82a] and [Ber82b, §1.5]. Projecting the iterates on a compact set is a well-known approach to enhance the theoretical convergence properties of the EKF (see [Lju79]).

In practice, the method seems to converge considerably faster if λ is initially less than 1 and is progressively increased toward 1 in a judicious manner. On the other hand, an implicit diminishing stepsize as indicated by Lemma 2 is essential for the convergence of the method, and such a stepsize induces a generically sublinear convergence rate. This characteristic is shared with the backpropagation method where, to achieve a linear convergence rate, it is essential to use a stepsize that is bounded away from zero, but when such a stepsize is used, the method tends to converge to oscillate [Luo91].

We finally note that as a result of its sublinear convergence rate, the EKF will typically become ultimately slower than the Gauss–Newton method, even though it may be much faster in the initial iterations. The ultimate convergence rate of both the EKF and the backpropagation method may be improved by modifications that gradually change the incremental character of these methods and ultimately make them identical to the pure Gauss–Newton method and the steepest-descent method, respectively. In particular, as convergence is approached, one may adaptively combine ever larger groups of data blocks together into single data blocks. When all data blocks are combined into a single block, the EKF and the backpropagation method will become equivalent to the pure Gauss–Newton method and the steepest-descent method, respectively.

4. Conclusions. In this paper we have considered EKF algorithms for least squares problems, which consist of repeated cycles through the data set. The computational significance of these algorithms is well known from the control theory literature and has been documented by Davidon [Dav76] in the optimization literature. The algorithms are incremental in nature and bear a similar relation to the Gauss–Newton method as incremental backpropagation methods bear to the steepest-descent method. Because of their incremental character, EKF methods seem particularly well suited for neural network training problems. However, there has been no convergence analysis of these methods for the case of a finite data set, and the present paper fills this gap.

REFERENCES

[AWT69] M. ATHANS, R. P. WISHNER, AND A. BERTOLINI, *Suboptimal state estimation for continuous time nonlinear systems from discrete noisy measurements*, IEEE Trans. Automat. Control, AC-13 (1968), pp. 504–514.

[AnM79] B. D. O. ANDERSON AND J. B. MOORE, *Optimal Filtering*, Prentice–Hall, Englewood Cliffs, NJ, 1979.

[BeT89] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[BeT96] ———, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[Bel94] B. M. BELL, *The iterated Kalman smoother as a Gauss–Newton method*, SIAM J. Optim., 4 (1994), p. 626.

[Ber82a] D. P. BERTSEKAS, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[Ber82b] ———, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[Ber95] ———, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[Dav76] W. C. DAVIDON, *New least squares algorithms*, J. Optim. Theory Appl., 18 (1976), pp. 187–197.

[Gai93] A. A. GAIVORONSKI, *Convergence Analysis of Parallel Backpropagation Algorithm for Neural Networks*, Symposium on Parallel Optimization 3, Madison, WI, July 7-9, 1993.

[Gri93] L. GRIPPO, *A Class of Unconstrained Minimization Methods for Neural Network Training*, Symposium on Parallel Optimization 3, Madison, WI, July 7–9, 1993.

[Jaz70] A. H. JAZWINSKI, *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.

[KuC78] H. J. KUSHNER AND D. S. CLARK, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York, 1978.

[Lju77] L. LJUNG, *Analysis of recursive stochastic algorithms*, IEEE Trans. Automat. Control, AC-22 (1977), pp. 551–575.

[Lju79] ———, *Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems*, IEEE Trans. on Automat. Control, AC-24 (1979), pp. 36–50.

[LuT93] Z. Q. LUO AND P. TSENG, *Analysis of an approximate gradient projection method with applications to the backpropagation algorithm*, Optim. Meth. Software, 4 (1994), pp. 85–101.

[Luo91] Z. Q. LUO, *On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks*, Neural Comput., 3 (1991), pp. 226–245.

[MaS94] O. L. MANGASARIAN AND M. V. SOLODOV, *Serial and parallel backpropagation convergence via nonmonotone perturbed minimization*, Optim. Methods Software, 4 (1994), pp. 103–116.

[Man93] O. L. MANGASARIAN, *Mathematical programming in neural networks*, ORSA J. Comput., 5 (1993), pp. 349–360.

[Meh71] R. K. MEHRA, *A comparison of several nonlinear filters for reentry vehicle tracking*, IEEE Trans. Automat. Control, AC-16 (1971), pp. 307–319.

[Pap82] T. N. PAPPAS, *Solution of Nonlinear Equations by Davidon's Least Squares Method*, M.S. thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1982.

[PoT73] B. T. POLJAK AND Y. Z. TSYPKIN, *Pseudogradient adaptation and training algorithms*, Automation Remote Control, 12 (1973), pp. 45–68.

[Pol87] B. T. POLJAK, *Introduction to Optimization*, Optimization Software Inc., NY, 1987.

[RRK92] D. W. RUCK, S. K. ROGERS, M. KABRISKY, P. S. MAYBECK, AND M. E. OXLEY, *Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons*, IEEE Trans. Pattern Analysis and Machine Intelligence, 14 (1992), pp. 686–691.

[TBA86] J. N. TSITSIKLIS, D. P. BERTSEKAS, AND M. ATHANS, *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 803–812.

[THS77] R. R. TENNEY, R. S. HEBBERT, AND N. R. SANDELL, JR., *Tracking filter for maneuvering sources*, IEEE Trans. Automat. Control, AC-22 (1977), pp. 246–251.

[Tsi84] J. N. TSITSIKLIS, *Problems in Decentralized Decision Making and Computation*, Ph.D. dissertation, Dept. of Electrical Enginering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[Urs80] B. URSIN, *Asymptotic convergence properties of the extended Kalman filter using filtered state estimates*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 1207–1211.

[Whi89a] H. WHITE, *Some asymptotic results for learning in single hidden-layer feedforward network models*, J. Amer. Statist. Assoc., 84 (1989), pp. 1003–1013.

[Whi89b] ———, *Learning in artificial neural networks: A statistical perspective*, Neural Comput., 1 (1989), pp. 425–464.

[WaT90] K. WATANABE AND S. G. TZAFESTAS, *Learning algorithms for neural networks with the Kalman filters*, J. Intelligent Robotic Systems, 3 (1990), pp. 305–319.

[WeM80] H. WEISS AND J. B. MOORE, *Improved extended Kalman filter design for passive tracking*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 807–811.

[WiH60] B. WIDROW AND M. E. HOFF, *Adaptive Switching Circuits*, Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, part 4, 1960, pp. 96–104.

[WiS85] B. WIDROW AND S. D. STEARNS, *Adaptive Signal Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1985.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 823–837, August 1996
© 1996 Society for Industrial and Applied Mathematics
016

NETWORK DESIGN USING CUT INEQUALITIES*

FRANCISCO BARAHONA†

Abstract. We study the network loading problem, with and without bifurcations. We use a relaxation based on the cut condition for multicommodity flows. We use a solution of the bifurcated case to derive a solution to the nonbifurcated problem. A standard procedure is to aggregate the problem into a *backbone network*. We applied this method to backbone networks coming from practical instances; we obtained feasible solutions and bounds for the gap from optimality.

Key words. network loading problem, cut condition

AMS subject classifications. 90C10, 90C27

1. Introduction.

The *network loading problem* plays a central role in the design of telecommunications networks. It consists of choosing a set of links of minimum cost that connect a set of cities. The links have a finite capacity, and they should allow the existence of a multicommodity flow. If for each commodity the flow is required to follow a single path, we called it the *nonbifurcated network loading problem*, and denote it by NBP. If bifurcations are allowed, the problem is denoted by BP. In this paper we consider the undirected version, i.e., the links and the requirements are undirected. This is a model for voice networks. Most of the procedures described here extend to the directed case.

The input for this problem consists of an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}$, and a set of requirements $R = \{(s_k, t_k, d_k) \mid 1 \leq k \leq p\}$. A requirement is given by two nodes s_k, t_k in V and a demand $d_k \in \mathbb{R}_+$. There is a link-capacity u for every edge $e \in E$. For each edge $e = \{i, j\}$, we have a *link* variable x_e, taking integer values that could be greater than 1. We also have the *flow* variables f_{ij}^k and f_{ji}^k for $k = 1, \ldots, p$. The formulation of BP is

$$
(1) \quad \begin{cases}
\text{minimize } \displaystyle\sum_{e \in E} c_e x_e & \\
\text{subject to} & \\
\displaystyle\sum_j f_{ij}^k - \sum_j f_{ji}^k = \begin{cases} 1 & \text{if } i = s_k, \\ 0 & \text{if } i \neq s_k,\ i \neq t_k, \\ -1 & \text{if } i = t_k, \end{cases} & \\
\text{for } k = 1, \ldots, p, & \\
\displaystyle\sum_k d_k f_{ij}^k + \sum_k d_k f_{ji}^k \leq x_e u \quad \text{for every edge } e = \{i, j\}, & \\
f \geq 0, & \\
x \geq 0, \text{ integer.} &
\end{cases}
$$

*Received by the editors December 22, 1994; accepted for publication March 16, 1995.

†IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598 (barahon @watson.ibm.com).

This formulation can be strengthened by adding

$$f_{ij}^k + f_{ji}^k \leq x_e \quad \text{for every edge } e = \{i,j\},\ k = 1,\ldots,p. \tag{2}$$

To formulate NBP we replace $f \geq 0$ by

$$f_{ij}^k \in \{0,1\} \quad \text{for all } i,j,k.$$

This problem is NP-hard; it contains the Steiner tree problem as a special case.

The bifurcated problem has been studied in [20], [21], and [6]. The nonbifurcated case has been studied in [13]. The instances solved in the references above have different demand patterns and different graph densities, so it is not possible to make comparisons. However, one can say that large scale problems have not yet been solved. Also, several heuristics have been used; see [19], [18] and their references. These heuristics are fast, but do not provide information about the gap from optimality.

One difficulty of this type of problem comes from the multicommodity flow aspect. This has been noticed for other related network design problems in [7]. They describe an 8-node problem that has not been solved to optimality, despite the efforts of several groups. Just solving the linear program given by the multicommodity flow formulation (1) takes considerable computing time. The situation becomes worse if one resorts to branch and bound. For this reason we decided to work with a relaxation based on the *cut condition* for multicommodity flows. The main advantage is that we eliminate the flow variables. One complication is that testing the cut condition reduces to the *max cut problem* that is NP-hard. Fortunately, in all the cases we tried the max cut instances were relatively easy to solve. Using branch and bound with this relaxation is also much faster.

The nonbifurcated case has the extra difficulty of involving integer multicommodity flows. A common practice in network design is to work with a *backbone network*; see [18]. In a backbone network several nodes are identified into one supernode. Since the requirements are aggregated, the backbone problem allows bifurcations. From a solution of the backbone problem we were able to derive a solution to the nonbifurcated problem, with a few percent increase of the objective value.

We used this methodology with instances arising in practice. After aggregating into the backbone network, we obtained feasible solutions and bounds for the gap from optimality.

In §2 we discuss the choice of the backbone network. In §3 we present the linear programming relaxation. Sections 4 and 5 are devoted to BP and NBP, respectively. Our experiments are described in §6.

2. Choosing a backbone network. A usual procedure in network design is to first classify the nodes into *local* and *backbone*. Local nodes send their traffic through backbone nodes. Usually the backbone nodes are chosen by the importance of the cities and the amount of traffic that they originate. Also, one might want to cover different physical locations. Clustering algorithms have been proposed in [8] to make this choice.

We found useful the information given by the Gomory–Hu tree of the demand graph; see [14]. Hu [16] proved that the GH tree solves the following problem.

COMMUNICATION TREE. *Given a set of requirements between nodes and a spanning tree to route the flow, its cost is the sum of the amounts of flow going through each edge of the tree. We want to find a minimum cost tree.*

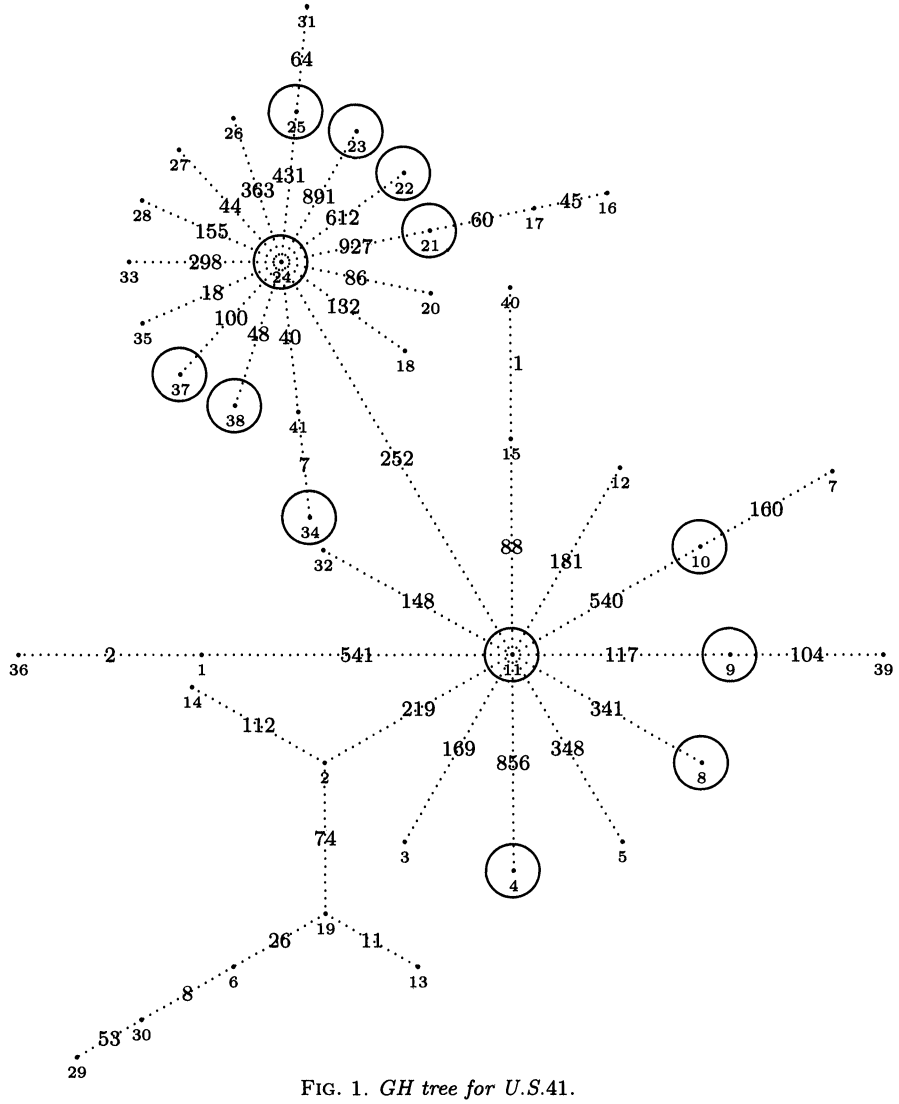


FIG. 1. *GH tree for U.S.*41.

In a solution to this problem, pairs of nodes having a high demand between them tend to be closer. So this tree gives some information about the demand pattern. For additional work on the communication tree problem, see [23]. In Figure 1 we show the GH tree associated with problem U.S.41. Nodes 11 and 24 are the two main centers. Nodes marked by a circle have been chosen as backbone nodes. Figure 2 shows the physical locations. We describe this problem in §6.

3. A relaxation based on cut inequalities. As a first approximation we are going to solve a linear program involving only the link variables. Because of its smaller dimensions it is much easier than the LP defined by (1) and (2).

A well-known necessary condition for the existence of a multicommodity flow is the *cut condition*, which can be stated as follows. For any set of nodes $S \subset V$, let $\delta(S)$ be the set of edges with one endnode in S and the other in $V \backslash S$. The set $\delta(S)$ is called a *cut*. Let $\gamma(S)$ be the set of requirements with one endnode in S and the other

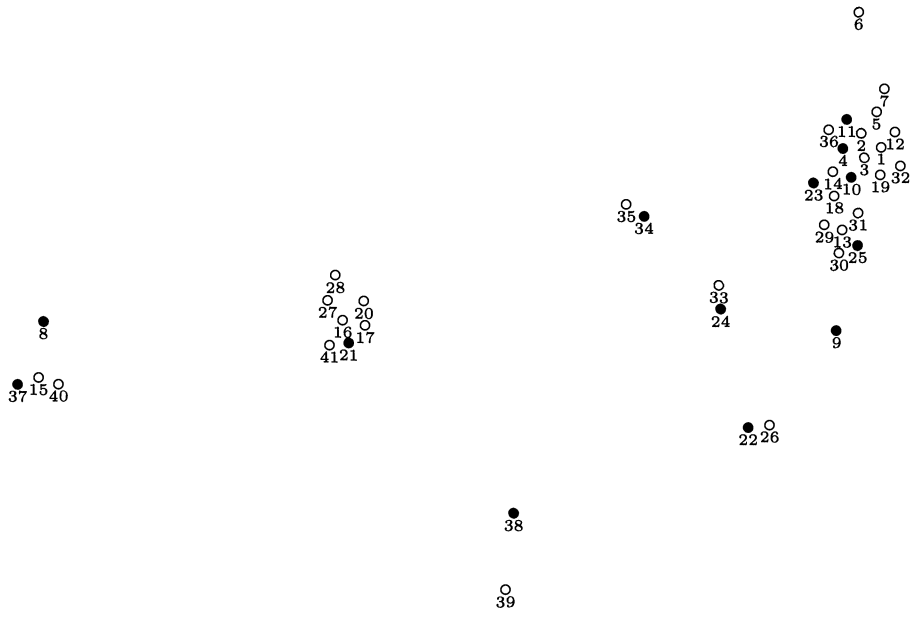


FIG. 2. *Locations for U.S.41. Backbone nodes are marked by a bullet.*

in $V \backslash S$. The cut condition is

$$\sum_{e \in \delta(S)} u\, x_e \geq \sum_{k \in \gamma(S)} d_k. \tag{3}$$

This says that the capacity of this cut should be at least the sum of the requirements separated by this cut. Since the variables x take integer values, (3) can be strengthened to

$$\sum_{e \in \delta(S)} x_e \geq \left\lceil \sum_{k \in \gamma(S)} d_k / u \right\rceil. \tag{4}$$

Under some simple assumptions, inequalities (4) define facets of the polyhedron (1); see [20].

Consider a partition $V_1, \ldots, V_p$ of V. Denote by $\delta(V_1, \ldots, V_p)$ the set of edges whose endnodes are in different members of the partition. In most cases the network should be connected, i.e., it should contain a spanning tree. The spanning tree polyhedron is defined by nonnegativity and the *multicut* inequalities below; see [22], [25], [10].

$$\sum_{e \in \delta(V_1, \ldots, V_p)} x_e \geq p - 1 \text{ for every partition of } V. \tag{5}$$

In many cases, when the capacities are large, the solution is a spanning tree. Therefore in these cases, inequalities (5) define facets of the polyhedron (1).

For reliability reasons, sometimes the network is required to be 2-*edge-connected.* This means that every cut should contain at least two links. This can be modeled by the system

$$\sum_{e \in \delta(S)} x_e \geq 2 \text{ for every set } S \subset V. \tag{6}$$

One could also ask for a 2-*node-connected* network. This means that if a node is deleted, the remainder should still be connected. This is equivalent to saying that the remainder should contain a spanning tree, so this can be modeled by

$$\sum_{e\in\delta(V_1,\ldots,V_p)} x_e \geq p-1 \quad \text{for every partition of } V\backslash v, \text{ for every node } v. \tag{7}$$

This last set of constraints has been used in [15] and [5] to handle node-connectivity.

Inequalities (2) are implied by (1) and (3). So one could conceive solving the problem defined by (1), (4), and (5). In our experience, this can take too much time. Actually solving (1) as a linear program could also take a large amount of time. For this reason, we first obtain a lower bound by solving

$$\begin{cases} \text{minimize } \displaystyle\sum_{e\in E} c_e x_e \\ \text{subject to} \\ \displaystyle\sum_{e\in\delta(S)} x_e \geq \left\lceil \sum_{k\in\gamma(S)} d_k/u \right\rceil \quad \text{for every set } S\subset V, \\ \displaystyle\sum_{e\in\delta(V_1,\ldots,V_p)} x_e \geq p-1 \text{ for every partition of } V, \\ x \geq 0. \end{cases} \tag{8}$$

This problem has many fewer variables than (1), but it has an exponential number of inequalities. We tackle it with a *cutting plane approach* as follows. We first solve it with only a small subset of its inequalities. Given a solution of this, we check for new inequalities that are violated. If there are any, we add them and solve this linear program again. If there is no violated constraint we stop. Now we must show how to find violated inequalities, if there are any. This is called the *separation problem.*

Given a vector $\bar{x}$, we can find a constraint (3) that is violated by first building a weighted graph H. For every edge e in G with $\bar{x}_e > 0$, we add the edge e to H with weight $-\bar{x}_e u$. And for every requirement (s_k, t_k, d_k), we add an edge between s_k and t_k, with weight d_k. For a cut in H, its weight is the sum of the weights of the edges in it. A cut of positive weight gives a violated constraint. To find it we solve the *max cut problem* in H, as described in Appendix A. The max cut problem is NP-hard. However, in the cases that we treated, the graph H was sparse and with few nodes; thus it was relatively easy to solve.

The separation problem for (5) reduces to a sequence of $|V|$ minimum cut problems, as described in [1]. We sketch this procedure in Appendix B.

If 2-edge-connectivity is required, we add (6) to (8). The separation problem in this case is a sequence of $|V|$ minimum cut problems.

If 2-node-connectivity is required, we add (7). For this, the separation problem can be solved with a sequence of $|V|^2$ minimum cut problems with the procedure of Appendix B.

Other families of inequalities could be included here, such as 3-cut inequalities; see [20], [6]. We considered only those for which we had a way to handle the separation problem.

4. Finding a feasible solution of BP. Usually when we solve (8), we obtain a fractional solution. At this point we start a branch and bound procedure. This time, in the nodes of the branch and bound tree, we only use a max cut heuristic to generate violated inequalities (4).

Having an integer vector $\bar{x}$ that satisfies the cut condition does not guarantee the existence of a multicommodity flow. We use $\bar{x}$ as a set of lower bounds for the variables x in (1), and we solve this problem with branch and bound. Notice that we deal with (1) only after we know the vector $\bar{x}$ that gives an approximation of the final values of x. Since most of the variables x remain fixed to their lower bounds $\bar{x}$, this part takes a relatively small effort.

At this point we have a feasible solution of the bifurcated problem. As we shall see in §6, the value of this was close to the lower bound given by (8).

In the next section we use this to derive a solution of the nonbifurcated problem.

5. The nonbifurcated case. Given a solution of BP, we use it as a guide to derive a feasible solution of NBP. One basic approach is to keep the requirements that have not been bifurcated and route the others using shortest paths. Since this is a fast procedure, one can introduce random perturbations of the costs to produce many different solutions, and keep the best. The requirements are treated sequentially, so one should also treat them in many different orders. One of these orders can be "largest first." The procedure consists of the following three steps.

- Step 1. Start with the partial solution given by the requirements that have not been bifurcated in the solution of BP; denote this by $\mathbb{F}$. Denote by $\mathbb{L}$ the set of links used in BP.
- Step 2. To route the requirements not in $\mathbb{F}$, we use shortest paths in a graph as follows.
  - For each link in $\mathbb{L}$ add an edge with upper bound equal to the residual capacity of the link. If the link is being used in $\mathbb{F}$ set its cost equal to 1; otherwise set it to the cost of the link multiplied by a random number in $[0, 0.01]$.
  - For each edge of the original problem, add an edge with upper bound equal to the capacity of the link. Its cost is equal to the original cost of the link multiplied by a random number α. If the link is in $\mathbb{L}$, then $\alpha \in [0, 0.1]$; otherwise $\alpha \in [0, 1]$. Notice that this second set of edges will tend to be more expensive than the edges defined above.
  - Choose a requirement k not in $\mathbb{F}$. Find a shortest path from s_k to t_k, using edges with upper bound at least the demand d_k. Add this requirement to $\mathbb{F}$. Reduce the weights of the edges in the shortest path to the value 1. Also update the upper bounds of these edges. If any of these edges are empty, add a new empty edge. Its weight should be the original cost multiplied by a random number in $[0, 1]$. Continue until all requirements have been routed.
- Step 3. Now we revise the solution, deleting less-used links. We pick a random number $\beta \in [0, 80]$. We delete all links whose utilization is less than $\beta\%$. We also delete all requirements using these links. This defines a new set $\mathbb{L}$ and new partial routing $\mathbb{F}$. We use them to perform Step 2 a second time.

The links in $\mathbb{L}$ are considered good candidates to be used. At any stage of the routing, we have three types of weights. The links that are already being used have weight 1, the links in $\mathbb{L}$ that are not being used are more expensive, and the links not used and not in $\mathbb{L}$ are even more expensive. After we find a first routing we revise the solution by deleting links with less utilization and also deleting the routes that use

them. This defines a new set of good candidates $\mathbb{L}$ and a new partial routing $\mathbb{F}$. We start from them to find a second routing. We repeat these three steps many times and keep the best solution.

We repeated this procedure between 200 and 500 times. In the worst case we tried, the solution we obtained was 6% more expensive than the solution of BP.

6. Experiments. Now we describe the cases we tackled. Each time, we first choose a backbone network as in §2. For this backbone network, we find a lower bound solving (8), as described in §3. Then we find a solution of BP for the backbone network as in §4. This is expanded to a solution of BP for the entire network. Starting from it, we apply the procedure of §5 to find a solution of NBP for the entire network.

Given a design, for each link we compute its unused capacity multiplied by the cost of the link. The sum of all these values is a parameter U_N, which we call the *cost of the unused capacity.* In the same way we compute this value for the local networks and we denote it by U_L. We have used these parameters to measure the quality of the design. Sometimes, when we had a local network with high unused capacity, we added one of these nodes to the backbone network. This would give a closer connection for the local nodes.

We have done all calculations on an RS6000-340 workstation. We used OSL [11] and MINTO [24] as optimization software.

6.1. A 41 cities problem in the U.S. The first problem we tackled consists of 41 cities in the U.S. and has 646 requirements. Links between all pairs of cities could be used. We denote it by U.S.41.

After choosing 13 backbone nodes, we had a cost of 120000 \$/month associated with the local networks. The cost U_L was 52000 \$/month. We solved (8) in seven minutes, and the LP value was 420000. Then we used branch and bound to derive an integer vector satisfying the cut condition. Its value was 438000 and this took six minutes. We used this as lower bounds for the variables x and solved (1) using branch and bound. We obtained a solution of value 443000 in less than one minute. At this point we had a solution of BP for the backbone network, and the gap from optimality was 6%.

Then we expanded the bifurcated solution to a solution of BP for the entire network, its value was 563000, and we had a lower bound of 540000. At this point we applied the procedure of §5 to obtain a routing for NBP. We ran 500 iterations in eight minutes. The value we obtained was 575000, and U_N was 79000. We plot this solution in Figure 3, where backbone nodes are marked with bullets.

We spent approximately two hours working on this problem, including the time to choose the backbone network. We summarize these results in Table 1. From the values of U_N, U_L, and the lower bound, it seems that now one should optimize the local networks.

TABLE 1

	Loc net cost	U_L	Lower bound	BP	NBP	U_N
Value	120K	52K	540K	563K	575K	79K
Comp time	–	–	7 min	6 min	8 min	–

6.2. A 47 cities problem in the U.S. The second instance had 47 cities in the U.S., 1081 requirements, and links between all pairs of cities could be used.

It had five cities with a large amount of traffic, so these were the first candidates

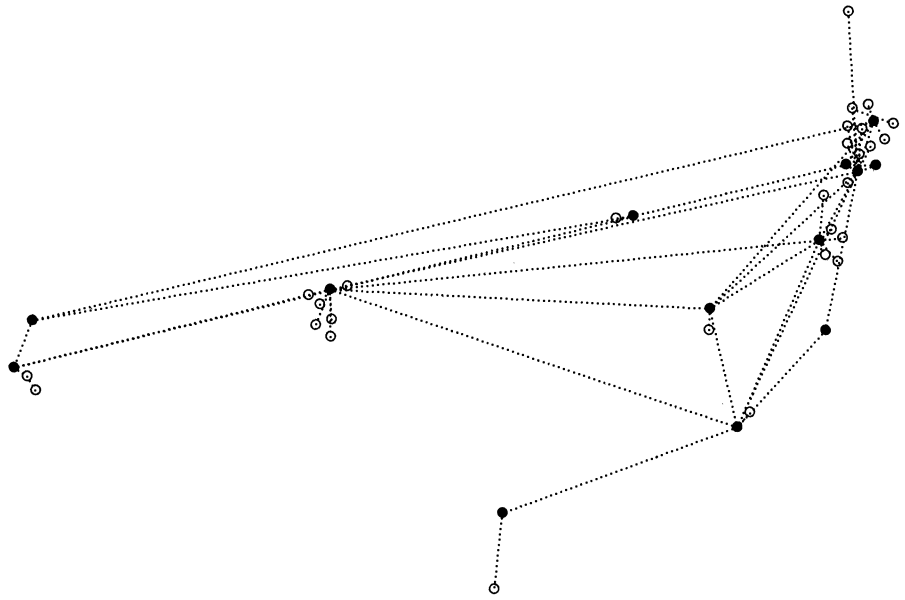


FIG. 3. *Solution for U.S.* 41.

for the backbone network. We added two other nodes to cover some geographical areas. The cost of the local networks was 26000, and U_L was 10000. We solved (8) in less than one minute, and we had a lower bound of 23000. Also in less than one minute we obtained a solution for BP of value 24000.

TABLE 2

	Loc net cost	U_L	Lower bound	BP	NBP	U_N
Value	26K	10K	49K	50K	53K	13K
Comp time	–	–	1 min	1 min	18 min	–

We expanded this to a solution of the entire network of value 50000, and we had a lower bound of 49000. We ran 500 iterations of the procedure in §5 to obtain a solution of NBP. Its value was 53000 and it took 18 minutes. The value of U_N was 13000. We summarize all this in Table 2. As before, from U_L, U_N, and the lower bound, it seems that now one should optimize the local networks or modify the choice of the backbone network. We plot the solution in Figure 4.

6.3. A 64 cities problem in Germany. The third problem we tried consisted of 64 cities in Germany. All pairs of cities could be linked, and we also had a requirement for each pair, i.e., 2016 requirements.

For the backbone network we included nine cities that had larger traffic, and we added one extra city to cover some geographical region. We obtained a cost 1582000 \$/month for the local networks, and U_L was 1370000.

The backbone network was required to be 2-connected, so we added (6) to (8). We solved it in less than two minutes; this gave a lower bound of 904000. After 20 minutes of branch and bound we obtained a lower bound of 973000 for the cost of the backbone network. We also obtained an integer vector $\bar{x}$ of value 1053000. We used this as lower bounds for x in (1), and we obtained a feasible solution of BP of the

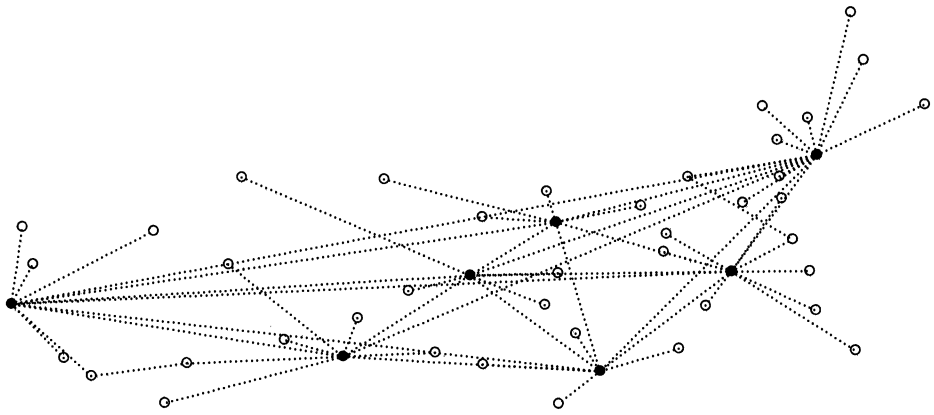


FIG. 4. *A 47 cities problem in the U.S.*

same value; this took less than one minute.

At this point, we had a gap of 8% from optimality for BP in the backbone network. We expanded this to a solution of value 2635000 for BP on the entire network. For it we had a lower bound of 2554000. To obtain a solution for NBP we ran 200 iterations of the routing algoritm; it took 35 minutes. The value was 2669000, and U_N was 1441000.

TABLE 3

	Loc net cost	U_L	Lower bound	BP	NBP	U_N
Value	1582K	1370K	2554K	2635K	2669K	1441K
Comp time	–	–	22 min	1 min	35 min	–

Again, from the value of the lower bound and the value of U_L, it seems that at this point one should optimize the local networks. A summary of this is in Table 3. The solution is shown in Figure 5.

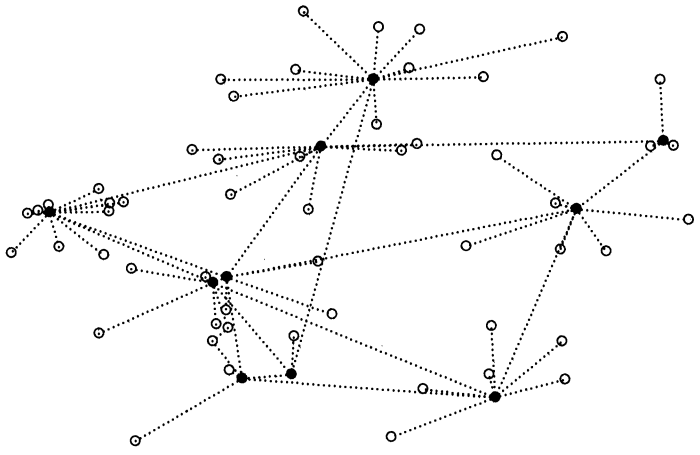


FIG. 5. *A 64 nodes case in Germany.*

6.4. A 10 cities problem in the U.S. Finally we tried a problem with 10

cities in the U.S. The link costs appear in Table 7.6, p. 318 of [18]. All pairs of cities can be linked, and between each pair of cities there is a requirement of 8000 bps (bits per second). The link capacity is 33600 bps. A solution of value 72916 obtained with the MENTOR algorithm [19] is given. Exercise 7.10 of [18] asks to find a better solution.

TABLE 4

	Lower bound	BP	NBP
Value	60251	63606	66273K
Comp time	2 min	1 min	1 min

Since we are trying to solve NBP, we changed the link capacity to 32000 bps. We solved (8) in two minutes to obtain a bound of 58000. After using branch and bound for one minute, we improved the lower bound to 60251, and we had an integer vector of the same value, satisfying the cut condition. We used it to start (1), and after one minute we had a solution of BP of value 63606. Then we ran the routing algorithm for one minute and we had a solution for NBP of value 66273. We summarize all this in Table 4.

We draw the solution in Figure 6, where there are 3 double links marked with solid lines. The routes are as follows: 1-10-2, 1-5-3, 1-6-4, 1-5, 1-6, 1-6-7, 1-10-8, 1-5-9, 1-10, 2-3, 2-7-9-4, 2-3-5, 2-7-6, 2-7, 2-10-8, 2-7-9, 2-10, 3-8-4, 3-5, 3-6, 3-2-7, 3-8, 3-6-4-9, 3-2-10, 4-9-5, 4-6, 4-9-7, 4-8, 4-9, 4-9-10, 5-1-6, 5-9-7, 5-3-8, 5-9, 5-1-10, 6-7, 6-3-8, 6-4-9, 6-7-10, 7-10-8, 7-10-9, 7-10, 8-4-9, 8-10, 9-10.

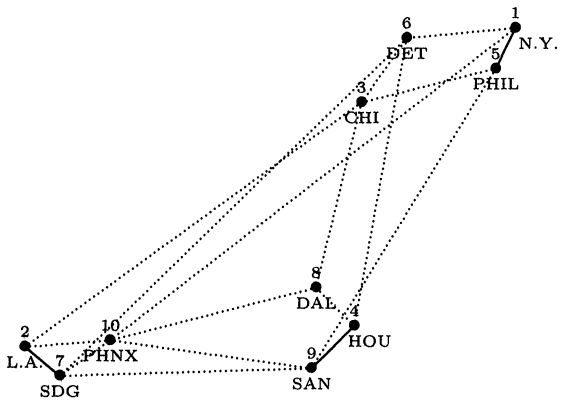


FIG. 6. *A 10 cities problem in the U.S.*

7. Final remarks. We spent at most two hours on each of the problems we tried. This includes trying two or three choices for the backbone network. Dividing the process into backbone optimization and local network optimization seems to be crucial to keep the time within reasonable limits. Notice that less exact heuristics take only a few minutes on these problems.

The important role of the cut inequalities has already been recognized in [13], [20], and [6]. One difference with our approach is that we use an exact algorithm for the separation problem instead of a heuristic. Another difference is that in a first

approximation we do not use the flow variables. This results in a considerable reduction on the number of variables.

We used an integer vector satisfying the cut condition to find a feasible solution, with a small difference in the objective value. It would be very interesting to have a theoretical bound for this gap.

From a solution of the bifurcated case we derived a solution to the nonbifurcated problem. Again, it would be interesting to know a theoretical bound for the gap between these two.

Appendix A. Testing the cut condition. We saw in §3 that testing the cut condition reduces to the max cut problem. We are going to present an algorithm for this, different from the *dual* algorithms used in [4] and [3]. The term *dual* means that a cut was obtained only at the end. We found that this type of method had disadvantages such as slow convergence, and that the information given by a good heuristic solution was not used.

This time we are using a *primal* algorithm. The term *primal* means that at every step, one moves from one cut to a better cut. Since in §3 we were looking for a violated inequality, we do not have to apply this method until the end. We stop when the value of the cut exceeds a certain threshold. This algorithm has been used in [2] for finding minimum-energy configurations for a problem in physics. The physics instances had about 1000 nodes and 3000 edges; it took several hours to solve them. The instances arising in network design were much smaller; it took a few seconds to solve them.

We have a graph $G = (V, E)$, with node set V, edge set E, and weights $w(e)$ for each edge $e = \{i, j\} \in E$. For any node-set $S \subseteq V$ the edge-set

$$\delta(S) = \Big\{ e = \{i, j\} \in E \mid i \in S,\ j \notin S \Big\}$$

is called a *cut*. Notice that the empty set is a cut, and for any two cuts $\delta(S)$ and $\delta(T)$ their symmetric difference $\delta(S) \triangle \delta(T)$ is also a cut, and

$$\delta(S) \triangle \delta(T) = \delta(S \triangle T).$$

The weight of a cut $\delta(S)$ is $\sum\{w(e) : e \in \delta(S)\}$.
Let $\bar{C}$ be any cut. Define

$$w'(e) = \begin{cases} -w(e) & \text{if } e \in \bar{C}, \\ w(e) & \text{if } e \notin \bar{C}. \end{cases}$$

We use the notation $w(C) = \sum\{w(e) : e \in C\}$.
If C is a cut with $w'(C) > 0$,

$$w(\bar{C} \triangle C) = w(\bar{C}) + w'(C) > w(\bar{C}).$$

Thus if we have a new cut C with positive weight with respect to w', then $\bar{C} \triangle C$ is a cut whose weight is larger than the weight of $\bar{C}$.

This suggests a simple method: given a cut $\bar{C}$, define the weights w' and find a new cut C that has positive weight with respect to w', and then take $\bar{C} \triangle C$. The way to ensure that a maximum cut has been found is to prove that there is no cut of positive weight with respect to w'.

To look for a cut of positive weight we first solve the following linear programming problem:

$$
(9)\qquad \begin{cases} \text{maximize } \sum w'(e)\, y(e) \\ \text{subject to} \\ \displaystyle\sum_{f \in C \setminus e} y(f) - y(e) \geq 0, \\ \text{for every cycle } C, \text{ for each edge } e \in C, \\ 1 \geq y(e) \geq 0 \quad \text{for each edge } e. \end{cases}
$$

Since there are an exponential number of inequalities, we use a *cutting plane* approach as follows. We first solve (9) with only a small subset of its inequalities. Given a solution of this, we check for new inequalities that are violated. If there are any, we add them and solve this linear program again. If there is no violated constraint we stop. Finding violated inequalities reduces to a sequence of *shortest path* problems in a graph.

If we obtain a solution of (9) that is a 0-1 vector, then it gives us a cut. If we obtain a fractional vector $\bar{y}$, then we must *branch*: for $0 < \bar{y}(f) < 1$, we decompose (9) into two subproblems, one with $y(f) = 0$ and the other with $y(f) = 1$, and solve again. Branching may end up producing an exponential number of subproblems. Fortunately in all the cases we tried, very little or no branching was needed.

When we obtain a solution of value 0, we have a proof that a maximum cut has been reached.

Many times it was not necessary to solve (9) until the end. Given a fractional vector $\bar{y}$, we used it as the input of a heuristic that would try to derive a cut of positive weight. This has been very important for the efficiency of the method.

Appendix B. Separation of multicut inequalities. Given a graph $G = (V, E)$, the spanning tree polyhedron $P(G)$ is the convex hull of incidence vectors of spanning trees plus the nonnegative orthant.

We use $x(S)$ to denote $\sum_{e \in S} x(e)$. The system below defines $P(G)$.

$$
(10)\qquad \begin{aligned} &x(\delta(V_1, \ldots, V_p)) \geq p - 1 \text{ for every partition of } V, \\ &x \geq 0. \end{aligned}
$$

It was shown in [1] that the separation problem reduces to $|V|$ minimum cut problems. We sketch this procedure.

Jünger and Pulleyblank [17] have given an extended formulation for $P(G)$ as follows. Associate the variables x with the edges and the variables y with the nodes. The following system defines a polyhedron whose projection onto the variables x is $P(G)$. The node r is an arbitrary element of V.

$$
(11)\qquad \begin{cases} x(\delta(S)) + y(S) \geq 2 & \text{if } r \notin S,\ S \subset V, \\ x(\delta(S)) + y(S) \geq 0 & \text{if } r \in S,\ S \subset V, \\ y(V) = 0. \end{cases}
$$

So given a vector $\bar{x}$ we can try to find a vector $\bar{y}$ such that $(\bar{x}, \bar{y})$ satisfies (11) or prove that $\bar{y}$ does not exist.

Let

$$f(S) = \begin{cases} 2 - \bar{x}(\delta(S)) & \text{if } r \notin S, \\ -\bar{x}(\delta(S)) & \text{if } r \in S \end{cases}$$

for $\emptyset \neq S \subseteq V$. The function $-f$ is submodular.

We are going to solve

$$\begin{cases} \text{minimize } y(V) \\ \text{subject to} \\ y(S) \geq f(S) \text{ for } S \subseteq V. \end{cases} \tag{12}$$

As shown by Edmonds [12], the greedy algorithm solves this linear program. This algorithm also produces an optimal solution of the dual problem. This will give us a most violated inequality (10), if there is any. The dual problem is

$$\begin{cases} \text{maximize } \sum z_S f(S) \\ \text{subject to} \\ \sum \{z_S | u \in S\} = 1 \text{ for all } u \in V, \\ z \geq 0. \end{cases} \tag{13}$$

Given a vector y satisfying (12), a set S is called tight if $y(S) = f(S)$. The function $y(\cdot) - f(\cdot)$ is nonnegative and submodular. So if S and T are tight and $S \cap T \neq \emptyset$, then $S \cap T$ and $S \cup T$ are also tight.

We start with $y(v_i) = 2\ \forall i$ and decrease the value of each $y(v_i)$ until a set becomes tight. We denote by $\mathcal{F}$ the family of tight sets with a positive dual variable. If we try to add S to $\mathcal{F}$ and there is a set $T \in \mathcal{F}$ with $S \cap T \neq \emptyset$, then we replace S and T by $S \cup T$. This is also tight.

Let $V = \{v_1, \ldots, v_n\}$. The algorithm is below.

- Step 0. Set $\bar{y}(v_i) \leftarrow 2$ for $i = 1, \ldots, n$; $k \leftarrow 1$; $\mathcal{F} \leftarrow \emptyset$.
- Step 1. If v_k belongs to a set in $\mathcal{F}$ go to Step 3; otherwise
  set $\alpha \leftarrow f(\bar{S}) - \bar{y}(\bar{S}) = \max\{f(S) - \bar{y}(S) \mid v_k \in S\}$,
  $\bar{y}(v_k) \leftarrow \bar{y}(v_k) + \alpha$,
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{\bar{S}\}$.
- Step 2. While there are two sets S and T in $\mathcal{F}$ with $S \cap T \neq \emptyset$ do
  $\mathcal{F} \leftarrow (\mathcal{F} \backslash \{S, T\}) \cup \{T \cup S\}$.
- Step 3. Set $k \leftarrow k + 1$. If $k \leq n$ go to Step 1; otherwise stop.

The vector $\bar{y}$ is built to satisfy (12), the family $\mathcal{F}$ defines a partition of V, and $\bar{y}(S) = f(S)$ for every $S \in \mathcal{F}$. We set $\bar{z}_S = 1$ if $S \in \mathcal{F}$ and $\bar{z}_S = 0$ otherwise. We have

$$\bar{y}(V) = \sum \{\bar{y}(S) \mid S \in \mathcal{F}\} = \sum \{f(S) \mid S \in \mathcal{F}\} = \sum \{f(S)\, \bar{z}_S \mid S \subseteq V\}.$$

This proves that $\bar{y}$ and $\bar{z}$ are optimal solutions.

If the value of the optimum is 0 then $(\bar{x}, \bar{y})$ satisfies (11). In this case we can pick any partition of V into $V_1, \ldots, V_p$, add the inequalities in (11) associated with the sets $\{V_i\}$ and $-y(V) \geq 0$, and obtain an inequality (10). This shows that $\bar{x}$ satisfies all inequalities (10).

Now assume that the value of the optimum is greater than 0. Let $\tilde{z}$ be a 0-1 vector that satisfies the equations of (13), the family $\mathcal{G} = \{S \mid \tilde{z}_S = 1\} = \{S_1, \ldots, S_p\}$ give a partition of the set V, and

$$\sum f(S)\,\tilde{z}_S = 2(p-1) - 2\,\bar{x}\,(\delta(S_1, \ldots, S_p)).$$

Since $\bar{z}$ is an optimum of (13) it gives a most violated inequality (10).

It remains to show how to compute the number α in Step 1. Construct a directed graph $D = (N, A)$, where $N = V \cup \{s, t\}$ and $A = \{(i,j), (j,i) \mid ij \in E\} \cup \{(s,i), (i,t) \mid i \in V\}$.

Define

$$\eta(i) = \bar{y}(i) \quad \text{for } i \in V,\ i \neq r,$$
$$\eta(r) = \bar{y}(r) + 2.$$

Define capacities

$$c(s,i) = -\eta(i),\ c(i,t) = 0 \quad \text{if } \eta(i) < 0,\ i \neq v_k,\ i \in V,$$
$$c(i,t) = \eta(i),\ c(s,i) = 0 \quad \text{if } \eta(i) \geq 0,\ i \neq v_k,\ i \in V,$$
$$c(s, v_k) = \infty,\ c(v_k, t) = 0,$$
$$c(i,j) = c(j,i) = \bar{x}(i,j) \quad \text{for } ij \in E.$$

If $\{s\} \cup T$ induces a cut separating s from t that has capacity λ, we have

$$\bar{y}(T) + \bar{x}(\delta(T)) = \begin{cases} \lambda + \sum\{\eta(v) \mid \eta(v) < 0\} - 2 & \text{if } r \in T, \\ \lambda + \sum\{\eta(v) \mid \eta(v) < 0\} & \text{if } r \notin T. \end{cases}$$

Therefore if β is the minimum capacity of a cut separating s from t, then the value of α is

$$2 - \beta - \sum\{\eta(v) \mid \eta(v) < 0\}.$$

Acknowledgments. I am grateful to Robert S. Cahn and Aaron Kershenbaum for providing the problems and many helpful comments.

REFERENCES

[1] F. BARAHONA, *Separating from the dominant of the spanning tree polytope*, Oper. Res. Lett., 12 (1992), pp. 201–204.

[2] ———, *Magnetization of ground states of Ising spin glasses*, Phys. Rev. B (3), 49 (1994), pp. 12864–12867.

[3] F. BARAHONA, M. GRÖTSCHEL, M. JÜNGER, AND G. REINELT, *An application of combinatorial optimization to statistical physics and circuit layout design*, Oper. Res., 36 (1988), pp. 493–513.

[4] F. BARAHONA AND E. MACCIONI, *On the exact ground-states of three-dimensional Ising spin glasses*, J. Phys. A, 15 (1982), pp. L611–L615.

[5] F. BARAHONA AND A. R. MAHJOUB, *On 2-connected subgraph polytopes*, Discrete Math., 147 (1995), pp. 19–34.

[6] D. BIENSTOCK AND O. GÜNLÜK, *Capacitated Network Design–Polyhedral Structure and Computation*, Tech. report, Columbia University, New York, 1994.

[7] ———, *Computational experience with a difficult multicommodity flow problem*, Math. Programming, 68 (1995), pp. 213–237.

[8] R. S. CAHN, P. CHANG, P. KERMANI, AND A. KERSHENBAUM, *A Study of Network Clustering Algorithms*, Research report RC 16046, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1990.

[9] ———, *INTREPID: An integrated network tool for routing, evaluation of performance, and interactive design*, IEEE Communications Magazine, 29 (1991), pp. 40–47.
[10] S. CHOPRA, *On the spanning tree polyhedron*, Oper. Res. Lett., 8 (1989), pp. 25–29.
[11] IBM CORPORATION, *Optimization subroutine library: Guide and reference*, 1991.
[12] J. EDMONDS, *Submodular functions, matroids, and certain polyhedra*, in Combinatorial Structures and their Applications, R. K. Guy, E. Milner, and N. Sauer, eds., Gordon and Breach, New York, 1970, pp. 69–87.
[13] B. GAVISH AND K. ALTINKEMER, *Backbone network design tools with economic tradeoffs*, ORSA J. Comput., 2 (1990), pp. 236–252.
[14] R. E. GOMORY AND T. C. HU, *Multi-terminal networks flows*, J. SIAM, 9 (1961), pp. 551–570.
[15] M. GRÖTSCHEL, C. L. MONMA, AND M. STOER, *Facets for polyhedra arising in the design of communication networks with low-connectivity constraints*, SIAM J. Optim., 2 (1992), pp. 474–504.
[16] T. C. HU, *Optimum communication spanning trees*, SIAM J. Comput., 3 (1974), pp. 188–195.
[17] M. JÜNGER AND W. R. PULLEYBLANK, *New Primal and Dual Matching Heuristics*, Report 91.105, Institut für Informatik, Universität zu Köln, 1991.
[18] A. KERSHENBAUM, *Telecomunications Network Design Algorithms*, McGraw–Hill, New York, 1993.
[19] A. KERSHENBAUM, P. KERMANI, AND G. A. GROVER, *MENTOR: An algorithm for mesh network topological optimization and routing*, IEEE Trans. Comm., 39 (1991), pp. 503–513.
[20] T. L. MAGNANTI, P. MIRCHANDANI, AND R. VACHANI, *Modeling and solving the capacitated network loading problem*, Oper. Res., 43 (1995), pp. P142–157.
[21] ———, *The convex hull of two core capacitated network design problems*, Math. Programming, 60 (1993), pp. 233–250.
[22] C. S. J. A. NASH-WILLIAMS, *Edge-disjoint spanning trees of finite graphs*, J. London Math. Soc., 36 (1961), pp. 445–450.
[23] C. C. PALMER AND A. KERSHENBAUM, *Representing Trees in Genetic Algorithms*, Research report RC 19718, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1994.
[24] M. W. P. SAVELSBERGH AND G. L. NEMHAUSER, *Functional Description of MINTO, A Mixed Integer Optimizer*, Tech. report, Georgia Institute of Technology, School of Industrial and Systems Engineering, Atlanta, GA, 1993.
[25] W. T. TUTTE, *On the problem of decomposing a graph into n connected factors*, J. London Math. Soc., 36 (1961), pp. 221–230.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 838–857, August 1996
© 1996 Society for Industrial and Applied Mathematics
017

A BILEVEL PROGRAMMING METHOD FOR PIPE NETWORK OPTIMIZATION*

JIANZHONG ZHANG† AND DETONG ZHU†

Abstract. This paper is concerned with the optimal dimensioning problem in which the layout of a pipe network is given but several available diameters can be selected for each pipe. The purpose is to minimize total cost while satisfying certain restrictions. A new method is proposed, the main features of which are to treat the problem as a bilevel programming problem, then to simplify the lower level problem by conjugate duality theory and to handle the upper level problem by its piecewise linear and convex nature. Some numerical testing results are also presented.

Key words. pipe network, bilevel programming, convex programming, conjugate dual, trust region method

AMS subject classifications. 90C90, 90C30, 90C25, 65K05

1. Introduction.

Pipe network optimization is a class of problems dealing with the optimal design issues for supplying gas, natural gas, water, fresh air, etc. by pipeline systems. For clarity we discuss in this paper only gas networks. One of the problems in this field is the following: suppose the layout of the network is given but the diameters of the pipes can be chosen from a set of available sizes with different unit costs. We want to determine diameters for all pipelines so that the cost of realizing the network can be minimized under the condition that the demands of all users are satisfied with high enough gas pressure. There have been quite a few methods proposed for this purpose; see [1]–[6] and the literature quoted therein.

In this paper we propose a new method for solving minimum cost problems. In comparison to existing methods, this one is intended to be more solid from the theoretical point of view, while very convenient to use in practical computation. The main features of our method are that we treat the problem as a bilevel programming problem, then use conjugate duality theory to simplify the lower (inner) level problem and develop a trust region algorithm for the upper (outer) level problem that is based on the piecewise linear nature of its objective function. These ideas and their implementation make the method distinct from all existing methods.

The paper is organized as follows. In §2 we describe the network problem and introduce our notation. In §3 we replace the discrete decision variables by some continuous variables so that the use of nonlinear programming theory and technique in later sections becomes rational. The problem is then formulated as a bilevel programming problem in §4 by introducing some auxiliary functions. We further convert the constrained lower level problem to an unconstrained one. Section 5 provides methods to solve the lower level problem and to obtain its sensitivity information. We study the upper level problem in §6 and simplify the problem by noticing its piecewise linear and convex objective function. In §7 we suggest a subproblem for generating the moving step in the upper level optimization. Based on all previous sections, in §8 we put forward a complete algorithm for solving the network optimization problem and report some numerical testing results in §9. Section 10 contains a short discussion about more general models.

* Received by the editors December 29, 1993; accepted for publication (in revised form) February 1, 1995. This research was supported in part by City University of Hong Kong grant 700126.

† Department of Mathematics, City University of Hong Kong, Kowloon, Hong Kong (mazhang@cityu.edu.hk).

2. Description of the problem. We consider a pipe network for distributing gas in a city or a town. Such a network consists of r supply nodes, m demand nodes, and n pipes (or arcs). The demand nodes are numbered 1 through m and the supply nodes are numbered $m+1$ through $m+r$. Each arc connects two nodes with an assigned direction.

For pipe j ($j = 1, \ldots, n$), let $l_j > 0$ be its length (given), $d_j > 0$ be its diameter, and q_j be the flowrate. We denote by i'_j and i''_j, respectively, the start node and end node of pipe j according to the designated direction. A positive value of q_j means that gas flows in pipe j at the rate of q_j unit volume per unit time from i'_j to i''_j, whereas $q_j < 0$ implies a flow along the reverse direction at a rate $-q_j$.

At node i ($i = 1, \ldots, m+r$), let $p_i > 0$ be the gas pressure, $y_i = p_i^2$ be its energy head, and $b_i \geq 0$ be the demand (if $i \leq m$) or supply (if $i > m$) per unit time (note that if some nodes are neither supply nor demand, but are instead just points where pipes branch or join, we can treat them as demand nodes with $b_i = 0$). We assume in this research that the capacity of each supplier is unlimited: $b_i = +\infty$ for $i = m+1, \ldots, m+r$.

All together there are s different sizes of pipes to choose with the diameters

$$D_1 < D_2 < \cdots < D_s,$$

and the price per unit length of diameter D_k pipe is c_k (known). They satisfy the condition

$$c_1 < c_2 < \cdots < c_s \ .$$

A node-arc incidence matrix $\hat{A} \in M^{(m+r)\times n}$

$$\hat{A} = \begin{bmatrix} A \\ \bar{A} \end{bmatrix},$$

where $A \in M^{m\times n}$ and $\bar{A} \in M^{r\times n}$, is defined by the following formula:

$$a_{ij} = \begin{cases} -1 & \text{if } \ i = i'_j, \\ 1 & \text{if } \ i = i''_j, \\ 0 & \text{otherwise} \ . \end{cases} \tag{2.1}$$

It is easy to see that each column of $\hat{A}$ contains exactly two nonzero entries: a value 1 and a value -1. According to the theory of network flow (see [7], for example), the rank of matrix $\hat{A}$ is $m + r - 1$ and its upper part, i.e., the demand node incidence matrix A, has full row rank m. In other words, all rows of A are linearly independent.

At each demand node a continuity equation

$$\sum_{i=1}^{m} a_{ij}\, q_j = b_i, \qquad i = 1, \ldots, m$$

holds, or by using matrix and vector symbols,

$$Aq = b, \tag{2.2}$$

where $q = (q_1, \ldots, q_n)^T$ and $b = (b_1, \ldots, b_m)^T$.

It is known that under the assumptions of (a) steady state flow, (b) isothermal process, (c) equal elevation, (d) no deviation from ideal gas behavior, (e) constant

friction coefficient over length of pipe, and (f) constant friction factor with respect to q, d, etc. in each pipe the relation between the energy head loss and the flowrate is characterized by the equation

$$K\, l_j\, d_j^{\beta}\, |q_j|^{\alpha-1}\, q_j = y_{i'_j} - y_{i''_j}, \qquad j = 1, \ldots, n, \tag{2.3}$$

where the two technical parameters α and β are in the ranges $1.8 \le \alpha \le 2.0$ and $-5.3 \le \beta \le -4.8$. In our later working examples (see §9), $\alpha = 1/0.54 \approx 1.85$ and $\beta = -4.9074$ are chosen. The value $K > 0$ depends on many factors including the units to be used for l, d, q, and p. (More technical details can be found in [8].)

The pressures at supply nodes are fixed:

$$p_i = \bar{p}_i \text{ (constant)}, \qquad i = m+1, \ldots, m+r,$$

and the pressures at demand nodes must meet the requirement

$$p_i \ge p_i^{\min}, \qquad i = 1, \ldots, m,$$

where the values $p_i^{\min}$ are given lower limits of acceptable gas pressures at the demand nodes of the network. We can introduce $y = (y_1, \ldots, y_m)^T$ and $y^{\min} = (y_1^{\min}, \ldots, y_m^{\min})$, where $y_i^{\min} = (p_i^{\min})^2$, to express the previous requirement equivalently as

$$y \ge y^{\min}. \tag{2.4}$$

Let $d = (d_1, \ldots, d_n)^T$, where

$$d_j \in \Omega \stackrel{\text{def}}{=} \{ D_1, \ldots, D_s \}, \qquad j = 1, \ldots, n \tag{2.5}$$

and denote by $c(d_j)$ the cost per unit length of pipe j if the diameter d_j is selected. Then the total cost of constructing the gas network is given by

$$\text{cost}\,(d) = \sum_{j=1}^{n} l_j\, c(d_j). \tag{2.6}$$

Note that if $d_j = D_k$, then actually $c(d_j) = c_k$.

The optimization problem which we will solve is to minimize the total cost of realizing the gas network under the constraint that the energy heads at the demand nodes are above given lower limits. In other words, we should determine diameters of the pipes by solving the minimization problem

(P1) $\quad \min \ \text{cost}\,(d)$
$\quad\quad$ s.t. constraints (2.1)–(2.5).

3. A transition from discrete variables to continuous ones. As we saw in the last section, the decision variables of the model are discrete, so problem (P1) is actually a nonlinear integer programming problem. So far there are still no effective ways to solve large integer NLP problems. Most methods for problem (P1) or other similar problems are heuristic.

Some methods first disregard the discrete requirement and treat (P1) as a continuous NLP problem by allowing all d_j to select values within a certain range. Once the optimal solution is obtained for the continuous variable problem, the diameters of

the pipes must be rounded to some actually available sizes, and the rounding must be done in such a way that all constraints remain satisfied. We know from integer programming theory that in a design space of high dimension, this type of approximate method may not obtain a near-optimal solution for the original discrete problem (P1). Furthermore, checking feasibility is not a trivial job in this problem because for each trial point, we must solve a set of nonlinear equations to get q and p. Also, since (P1) contains a large number of variables and constraints, trying to solve its corresponding continuous problem in a lump may fail to obtain a solution.

Some papers consider the discrete requirement, and during the process each d_j is chosen from a given set of available sizes. See, for example, [4]–[6]. However, for lack of a suitable technique to use, these papers still resort to some methods such as the trust region technique or one-dimensional search, which are rational only for continuous variable problems. In fact, from the basic principle of the trust region method, it is easy to know that when this type of method is applied to a discrete problem, since the trust region must be large enough to contain adjacent sizes, convergence of the method cannot be assured. Employing a one-dimensional search to a discrete optimization problem shall meet with the same kind of trouble. Moreover, when one uses this kind of continuous method, rounding is again inevitable. As a matter of fact, unlike the first type of method, which rounds only once at the final stage of computation, these methods must complete a rounding and feasibility checking procedure at each iteration.

In this paper we try to avoid these shortcomings in several ways. The first one is to transform discrete decision variables d_j to some continuous variables. To attain the purpose, we allow each pipeline to use different diameters of pipes and let x_{jk} be the percentage of arc j that uses the pipe of diameter D_k. These x_{jk} of course satisfy the conditions

$$x_{jk} \geq 0 \qquad \text{for } j = 1, \ldots, n \quad \text{and} \quad k = 1, \ldots, s \tag{3.1}$$

and

$$\sum_{k=1}^{s} x_{jk} = 1 \qquad \text{for } j = 1, \ldots, n. \tag{3.2}$$

We use them as new decision variables to replace $d_1, \ldots, d_n$. Note that x_{jk} are continuous variables in the range [0, 1]. Since it can be assured (see §§7 and 9) that in the optimal solution each pipeline consists of no more than two sizes of pipes and, furthermore, if along a line two diameters of pipes must be used by our method and in most cases would be of two successive sizes, this revision of the model is acceptable to the local gas company and also to many gas suppliers elsewhere. In fact, using two sizes along an arc can be regarded as splitting the arc into two arcs by inserting a new transferring node with zero demand.

Since the total energy head loss in a series connection is the sum of the losses flowing through each individual subpart, it is easy to see from (2.2) that if we introduce

$$z_j \overset{\text{def}}{=} \sum_{k=1}^{s} x_{jk} D_j^{\beta}, \qquad j = 1, \ldots, n, \tag{3.3}$$

then the flow equations become

$$K l_j z_j |q_j|^{\alpha-1} q_j = y_{i'_j} - y_{i''_j}, \qquad j = 1, \ldots, n. \tag{3.4}$$

If we further define an $n \times n$ diagonal matrix

$$B(q,z) \stackrel{\text{def}}{=} \operatorname{diag}\{\, K l_j z_j \,|q_j|^{\alpha-1} \,\}$$

and use the definition of matrix $\hat{A}$, then equation (3.4) can be simplified to

$$B(q,z)\, q + \hat{A}^T \begin{pmatrix} y \\ \bar{y} \end{pmatrix} = 0,$$

where

$$\bar{y} = (\bar{p}^2_{m+1}, \ldots, \bar{p}^2_{m+r})^T$$

is a given vector. In fact, after introducing a constant vector

$$h \stackrel{\text{def}}{=} \bar{A}^T \bar{y} = (h_1, \ldots, h_m)^T,$$

we can express the equation as

$$B(q,\ z)q + A^T y + h = 0. \tag{3.5}$$

On the other hand, the cost of realizing pipeline j now becomes $\sum_{k=1}^{s} c_k(l_j\, x_{jk})$.

Therefore, after replacing discrete variables d_j $(j = 1, \ldots, n)$ by x_{jk}'s, the model under study can be formulated as

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} l_j \left(\sum_{k=1}^{s} c_k\, x_{jk} \right) \\
\text{s.t.} \quad & z_j = \sum_{k=1}^{s} x_{jk}\, D_k^{\beta}, \qquad j = 1, \ldots, n \\
& Aq = b, \\
& B(q,z)q + A^T y + h = 0, \\
& y \geq y^{\min}, \\
& x_{jk} \geq 0\,, \qquad j = 1, \ldots, n, \quad k = 1, \ldots, s, \\
& \sum_{k=1}^{s} x_{jk} = 1, \qquad j = 1, \ldots, n,
\end{aligned} \tag{P2}$$

which is a nonlinear programming problem with continuous variables x_{jk}, z_j, q_j, and y_i for $i = 1, \ldots, m$, $j = 1, \ldots, n$, and $k = 1, \ldots, s$. As we explained before, in this method no rounding and feasibility checking is needed once we obtain the solution of problem (P2).

4. A bilevel programming formulation. Because solving problem (P2) in a lump often causes numerical difficulties, to solve the problem effectively we treat it as a hierarchical programming problem so that the whole problem is decomposed into several smaller problems with fewer variables and constraints. Let us first consider the set of nonlinear equations

$$\begin{cases} B\,(q,z)\, q + A^T\, y + h = 0, \\ Aq = b \end{cases} \tag{4.1}$$

in the group of constraints for problem (P2). It contains $m+n$ equations, and for any given z, the system has $m+n$ unknowns $q \in R^n$ and $y \in R^m$. As the first author of this paper pointed in his previous research (see [9]), for a given z, if we define

$$f_j(q_j) \stackrel{\text{def}}{=} \frac{1}{1+\alpha} K l_j z_j |q_j|^{\alpha+1} + h_j q_j, \quad j = 1, \ldots, n, \tag{4.2}$$

then

$$f_j'(q_j) = K l_j z_j |q_j|^{\alpha-1} q_j + h_j.$$

So, for the function

$$F(q) \stackrel{\text{def}}{=} \sum_{j=1}^{n} f_j(q_j)$$

we know that

$$\nabla F(q) = B(q,z)q + h. \tag{4.3}$$

Furthermore, since $\alpha > 0$, $F(q)$ must be a strictly convex function. The previous facts mean that $(q(z), y(z))$ is a solution of the nonlinear equations (4.1) if and only if $q(z)$ is the unique solution of the minimization problem

$$\begin{array}{ll} \min & F(q), \\ \text{s.t.} & Aq = b \end{array} \tag{L1}$$

with the associated Lagrange multiplier $y(z)$ (since A has full row rank, the multiplier vector is unique).

The previous analysis shows that problem (P2) can be treated as a bilevel programming problem:

$$\begin{array}{ll} \min & \displaystyle\sum_{j=1}^{n} l_j \left(\sum_{k=1}^{s} c_k x_{jk} \right) \\ \text{s.t.} & \displaystyle z_j = \sum_{k=1}^{s} x_{jk} D_k^{\beta}, \qquad j = 1, \ldots, n, \\ & \displaystyle x_{jk} \geq 0 \quad \forall j, \ \forall k, \qquad \sum_{k=1}^{s} x_{jk} = 1 \quad \forall j, \\ & y(z) \geq y^{\min}, \\ & \text{and } y(z) \text{ is the L-multiplier of problem (L1).} \end{array} \tag{U1}$$

We further transform the lower level problem (L1) into a simpler form. Let

$$\bar{b} \stackrel{\text{def}}{=} A^+ b \qquad \text{and} \qquad u = q - \bar{b}, \tag{4.4}$$

where $A^+ = A^T(AA^T)^{-1}$ is the generalized inverse of A. It is obvious that $Aq = b$ if and only if $Au = 0$. If we define

$$\Phi(u) \stackrel{\text{def}}{=} \sum_{j=1}^{n} \varphi_j(u_j) \stackrel{\text{def}}{=} \sum_{j=1}^{n} f_j(u_j + \bar{b}_j),$$

where $u = (u_1, \ldots, u_n)^T$ and $\bar{b} = (\bar{b}_1, \ldots, \bar{b}_n)^T$, then problem (L1) is equivalent to

$$\text{(L2)} \qquad \begin{array}{ll} \min & \Phi(u), \\ \text{s.t.} & Au = 0. \end{array}$$

For the convex function f_j, we may compute its conjugate function (see [10] or [11]).

$$\begin{aligned} f_j^*(t_j) &\stackrel{\text{def}}{=} \sup_{q_j}\{\, t_j\, q_j - f_j\,(q_j)\,\} \\ &= \sup_{q_j}\left\{ t_j\, q_j - \left(\frac{1}{1+\alpha} K\, l_j\, z_j\, |q_j|^{1+\alpha} + h_j\, q_j\right)\right\} \\ &= K\, l_j\, z_j \sup_{q_j}\left\{ \frac{t_j - h_j}{K\, l_j\, z_j}\, q_j - \frac{1}{1+\alpha}|q_j|^{1+\alpha}\right\}. \end{aligned}$$

Since the supremum value is attained at

$$q_j = (K\, l_j\, z_j)^{-\sigma}\, |t_j - h_j|^{\alpha-1}\,(t_j - h_j),$$

where

$$\sigma = \frac{1}{\alpha} < 1,$$

we obtain

$$\text{(4.5)} \qquad f_j^*(t_j) = (1+\sigma)^{-1}\,(K\, l_j\, z_j)^{-\sigma}\, |t_j - h_j|^{1+\sigma}, \qquad j = 1, \ldots, n.$$

Let the conjugate function of φ_j be ψ_j:

$$\begin{aligned} \psi_j(t_j) &= \sup_{u_j}\,\{\, t_j u_j - \varphi_j(u_j)\,\} \\ &= \sup_{u_j}\,\{\, t_j u_j - f_j(u_j + \bar{b}_j)\,\} \\ &= \sup_{u_j}\,\{\, t_j(u_j + \bar{b}_j) - f_j(u_j + \bar{b}_j)\,\} - t_j \bar{b}_j \\ &= f_j^*(t_j) - t_j \bar{b}_j. \end{aligned}$$

If we denote by $\Psi(t)$ the conjugate function of $\Phi(u)$, where $t = (t_1, \ldots, t_n)^T$, then it is easy to see that

$$\begin{aligned} \Psi(t) &= \sum_{j=1}^{n} \psi_j(t_j) \\ &= \sum_{j=1}^{n} \left\{\, (1+\sigma)^{-1}(K\, l_j\, z_j)^{-\sigma}\, |t_j - h_j|^{1+\sigma} - t_j \bar{b}_j \,\right\}. \end{aligned} \qquad \text{(4.6)}$$

Let $L \stackrel{\text{def}}{=} \{\, u | Au = 0\,\}$ be the null space of A. Then, since Φ and Ψ are both differentiable convex functions with dom Φ = dom $\Psi = R^n$, by Fenchel's conjugate duality theorem (see Corollary 31.4.2 of [10]) we know that

$$\text{(4.7)} \qquad \min_{u \in L} \Phi(u) = -\min_{t \in L^{\perp}} \Psi(t),$$

and $\hat{u}$ and $\hat{t}$ are a pair of optimal solutions if and only if

$$\hat{u} \in L, \quad \hat{t} \in L^{\perp}, \quad \hat{u} = \nabla\Psi(\hat{t}), \quad \hat{t} = \nabla\Phi(\hat{u}). \tag{4.8}$$

Obviously $L^{\perp} = \mathcal{R}(A^T) = \{ A^T e \mid \forall e \in R^n \}$. Therefore, the lower level minimization problem (L2) is equivalent to

$$\begin{aligned} &\min \{(1+\sigma)^{-1} \sum_{j=1}^{n} (K l_j z_j)^{-\sigma} |t_j - h_j|^{1+\sigma} - e^t b\}, \\ &\text{where } t = A^T e \quad \forall e \in R^n. \end{aligned} \tag{L3}$$

Furthermore, from (4.4) and (4.8) we obtain

$$q_j = \hat{u}_j + \bar{b}_j = \psi_j'(\hat{t}_j) + \bar{b}_j = (K l_j z_j)^{-\sigma} |\hat{t}_j - h_j|^{\sigma-1} (\hat{t}_j - h_j) \tag{4.9}$$

and

$$\hat{t}_j = \varphi_j'(\hat{u}_j) = f_j'(q_j) = K l_j z_j |q_j|^{\alpha-1} q_j + h_j. \tag{4.10}$$

In fact the two previous equations reflect the same relation between q_j and $\hat{t}_j$.

It is interesting to see from (4.10) that

$$A^T \hat{e} = \hat{t} = B(q, z) + h.$$

Comparing this result with equation (3.5) we know that

$$y(z) = -\hat{e}. \tag{4.11}$$

So, using (L3) to replace (L2) has two important advantages. First, (L2) is a constrained optimization problem, whereas (L3) is an unconstrained one (we substitute t in its objective function by $A^T e$) with the same number of variables, and hence is much easier to solve. Second, in the upper level problem (U1) it is $y(z)$, rather than $q(z)$, that should be controlled by the constraint $y(z) \geq y^{\min}$. By solving (L3) we are able to obtain $y(z)$ explicitly with great ease because it is just the opposite value of the solution $\hat{e}$.

5. Further details of the lower level calculation. If we define

$$\Theta(e) \stackrel{\text{def}}{=} \Psi(A^T e),$$

then problem (L3) becomes

$$-\min_{e \in R^n} \Theta(e). \tag{L4}$$

To solve problem (L4) efficiently, it is important to know the first- and second-order differential information. Let

$$\theta_j(t_j) = (1+\sigma)^{-1} (K l_j z_j)^{-\sigma} |t_j - h_j|^{1+\sigma} - t_j \bar{b}_j. \tag{5.1}$$

Then

$$\Theta(e) = \sum_{j=1}^{n} \theta_j(t_j) \Big|_{t_j = a_{(j)}^T e},$$

where $a_{(j)}$ is the jth column of A. Thus,

$$\nabla_e \Theta = \sum_{j=1}^{n} \frac{d\theta_j}{dt_j} a_{(j)}$$

$$= A\,(\theta_1'(t_1), \ldots, \theta_n'(t_n))^T \tag{5.2}$$

and

$$\nabla_e^2 \Theta = \sum_{j=1}^{n} \frac{d^2\theta_j}{dt_j^2} a_{(j)}\, a_{(j)}^T$$

$$= A \operatorname{diag}\{\, \theta_1''(t_1), \ldots, \theta_n''(t_n) \,\} A^T. \tag{5.3}$$

As for $j = 1, \ldots, n$,

$$\theta_j'(t_j) = (K\, l_j\, z_j)^{-\sigma} |t_j - h_j|^{\sigma-1}\, (t_j - h_j) - \bar{b}_j$$

and

$$\theta_j''(t_j) = \sigma (K\, l_j\, z_j)^{-\sigma} |t_j - h_j|^{\sigma-1}.$$

It is immediate that

$$\nabla_e \Theta = A\omega(z, e) - b \tag{5.4}$$

and

$$\nabla_e^2 \Theta = A \operatorname{diag}\{\, \sigma (K\, l_j\, z_j)^{-\sigma} |t_j - h_j|^{\sigma-1} \,\} A^T, \tag{5.5}$$

where

$$\omega = (\omega_1, \ldots, \omega_n)^T \in R^n$$

and

$$\omega_j = (K\, l_j\, z_j)^{-\sigma} |t_j - h_j|^{\sigma-1}\, (t_j - h_j), \qquad j = 1, \ldots, n. \tag{5.6}$$

In formulas (5.5)–(5.6), t should be substituted by $A^T e$. Since the Hessian matrix of Θ can be expressed explicitly in a concise way, we recommend solving the lower level problem by Newton's method with some modifications to ensure global convergence. In our practice a trust region strategy has been adopted; see [12]–[14] for details. Since the Hessian is at least positive semidefinite, Powell's single dogleg method [12] and Dennis and Mei's double dogleg method [14] are both applicable to solving the quadratic trust region subproblem quickly and approximately (our practice shows that the performances of the two variations do not differ much; see [15]). Note that the structure of $\nabla^2 \Theta$ is very similar to that of the matrices encountered in interior point methods for LP problems; see [16]–[18], for instance. Therefore, the numerical techniques developed in recent years for interior point methods to handle large problems can be employed in our case. But we shall not discuss this aspect in the paper.

So far we see that for a given z, the solution $(q(z), y(z))$ of equations (4.1) can be obtained by solving problem (L4). It is also useful to obtain sensitivity information:

$$\nabla_z q = \begin{bmatrix} \frac{\partial q_1}{\partial z_1} & \cdots & \frac{\partial q_1}{\partial z_n} \\ & \cdots & \\ \frac{\partial q_n}{\partial z_1} & \cdots & \frac{\partial q_n}{\partial z_n} \end{bmatrix} \quad \text{and} \quad \nabla_z y = \begin{bmatrix} \frac{\partial y_1}{\partial z_1} & \cdots & \frac{\partial y_1}{\partial z_n} \\ & \cdots & \\ \frac{\partial y_m}{\partial z_1} & \cdots & \frac{\partial y_m}{\partial z_n} \end{bmatrix}.$$

By differentiating equations (4.1) with respect to z, it is not difficult to obtain

$$\begin{cases} M(q,z)\,\nabla_z q + A^T \nabla_z y &= -\Gamma(z), \\ A\,\nabla_z q &= 0, \end{cases} \tag{5.7}$$

where

$$M(q,z) = \text{diag}\ \left\{ \alpha K\, l_j\, z_j\, |q_j|^{\alpha-1} \right\}_{n\times n} \tag{5.8}$$

and

$$\begin{aligned} \Gamma(z) &= \text{diag}\ \left\{ K\, l_j\, |q_j|^{\alpha-1} q_j \right\}_{n\times n} \\ &= \text{diag}\ \left\{ \frac{\hat{t}_j - h_j}{z_j} \right\}_{n\times n}. \end{aligned} \tag{5.9}$$

The last equality is obtained from (4.10). From (5.7) we derive that

$$\nabla_z y = -(A\,M(q,z)^{-1}\,A^T)^{-1} A\,M(q,z)^{-1}\,\Gamma(z).$$

In other words, $\nabla_z y$ can be obtained by solving the equations

$$(AM(q,z)^{-1}\,A^T)\,\nabla_z y = -AM(q,z)^{-1}\,\Gamma(z). \tag{5.10}$$

Using (4.9) and (4.10), we know that

$$\begin{aligned} \sigma(K\, l_j\, z_j)^{-\sigma} |\hat{t}_j - h_j|^{\sigma-1} &= \frac{1}{\alpha}\,\frac{q_j}{\hat{t}_j - h_j} \\ &= (\alpha K\, l_j\, z_j\, |q_j|^{\alpha-1})^{-1}. \end{aligned}$$

Therefore, from (5.5) and (5.8) we have

$$\nabla^2 \Theta(\hat{e}) = AM(q,z)^{-1}\,A^T, \tag{5.11}$$

which means that the coefficient matrix of equations (5.10) is nothing but the Hessian matrix of $\Theta(e)$ at the solution point $\hat{e}$. So, this matrix is readily provided after solving the lower level problem (L4) by any Newton-type method. Note also that (5.10) actually contains n sets of linear equations, one set for obtaining each column of ∇y. But since they share a common coefficient matrix, one Cholesky decomposition is enough to solve all of them.

6. Simplification of the upper level problem. We now turn to the upper level problem. First we consider the following: for a given $\zeta \in [D_s^\beta, D_1^\beta]$, determine $\hat{\rho} = (\hat{\rho}_1, \ldots, \hat{\rho}_s)^T$ to solve the auxiliary problem

$$\begin{aligned} g(\zeta) \overset{\text{def}}{=} \min & \sum_{k=1}^{s} c_k \rho_k \\ \text{s.t. } \zeta &= \sum_{k=1}^{s} D_k^\beta\, \rho_k, \\ & \sum_{k=1}^{s} \rho_k = 1, \\ & \rho_k \geq 0, \quad k = 1, \ldots, s, \end{aligned} \tag{A1}$$

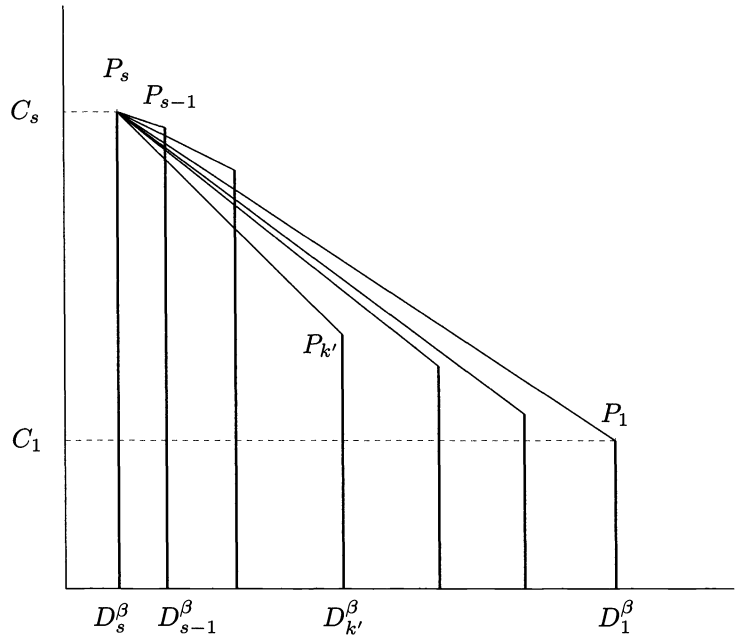


FIG. 6.1.

where the meanings of constants c_k and D_k are given in §2. If we denote by P_k in Figure 6.1 the point (D_k^β, c_k), then we can find a very simple way to solve the problem. Here we just explain the method graphically because corresponding analytic formulas can be obtained with ease.

Let $\gamma_1 = D_s^\beta$ and $\eta_1 = c_s$. From each $P_k (k = 1, \ldots, s-1)$ draw a line segment to link it with P_s. Suppose the lowest one (i.e., with the most negative slope) among the $s-1$ line segments is the one to link P_s and $P_{k'}$ $(1 \le k' \le s-1)$. Then set $\gamma_2 = D_{k'}^\beta$ and $\eta_2 = c_{k'}$. For any $\zeta \in [\gamma_1, \gamma_2]$, express it as a convex combination of γ_1 and γ_2, say $\zeta = \lambda_1\gamma_1 + \lambda_2\gamma_2$, where $\lambda_1, \lambda_2 \ge 0$ satisfy $\lambda_1 + \lambda_2 = 1$. It can be proved (see [19]) that for $\zeta \in [\gamma_1, \gamma_2]$, the optimal solution of problem (A1), denoted by $\hat{\rho}(\zeta) = (\hat{\rho}_1, \ldots, \hat{\rho}_s)$, is

$$\hat{\rho}_k = \begin{cases} \lambda_1 & \text{if } k = s, \\ \lambda_2 & \text{if } k = k', \\ 0 & \text{otherwise,} \end{cases} \tag{6.1}$$

and the optimal value $g(\zeta) = \lambda_1\eta_1 + \lambda_2\eta_2$. In other words, this part of $g(\zeta)$ is just the linear interpolation function of points (γ_1, η_1) and (γ_2, η_2).

For $\zeta > \gamma_2$, we may continue this process by replacing the start point P_s by $P_{k'}$ until the right end point of the lowest line segment becomes P_1; see Figure 6.2.

Therefore, we can prove the following in a constructive way.

PROPOSITION 6.1. *Problem* (A1) *has the following properties.*

1. *For any* $\zeta \in [D_s^\beta, D_1^\beta]$, *the optimal solution* $\hat{\rho}(\zeta)$ *contains no more than two nonzero components.*

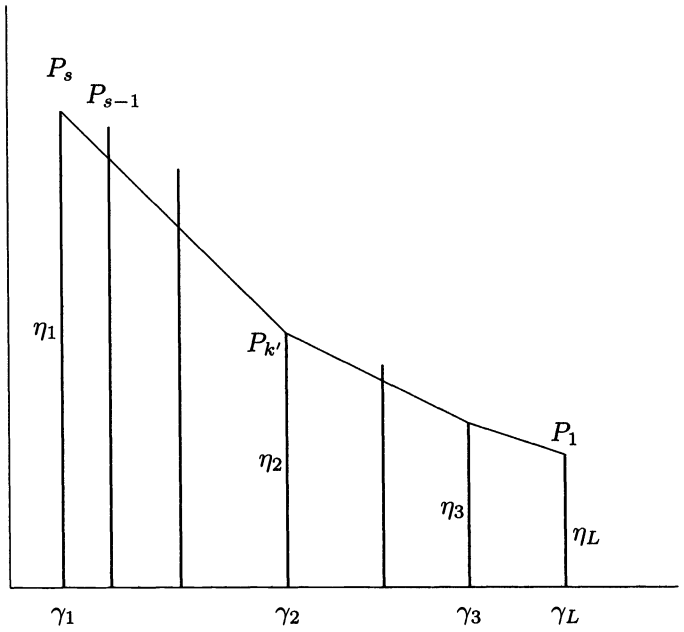


FIG. 6.2.

2. *The optimal value function $g(\zeta)$ is a piecewise linear convex function with a sequence of increasing but negative slopes.*

In what follows, we assume the turning points on the graph of $g(\zeta)$ are respectively $(\gamma_1, \eta_1) = P_s$, $(\gamma_2, \eta_2), \ldots, (\gamma_L, \eta_L) = P_1$ $(L \leq s)$.

It is easy to know from the previous method that if the function obtained by linking each pair of adjacent points P_k and P_{k+1} $(k = 1, \ldots, s-1)$ is a convex one over $[D_s^\beta, D_1^\beta]$, then $g(\zeta)$ is just this function, i.e., $L = s$ and $\gamma_2 = D_{s-1}^\beta$, $\gamma_3 = D_{s-2}^\beta, \ldots, \gamma_s = D_1^\beta$.

Based on the previous results, we know that for each $j = 1, \ldots, n$, the problem

$$\begin{aligned} \min \quad & \sum_{k=1}^{s} c_k x_{jk} \\ \text{s.t.} \quad & z_j = \sum_{k=1}^{s} x_{jk} D_k^\beta, \\ & x_{jk} \geq 0, \quad \sum_{k=1}^{s} x_{jk} = 1, \quad j = 1, \ldots, n \end{aligned}$$

has an optimal value $g(z_j)$ with the solution $(\hat{x}_{j1}, \ldots, \hat{x}_{js})^T = \hat{\rho}(z_j)$, which is obtained by substituting ζ in (A1) by z_j.

So, if we define

$$G_j(z_j) \stackrel{\text{def}}{=} l_j \, g(z_j) \tag{6.2}$$

and

$$G(z) \stackrel{\text{def}}{=} \sum_{j=1}^{n} G_j(z_j), \tag{6.3}$$

then the upper level problem (U1) can be simplified to

$$\begin{aligned} \text{(U2)} \qquad & \min \ G(z) \\ & \text{s.t.} \ \ y(z) \geq y^{\min} \\ & \qquad D_s^{\beta} \leq z_j \leq D_1^{\beta}, \qquad j = 1, \ldots, n. \end{aligned}$$

Note that for each z between the upper and lower bounds, $y(z)$ is obtained from the lower level problem (L4), whereas $G(z)$ is calculated by using the method proposed in this section. Proposition 6.1 ensures that for each $z_j \in [\, D_s^{\beta},\ D_1^{\beta}\,]$ at most two of $\hat{x}_{j1}, \ldots, \hat{x}_{js}$ can be positive. It means that in our solution each pipeline comprises no more than two sizes of pipes.

7. A subproblem for upper level calculation. Since $G(z)$ in problem (U2) is a piecewise linear convex function, along any direction $\bar{d}$ the directional derivative $G'(z;\ \bar{d})$ exists. In fact,

$$G'(z;\ \bar{d}) = \max_{t \in \partial G(z)} \langle t,\ \bar{d} \rangle, \tag{7.1}$$

where $\partial G(z)$ is the subdifferential of G : $\partial G(z) = \partial G_1(z_1) \times \partial G_2(z_2) \times \cdots \times \partial G_n(z_n)$. Since $G_j(z_j) = l_j\, g(z_j)$, all $G_j(\cdot)$ have the same set of turning points $(\gamma_1,\ \eta_1), \ldots, (\gamma_L,\ \eta_L)$ as $g(\cdot)$ does. Thus for each $j = 1, \ldots, n$, if $z_j \in (\gamma_k,\ \gamma_{k+1})$ for any $k = 1, \ldots, L-1$, then G_j is differentiable and

$$G_j'(z_j) = l_j \frac{\eta_{k+1} - \eta_k}{\gamma_{k+1} - \gamma_k}, \tag{7.2}$$

whereas if z_j coincides with a γ_k, then G_j' does not exist, but the two one-sided derivatives are still available:

$$G_{j+}'(\gamma_k) = l_j \frac{\eta_{k+1} - \eta_k}{\gamma_{k+1} - \gamma_k}, \quad k = 1, \ldots, L-1 \tag{7.3}$$

and

$$G_{j-}'(\gamma_k) = l_j \frac{\eta_k - \eta_{k-1}}{\gamma_k - \gamma_{k-1}}, \quad k = 2, \ldots, L. \tag{7.4}$$

We may further define

$$G_{j+}'(\gamma_L) = \infty \quad \text{and} \quad G_{j-}'(\gamma_1) = -\infty. \tag{7.5}$$

Using these derivatives, we have

$$\partial G_j(z_j) = \{\, t_j \in R \mid G_{j-}'(z_j) \leq t_j \leq G_{j+}'(z_j) \,\}.$$

Therefore,

$$\begin{aligned} G'(z;\ \bar{d}) &= \max_{t \in \partial G(z)} \langle t,\ \bar{d} \rangle \\ &= \max\{\, \langle t, \bar{d} \rangle \mid G_{j-}'(z_j) \leq t_j \leq G_{j+}'(z_j), \quad j = 1, \ldots, n \,\}, \end{aligned} \tag{7.6}$$

where t_j are the components of vector t.

To solve the upper level problem (U2), we try moving the iterative point z to reduce the objective function value $G(z)$ while maintaining feasibility. The following subproblem is a natural choice:

$$
\text{(S1)} \qquad \begin{aligned}
\min \quad & G'(z;\ \bar{d}) \\
\text{s.t.} \quad & y_i(z) + \nabla y_i(z)^T \bar{d} \geq y_i^{\min}, \qquad i = 1, \ldots, m, \\
& D_s^{\beta} \leq z_j + \bar{d}_j \leq D_1^{\beta}, \qquad j = 1, \ldots, n, \\
& \|\bar{d}\| \leq \triangle,
\end{aligned}
$$

where $G'(z;\ \bar{d})$ is provided by solving problem (7.6) and ∇y_i^T is the ith row of $\nabla_z y$, which is the solution of equations (5.10). This is a subproblem of trust region-type. The introduction of the trust region radius $\triangle$ is reasonable because the linear approximation for functions G and y is meaningful only in a neighbourhood of z.

We now consider a practical way to solve the subproblem. It is desirable to combine (S1) with the calculation of $G'(z;\ \bar{d})$. As we see, for any given $\bar{d}$, (7.6) is an LP problem. Thus, by duality theorem, $G'(z;\ \bar{d})$ can be expressed as a minimum value of the following dual problem:

$$
\text{(7.7)} \qquad \begin{aligned}
G'(z;\ \bar{d}) \ = \ \min \quad & \sum_{j=1}^{n} G'_{j-}(z_j)\, v_j + \sum_{j=1}^{n} G'_{j+}(z_j)\, w_j \\
\text{s.t.} \quad & v_j + w_j = \bar{d}_j\,, \quad j = 1, \ldots, n \\
& v_j \leq 0, \quad w_j \geq 0, \quad j = 1, \ldots, n.
\end{aligned}
$$

Now it is clear that we are able to incorporate the LP problem (7.7) with (S1):

$$
\text{(S2)} \qquad \begin{aligned}
\min \quad & G'(z;\ \bar{d}) = \sum_{j=1}^{n} G'_{j-}(z_j)\, v_j + \sum_{j=1}^{n} G'_{j+}(z_j)\, w_j \\
\text{s.t.} \quad & y_i(z) + \nabla y_i^T(z)\,(v + w) \ \geq \ y_i^{\min}, \quad i = 1, \ldots, m, \\
& D_s^{\beta} \leq z_j + v_j + w_j \leq D_1^{\beta}, \quad j = 1, \ldots, n, \\
& v_j \leq 0, \quad w_j \geq 0, \quad j = 1, \ldots, n, \\
& \|\, v + w \,\| \ \leq \ \triangle.
\end{aligned}
$$

At each iteration of the upper level calculation, (S2) is the subproblem we choose to solve because it is much easier to handle than problem (S1). Especially if the l_∞ norm or the l_1 norm is used in the trust region constraint, (S2) simply becomes an LP problem of the $2n$ decision variables v and w. After solving it, we take $\bar{d} = v + w$ and consider $z + \bar{d}$ as a candidate for the next iterative point, provided that $z + \bar{d}$ can pass a test stated in §8.

It can be proved (see [19]) that if the current z is not a stationary point, then for

any $\triangle > 0$, there must exist a vector $\hat{d}$ such that

$$G'(z;\ \hat{d}) < 0,$$
$$y(z) + \nabla y(z)\,\hat{d} \geq y^{\min},$$
$$D_s^{\beta}\mathbf{1} \leq z + \hat{d} \leq D_1^{\beta}\mathbf{1},$$
$$\|\hat{d}\| \ \leq \ \triangle,$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$. In other words, problems (S1) and (S2) must result in a solution vector along which the directional derivative of G is negative. This fact makes the proposition of problems (S1) and (S2) reasonable.

8. Algorithm. We are now ready to state our algorithm for the optimal pipe network dimensioning problem.

The initial work includes giving a tolerance parameter $\varepsilon > 0$, two parameters $0 < \sigma_1 < 1 < \sigma_2$ for adjusting the trust region in problem (S2), a positive parameter δ (< 0.3) for measuring the ratio of the actual reduction of G to the predicted reduction, an initial trust region radius $\triangle_0$, and an initial point z^0 that satisfies $y(z^0) \geq y^{\min}$ ($y(z^0)$ is known). Furthermore, using the given data c_k and D_k^{β} ($k = 1, \ldots, s$), we fully characterize function $g(\zeta)$ over the interval $[\,D_s^{\beta},\ D_1^{\beta}\,]$ by storing its turning points $(\gamma_i,\ \eta_i)$ $(i = 1, \ldots, L)$. Calculate $G(z^0)$ by (6.2)–(6.3). Set $k = 0$.

The following are iterative steps where a superscript k is used to indicate the sequence of points generated.

1. Compute $\nabla_z y(z^k)$ by solving equation (5.10) and $G'_{j-}(z^k)$ and $G'_{j+}(z^k)$ by formulas (7.2)–(7.5).
2. Solve subproblem (S2) to obtain $\bar{d}^k = v^k + w^k$ and $G'(z^k;\ \bar{d}^k)$.
3. If $G'(z^k;\ \bar{d}^k) \ \geq \ -\varepsilon$, go to step 7.
4. Calculate $G(z^k + \bar{d}^k)$ and

$$\xi \ \stackrel{\text{def}}{=} \ \frac{G(z^k) - G(z^k + \bar{d}^k)}{-G'(z^k;\ \bar{d}^k)}.$$

   If $\xi < \delta$, then let $\triangle_k \ \leftarrow \ \sigma_1 \triangle_k$ and go to step 2.
5. Calculate $y(z^k + \bar{d}^k)$ by solving problem (L4). If $y(z^k + \bar{d}^k) \geq y^{\min}$ does not hold, then let $\triangle_k \ \leftarrow \ \sigma_1 \triangle_k$ and go to step 2.
6. Set $z^{k+1} = z^k + \bar{d}^k$, $G(z^{k+1}) \ \leftarrow \ G(z^k + \bar{d}^k)$, and

$$\triangle_{k+1} \ = \ \begin{cases} \sigma_2\,\triangle_k & \text{if } \xi \geq 0.7, \\ \sigma_1\,\triangle_k & \text{if } \xi \leq 0.3, \\ \triangle_k & \text{otherwise;} \end{cases}$$

   let $k \ \leftarrow \ k + 1$ and go to step 1.
7. Calculate

$$(x_{j1}^*, \ldots, x_{js}^*)^T = \hat{\rho}(z_j^k), \quad j = 1, \ldots, n$$

   (see §6). x_{jk}^* $(j = 1, \ldots, n;\ k = 1, \ldots, s)$ provide an optimal solution of the problem. Stop.

9. Numerical test. We shall present two examples in this section. Example 1 is quoted from [5], where Hansen used it to illustrate his GASOP code.

Example 1.

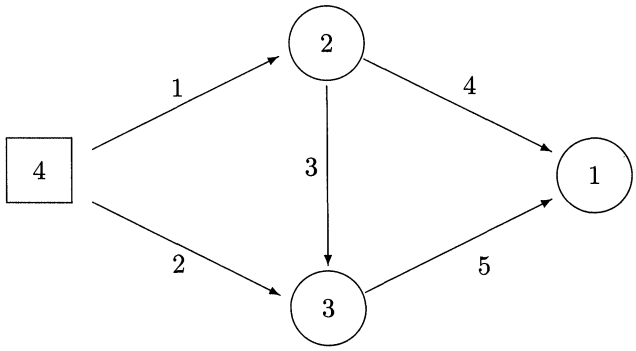


FIG. 9.1.

As we can see from Figure 9.1, $m = 3, r = 1, n = 5$. There are $s = 5$ available pipe diameters, as shown in Table 9.1.

TABLE 9.1.

k	D_k (10 cm)	c_k (DKK/m)
1	0.320	398
2	0.506	431
3	0.725	461
4	1.008	527
5	1.291	598

For each pipe j, its length l_j, start node i'_j, end node i''_j, and initial choice of the diameter type k (we use one diameter only when we start computation) are shown in Table 9.2.

TABLE 9.2.

j	l_j (km)	i'_j	i''_j	k
1	1.0	4	2	5
2	1.0	4	3	5
3	0.8	2	3	4
4	0.8	2	1	4
5	0.6	3	1	4

The data for the demands are

node i	1	2	3
b_i (m^3/h)	225	750	475

and $y_i^{\min} = 4.84$ (bar^2) for $i = 1, 2, 3$.

The other necessary data are as follows. Energy head at the supply node: $\bar{y} = \bar{p}_4^2 = 21.16$ (bar^2),

$$\sigma = 0.54, \quad \beta = -4.9074, \quad K = 261035.0.$$

Our computational results for this example are

$$\begin{aligned} x^*_{1,2} &= 0.312, & x^*_{1,3} &= 0.688, \\ x^*_{2,2} &= 1.0, \\ x^*_{3,1} &= 1.0, \\ x^*_{4,1} &= 0.145, & x^*_{4,2} &= 0.855, \\ x^*_{5,1} &= 1.0, \end{aligned}$$

and all other $x^*_{jk} = 0$, which means that pipelines 1 and 4 should consist of two diameters of pipes with the proportion shown previously, whereas lines 3 and 5 just use the pipes of the smallest diameter and the whole line 4 should use the second smallest size. Note that in this example and the next one, if a pipe line needs to use two different sizes, the two diameters must be adjacent because in the two examples c_k and D_k^β maintain a convex relation, as we discussed in §6.

The energy heads under this solution x^*_{jk} are $y_1 = 4.8567 \approx y_1^{\min}$, $y_2 = 6.9344$, $y_3 = 6.9506$, i.e., y_1 is already at its lower bound (with a small numerical error). Actually, this is the reason to prevent the cost from reducing further.

The total cost is 1.7808×10^6 (DDK), which is slightly less than that reported in [5]—1.7940×10^6 (DDK) (the solution in [5] for the example is $x_{1,3} = x_{2,2} = x_{3,1} = x_{4,2} = x_{5,1} = 1$ and other $x_{i,j} = 0$).

Example 2.

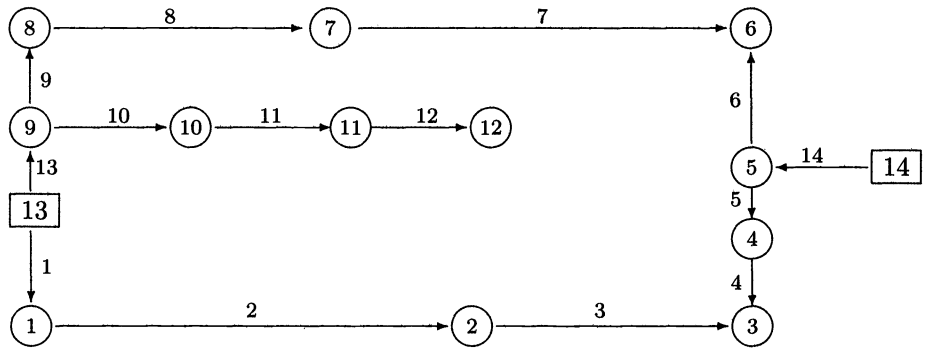


FIG. 9.2.

The data of this example is from a small town's gas network in which $m = 12$, $r = 2$, $n = 14$; see Figure 9.2. The network has $s = 6$ pipe sizes to choose; see Table 9.3.

TABLE 9.3.

k	D_k (10 cm)	c_k (HK\$/m)
1	1.0	1637
2	1.5	1796
3	2.0	2122
4	2.5	2908
5	3.0	2940
6	4.0	4139

The data for pipes is shown in Table 9.4.

The demands are

node i	1	2	3	4	5	6	7	8	9	10	11	12
b_i (m^3/h)	460	460	460	460	0	460	350	350	0	350	350	500

,

TABLE 9.4.

j	l_j (m)	i'_j	i''_j	initial k
1	90	13	1	5
2	120	1	2	5
3	110	2	3	5
4	130	4	3	4
5	20	5	4	4
6	90	5	6	4
7	180	7	6	6
8	55	8	7	6
9	70	9	8	6
10	20	9	10	6
11	70	10	11	5
12	60	11	12	5
13	120	13	9	5
14	25	14	5	4

where node 5 and node 9 are just transferring nodes. All $p_i^{\min} = 2.5$ (bar) for $i = 1, \ldots, 12$ and at the two supply nodes $\bar{p}_{13} = \bar{p}_{14} = 30$ (bar). σ, β, and K are the same as in Example 1.

Our computation results are

$$
\begin{aligned}
&x^*_{1,2} = 1, \\
&x^*_{2,1} = 0.79, && x^*_{2,2} = 0.21, \\
&x^*_{3,1} = 1, \\
&x^*_{4,1} = 0.887, && x^*_{4,2} = 0.113, \\
&x^*_{5,1} = 0.716, && x^*_{5,2} = 0.284, \\
&x^*_{6,2} = 0.89, && x^*_{6,3} = 0.11, \\
&x^*_{7,2} = 1, \\
&x^*_{8,2} = 1, \\
&x^*_{9,1} = 0.87, && x^*_{9,2} = 0.13, \\
&x^*_{10,2} = 1, \\
&x^*_{11,2} = 1, \\
&x^*_{12,2} = 0.233, && x^*_{12,3} = 0.767, \\
&x^*_{13,1} = 0.2471, && x^*_{13,2} = 0.7529, \\
&x^*_{14,2} = 0.89, && x^*_{14,3} = 0.11.
\end{aligned}
$$

All other $x^*_{jk} = 0$. The energy head $y_{12} = 6.25 = y_{12}^{\min}$, i.e., at the lower bound and all other $y_i > y_i^{\min}$ $(i = 1, \ldots, 11)$. (To save space, y_1–y_{11} are not listed). The total cost is HK\$2.03 $\times$ 10^6.

We calculated more test problems. But because we failed to find other reported results for the same type of gas network model, an extensive comparison with other methods is unavailable. However, our preliminary testing shows that this method is workable and competitive.

10. Discussion. The engineering factors that we took into account in previous sections are the same as those considered in [4]. Quite a few practical projects have more requirements. For example, some models also impose maximum pressure limits on each pipe arc. If we consider a simple case of constant maximum pressure prescribed for all the different diameter pipes, i.e., we have an additional set of constraints

$$y^{\max} \geq y$$

in problem (P2), then our algorithm does not need to change too much except that problem (S2) should contain an extra set of constraints:

$$y_i(z) + \nabla y_i^T(z)\,(v+w) \;\le\; y_i^{\max}, \quad i = 1, \ldots, m,$$

and in step 5 we also require

$$y(z^k + \bar{d}^k) \le y^{\max}.$$

Some pipe networks set compressors along certain arcs. If pipe j has a compressor which results in a head increase by τ_j (we use $\mathcal{J}$ to denote these pipes) and we assume that, by experience or by initial calculation, we know the actual flow direction in pipe j and assign it to the pipe as the positive direction, then for these arcs equation (2.3) should be revised to

$$K\, l_j\, d_j^{\beta}\, |q_j|^{\alpha-1}\, q_j - \tau_j = y_{i'_j} - y_{i''_j}, \qquad j \in \mathcal{J}. \tag{10.1}$$

It is easy to see that in this case, if we redefine function f_j as

$$f_j(q_j) \;\stackrel{\text{def}}{=}\; \frac{1}{1+\alpha} K\, l_j\, z_j\, |q_j|^{\alpha+1} + h_j\, q_j - \tau_j q_j, \quad j \in \mathcal{J},$$

i.e., add a term $-\tau_j q_j$ to the previous definition (4.2), then for any given z the solution $q(z)$ of equations (10.1) as well as (2.2) is still the solution of problem (L1). Since now $F(q) = \sum f_j(q_j)$ remains a convex function of q, it is very easy to make some minor revisions for the formulas of §§4 and 5. Note that in this case the solution q and its corresponding vector y for the lower level problem depend on both z and τ.

As for the upper level problem, if the cost of setting the compressor is governed by a function $\tilde{c}_j(\tau_j)$, then problem (U2) should be changed into

$$\text{(U3)} \qquad \begin{aligned} \min \quad & G(z) + \sum_{j \in \mathcal{J}} \tilde{c}_j(\tau_j) \\ \text{s.t.} \quad & y(z,\tau) \ge y^{\min}, \\ & D_s^{\beta} \le z_j \le D_1^{\beta}, \qquad j = 1, \ldots, n, \\ & 0 \le \tau_j \le \tau_j^{\max}, \qquad j \in \mathcal{J}, \end{aligned}$$

where $\tau_j^{\max}$ is the margin for the compressor's ability. Now there are two sets of decision variables z and τ in the upper level problem, and it is not difficult to make corresponding changes in our algorithm, assuming that function $\tilde{c}_j$ is differentiable and its particular form is given.

Another more challenging problem is that some models also restrict the maximum speed of gas flow in each pipe arc. We shall propose a method to meet this requirement in a forthcoming paper [20].

Acknowledgments. The authors are grateful to Professor U. Shamir at Technion, Israel, for his very helpful suggestions. They also thank Dr. M. L. Wong at the City University of Hong Kong for providing some test problems and Dr. X. Xu for joining them at the initial stage of this research.

REFERENCES

[1] U. SHAMIR, *Optimization in water distribution systems engineering*, Math. Programming Study, 11 (1979), pp. 65–84.

[2] A. KESSLER AND U. SHAMIR, *Decomposition technique for optimal design of water supply networks*, Engrg. Optim., 17 (1991), pp. 1–19.

[3] T. M. WALSKI, *State-of-the-Art Pipe Network Optimization*, American Society of Civil Engineering, Special Conference on Computer Applications in Water Resources, 1985, pp. 559–568.

[4] C. T. HANSEN, K. MADSEN, AND H. B. NIELSEN, *Optimization of pipe networks*, Math. Programming, 52 (1991), pp. 45–58.

[5] C. T. HANSEN, *Optimization of Large Networks for Natural Gas*, Ph.D. thesis (Report NI-88-09), Institute for Numerical Analysis, Technical University of Denmark, 1988.

[6] U. SHAMIR, *Optimal design and operation of water distribution systems*, Water Resources Res., 11 (1974), pp. 27–36.

[7] M. S. BAZARAA, J.J. JARRIS, AND H. D. SHERALI, *Linear Programming and Network Flows*, 2nd ed., John Wiley, New York, 1990.

[8] A. J. OSIADACZ, *Simulation and Analysis of Gas Networks*, E. & F.N. Spon, London, 1987.

[9] J. ZHANG, *An iterative method for nonlinear pipe network equations and its generalization*, Acta Math. Appl. Sinica, 4 (1981), pp. 175–189.

[10] T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[11] J. V. TIEL, *Convex Analysis*, John Wiley, New York, 1984.

[12] M. J. D. POWELL, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Gordon and Breach, London, 1970, pp. 87–114.

[13] D. C. SORENSEN, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[14] J. E. DENNIS JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[15] J. ZHANG, X. XU, AND D. ZHU, *A Nonmonotonic Trust Region Method for Unconstrained Optimization*, Research Report MA-93-07, City Polytechnic of Hong Kong, Hong Kong, 1993.

[16] I. ADLER, N. KARMARKAR, M. G. C. RESENDE, AND G. VEIGA, *An implementation of Karmarkar's algorithm for linear programming*, Math. Programming, 44 (1989), pp. 297–325.

[17] K. A. MCSHANE, C. L. MONMA, AND D. F. SHANNO, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Comput., 1 (1989), pp. 70–83.

[18] S. MEHROTRA, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[19] J. ZHANG AND D. ZHU, *A New Method for Pipe Network Optimization*, Research report MA-94-01, City Polytechnic of Hong Kong, Hong Kong, 1994.

[20] ———, *A Hierarchical Method for Optimal Dimensioning of Pipe Network Systems*, Research report MA-94-12, City Polytechnic of Hong Kong, Hong Kong, 1994.

SIAM J. OPTIMIZATION
Vol. 6, No. 3, pp. 858–877, August 1996
© 1996 Society for Industrial and Applied Mathematics
018

SOLVING MULTIPLE KNAPSACK PROBLEMS BY CUTTING PLANES*

C. E. FERREIRA†, A. MARTIN‡, AND R. WEISMANTEL‡

Abstract. In this paper we consider the multiple knapsack problem, which is defined as follows: given a set N of items with weights f_i, $i \in N$, a set M of knapsacks with capacities F_k, $k \in M$, and a profit function $c_{ik}, i \in N, k \in M$, find an assignment of a subset of the set of items to the set of knapsacks that yields minimum cost. We consider the multiple knapsack problem from a polyhedral point of view. The inequalities that we describe here serve as the theoretical basis for a cutting plane algorithm. We present some of the implementation details of this algorithm, including a discussion and evaluation of different separation and primal heuristics. Our algorithm is applied to practical problem instances arising in the design of main frame computers, in the layout of electronic circuits, and in sugar cane alcohol production.

Key words. multiple knapsack problem, cutting planes, branch and cut, separation

AMS subject classifications. 90, 68

1. Introduction.

Let a set N of *items* and a set M of *knapsacks* be given. With every item $i \in N$ there is associated a *weight* $f_i > 0$, and every knapsack $k \in M$ has a *capacity* $F_k > 0$. Moreover, an objective function $c_{ik} \in \mathbb{R}$, $i \in N, k \in M$ is given, which reflects the cost if item i is assigned to knapsack k. The task is to assign a subset of the set of items to the set of knapsacks that yields minimum cost. This problem is called the *weighted multiple knapsack problem.* In fact, we consider a slight generalization of the standard multiple knapsack problem in that the cost does not depend only on the items but also on the knapsacks.

The multiple knapsack problem is known to be $\mathcal{NP}$-hard in the strong sense [12], [13]. It has been extensively studied from an algorithmical point of view (see, for example, the excellent survey of Martello and Toth [13]). Unfortunately, the strategies that provide an exact solution for the problem have a prohibitive running time for instances of large scale. Our objective is to develop a cutting plane algorithm for the multiple knapsack problem and study its performance when applied to real world large scale instances.

The instances we have in mind arise in three different applications, namely, in the design of processors for main frame computers, in the layout of electronic circuits, and in sugar cane alcohol production in Brazil.

The first problem arises in the (global) design of a main frame computer. We are given a set of electronic components, whose size is typically between 200 and 1000. The most important property of the electronic circuits—for our purposes—is the area that these components cover. The electronic components must be integrated on printed circuit boards, multichip modules, or other devices. Each of these devices (whose number is usually between 5 and 10) is defined by several technical properties that we do not intend to describe here. Two properties of devices are important for us. Every device k has a *capacity* F_k, representing its "area" or the weight it can hold, and a *cut capacity* S_k, describing the number of wires that can be connected to

* Received by the editors August 19, 1993; accepted for publication (in revised form) January 25, 1995. This research was partially supported by Science Program SC1-CT91-620 of the EEC and by the DAAD in connection with the PROBRAL 1994 project for a scientific exchange with Brazil.

† Computer Science Department, Universidade de São Paulo, São Paulo, Brazil.

‡ Konrad-Zuse-Zentrum fur Informationstechnik Berlin, Heilbronner Strasse 10, D-10711 Berlin-Wilmersdorf, Germany (weismantel@zib-berlin.de).

this device. The electronic components have certain contact points, called pins, from which wires can extend to pins of other components. In the logical design phase it is determined which pins of which components must be connected by a wire to ensure certain functional properties. It is customary to call a collection of pins that must be connected a net.

The task is to assign the electronic components to the devices in such a way that a certain objective function is minimized and a number of technical side constraints are satisfied. Among them are two essential requirements.

- For each device k the sum of the areas of the electronic components that are assigned to this device must not exceed the capacity F_k.
- The number of nets that must leave some device k must not exceed its cut capacity S_k.

The mathematical problem that arises by thoroughly modelling all (or at least the most important) aspects of this question is a rather complicated integer program. The full model appears to be hopelessly difficult—at least for the present state of integer programming technology. Thus, we investigated a hierarchy of combinatorial relaxations of the complete model. A first relaxation of the general model is the multiple knapsack problem, where we neglect the nets completely and concentrate on the packing aspect of the problem. In knapsack terminology, the items correspond to the electronic components and the devices to the knapsacks. An appropriate objective function for the multiple knapsack problem is determined heuristically. For more details we refer the reader to [7].

The second application we have in mind arises in the layout of electronic circuits. Here, one major subproblem (the so-called placement problem) is to assign a set of logical units (cells) to locations on a given rectangle (silicon) subject to certain technical side constraints. In general, cells are of rectangular shape and have a particular weight representing their area. Because of the inherent complexity of the placement problem and its large scale (typically up to several thousands of cells must be assigned), it is further decomposed in practice. In a first step, the given rectangle is subdivided and it is determined which cells are assigned to which of the subareas such that the total weight of the cells that are assigned to the same subarea does not exceed the corresponding area capacity. This process of iteratively subdividing areas and assigning cells to subareas is continued until every subarea contains at most one cell. If we interpret cells as items and subareas as knapsacks, we can associate with every problem arising in this decomposition scheme a multiple knapsack problem. In fact, the multiple knapsack problem does not reflect the whole situation, since—besides the area requirements—there are many additional side constraints which must be taken into account and a complicated objective function that strongly depends on the underlying fabrication technology must be minimized. Nevertheless, solving the corresponding multiple knapsack problems seems to be a reasonable starting point to attack the much more complicated placement problems.

The third application arises in sugar cane alcohol production in Brazil. Since the late 70s, sugar cane alcohol has been used in Brazil as fuel for cars and plays an important role in Brazilian agriculture. Farmers sell their sugar cane to alcohol producers. The producers themselves collect the sugar cane from the farmers and deliver it to one of their factories. Each plantation i produces a certain amount f_i of sugar cane, and each factory k is able to treat a maximum amount F_k of sugar cane. The problem, from the side of the alcohol producer, is now to decide which plantation production to deliver to which factory such that the capacity constraints are satisfied

and the transportation cost is minimized. The transportation cost of delivering the production of plantation i to factory k is proportional to the production of the plantation and the distance between the plantation and the factory. We obtained two real data instances of this problem. In both instances the plantations must be distributed to four factories. In the first instance there are 450 plantations while in the second one there are 370.

In this paper we consider the multiple knapsack problem from a polyhedral point of view. Let an *instance* (N, M, f, F) *of the multiple knapsack problem* be given, i.e., a set N of items, a set M of knapsacks, a weight vector $f = (f_i)_{i \in N}$, and a capacity vector $F = (F_k)_{k \in M}$. We introduce variables $x_{ik} \in \mathbb{R}^{N \times M}$ with the interpretation $x_{ik} = 1$ if item i is assigned to knapsack k, and $x_{ik} = 0$ otherwise. The *multiple knapsack polytope* $MK(N \times M, f, F)$ is defined as the convex hull of all feasible solutions to the multiple knapsack problem. It is easy to see that the following relation holds.

$$\begin{aligned} MK(N \times M, f, F) = \operatorname{conv}\{x \in \mathbb{R}^{N \times M} \mid & \\ \textstyle\sum_{i \in N} f_i x_{ik} \le F_k & \quad \text{for all } k \in M, \quad (1) \\ \textstyle\sum_{k \in M} x_{ik} \le 1 & \quad \text{for all } i \in N, \quad (2) \\ x_{ik} \in \{0,1\} & \quad \text{for all } i \in N, k \in M\}. \end{aligned}$$

Constraints (1) are called *knapsack constraints* and constraints (2) *SOS (special ordered set) constraints.* The weighted multiple knapsack problem can be solved, in principle, via the following linear program:

$$\begin{aligned} \min \quad & \textstyle\sum_{i \in N} \sum_{k \in M} c_{ik} x_{ik}, \\ & x \in MK(N \times M, f, F). \end{aligned} \tag{1.1}$$

In order to apply linear programming techniques we need a description of the polytope $MK(N \times M, f, F)$ by means of inequalities. This issue is addressed in §§2 and 3. Since the number of facet-defining inequalities is exponential in the size of the input, we use a cutting plane based algorithm to solve (1.1). Here, the idea is the following.

Start with a small set of valid inequalities, for example, the SOS and knapsack inequalities. These inequalities define a polyhedron P' that contains $MK(N \times M, f, F)$. Optimize the linear objective function over P' and let y be an optimal solution. Obviously, y yields a lower bound for the optimum value of the weighted multiple knapsack problem. If y is also feasible, y is an optimal solution for the weighted multiple knapsack problem. Otherwise, there exists a valid inequality for $MK(N \times M, f, F)$ that is violated by y. Thus, we must solve the *separation problem*, which involves finding a valid inequality that is violated by y. If such an inequality is found, we add it to the linear program and solve it again. The procedure of iteratively solving linear programs and adding violated constraints is commonly called a *cutting plane algorithm.*

A cutting plane algorithm ends with an optimal solution or (at least) with a lower bound for the weighted multiple knapsack problem. The latter case is unavoidable with our algorithm, since the cutting planes we use do not describe the multiple knapsack polytope completely. In order to guarantee that our solution is optimal we embed the procedure into an enumeration scheme.

In this paper we discuss how to adapt the general cutting plane method to the weighted multiple knapsack problem. In detail, the paper is organized as follows. In §2 we report on some basic results known for the multiple knapsack polytope and related polyhedra and make some easy observations. In §3 we describe so-called joint

inequalities, inequalities that combine at least two knapsacks. Section 4 is devoted to the separation problem for the described classes of inequalities. We present several heuristic procedures, including lifting and complementing, to find violated inequalities. In §5 we deal with some important details of our implementation. In particular, primal heuristics for finding a good feasible solution for the multiple knapsack problem are presented. We have tested our cutting plane based algorithm on instances arising in the applications previously mentioned. The computational results we have obtained are shown in §6.

2. The multiple knapsack polytope: Some observations. Let an instance (N, M, f, F) of the multiple knapsack problem be given. It will turn out that we often refer to subinstances of the problem where certain items are not feasible for certain knapsacks. Thus, we define the polyhedron in a more general frame. For $T \subseteq N \times M$, let $A := \{i \in N \mid \text{there exists a knapsack } k \in M \text{ with } (i,k) \in T\}$ and $B := \{k \in M \mid \text{there exists an item } i \text{ such that } (i,k) \in T\}$.We define the *multiple knapsack polytope* by

$$MK(T,f,F) := \operatorname{conv}\{x \in \mathbb{R}^T \mid \begin{array}{ll} \sum_{i:(i,k)\in T} f_i x_{ik} \le F_k, & k \in B, \\ \sum_{k:(i,k)\in T} x_{ik} \le 1, & i \in A, \\ x_{ik} \in \{0,1\}, & (i,k) \in T\}. \end{array}$$

The polytope corresponding to the multiple knapsack problem defined in §1 coincides with $MK(N \times M, f, F)$, which we often abbreviate by MK. It is easy to see that MK is full dimensional if and only if $f_i \le F_k$ for all $i \in N$ and $k \in M$. Similarly, the dimension of the polytope $MK(T,f,F)$ equals $|T|$ if and only if $f_i \le F_k$ for all $(i,k) \in T$. In the following we assume that $f_i \le F_k$ for all $(i,k) \in T$. The *trivial inequalities* $x_{ik} \ge 0$ define facets of MK for all $i \in N, k \in M$. In case $m \ge 2$, the SOS constraints define facets of MK.

The multiple knapsack problem is a canonical generalization of the single 0/1 knapsack problem, where $|M| = 1$. In analogy to the definition of MK let

$$SK(N,f,F) := \operatorname{conv}\left\{x \in \mathbb{R}^N \mid \sum_{i\in N} f_i x_i \le F, x_i \in \{0,1\}, i \in N\right\}$$

denote the *single knapsack polytope.*

Although a lot of emphasis has been placed on studying the facial structure of $SK(N,f,F)$ (see, for example, [1], [16], [11], [14], [2], [15]), MK and generalizations of it have not yet been studied to the same extent. In a few papers we find investigations in this direction. Crowder, Johnson, and Padberg [5] consider general 0/1 linear programs with no apparent structure: Let be given a matrix $A \in \mathbb{Q}^{m\times n}$, $a_{ki} \ge 0$ for all $i = 1,\dots,n, k = 1,\dots,m$, a vector $b \in \mathbb{Q}^m$, and define $IP := \operatorname{conv}\{x \in \{0,1\}^n \mid Ax \le b\}$. The authors associate with each constraint k the single knapsack polytope $SK_k := SK(\{1,\dots,n\}, (a_{k1},\dots,a_{kn})^T, b_k)$. Clearly, $IP \subseteq \bigcap_{k=1}^m SK_k$. In [5] large scale 0/1 linear programs are solved by using single knapsack inequalities for the polytopes SK_k in order to chop off fractional solutions that are obtained during the run of a cutting plane algorithm. Gottlieb and Rao [9], [10] study the *generalized assignment problem*, a generalization of the multiple knapsack problem, where every item i may have "a particular weight f_{ik}" for each knapsack k.

Obviously, for every $k \in M$ the relation $MK(N \times M, f, F) \subseteq SK(N, f, F_k)$ holds and, hence, the question arises whether knowledge about these single knapsack polyhedra can be used to describe the corresponding multiple knapsack polytope.

Indeed, the answer to this question is yes. All nontrivial facet-defining inequalities associated with the single knapsack polytopes are inherited by MK (cf. [9]). More precisely, let $V \subseteq N$ and $k \in M$ and suppose $a^T x \leq \alpha$ is a nontrivial facet-defining inequality of $SK(V, f, F_k)$. Then $\overline{a}^T x \leq \alpha$ defines a facet of $MK(V \times M, f, F)$, where $\overline{a} \in \mathbb{R}^{V \times M}$ with $a_{il} := a_i$ if $l = k$ and $a_{il} := 0$; otherwise, for $i \in V$.

We call a valid inequality $\overline{a}^T x \leq \alpha$ for $MK(N \times M, f, F)$ *individual* if there is some $k \in M$ such that $\overline{a}_{il} = 0$ for all $l \in M \setminus \{k\}$. Otherwise, we call the inequality *joint*. Examples of individual inequalities are the minimal cover inequality and the (1,d)-configuration inequality that we want to present now.

A set $S \subseteq N$ is a *cover* with respect to some knapsack $k \in M$ if $\sum_{i \in S} f_i > F_k$. The cover is *minimal* with respect to k if $\sum_{i \in S \setminus \{s\}} f_i \leq F_k$ for all $s \in S$. Suppose that $S \subseteq N$ is a minimal cover with respect to some knapsack $k \in M$. The inequality

$$\sum_{i \in S} x_{ik} \leq |S| - 1$$

is called a *minimal cover inequality* corresponding to S and k. In [1], [16], and [11] it was shown that the minimal cover inequality corresponding to S and k defines a facet of $SK(S, f, F_k)$ (and, thus, of $MK(S \times M, f, F)$).

Another well-known class of individual inequalities consists of the (1,d)-configuration inequalities. A set $N' \cup \{z\}$ with $N' \subseteq N$ and $z \in N \setminus N'$ is called a $(1, d)$-*configuration* with respect to some knapsack $k \in M$ if $\sum_{j \in N'} f_j \leq F_k$ and $K \cup \{z\}$ is a minimal cover with respect to knapsack k for all $K \subset N'$ with $|K| = d$. Suppose now that $N' \cup \{z\} \subseteq N$ is a (1,d)-configuration with respect to some knapsack $k \in M$. The inequality

$$\sum_{i \in N'} x_{ik} + (|N'| - d + 1) x_{zk} \leq |N'|$$

is called a $(1, d)$-*configuration inequality* corresponding to $N' \cup \{z\}$ and k. In [15] it was shown that the (1,d)-configuration inequality corresponding to $N' \cup \{z\}$ and k defines a facet of $SK(N' \cup \{z\}, f, F_k)$ (and, therefore, of $MK((N' \cup \{z\}) \times M, f, F)$).

In addition to individual inequalities there are classes of joint inequalities that are valid for the multiple knapsack polytope. In the following section we present some of the joint inequalities we have found for the multiple knapsack polytope.

3. Joint inequalities. Before describing joint inequalities, we introduce some notation. Let $e_i \in \mathbb{R}^N$ be the unit vector with a value of one in the ith component and zero otherwise. We denote by e_{ik} the unit vector in the vector space $\mathbb{R}^{N \times M}$. For $x \in \mathbb{R}^N$ and $I \subseteq N$, we set $x(I) := \sum_{i \in I} x_i$. For $I \subseteq N$, we denote by χ^I the *incidence vector* of I in $\mathbb{R}^N$, i.e., $\chi^I_i = 1$ if $i \in I$ and $\chi^I_i = 0$ otherwise. For a vector $x' \in \{0,1\}^{N \times M}$ and some knapsack $k \in M$, we define $B_k(x') := \{i \in N \mid x'_{ik} = 1\}$. If $a^T x \leq \alpha$ is a valid inequality for some polytope P, we set $EQ(P, a^T x \leq \alpha) := \{x' \in P \mid a^T x' = \alpha\}$. In order to simplify notation, we frequently abbreviate $EQ(P, a^T x \leq \alpha)$ by $EQ(a^T x \leq \alpha)$ if there is no possibility of confusion.

In the following we describe a selection of joint inequalities we found for the multiple knapsack polytope. These inequalities will be used in our cutting plane algorithm for solving practical problem instances. We refer the reader interested in further (facet-defining) inequalities to [6].

The multiple cover inequality. In [17], it was observed that, given a set $S \subseteq N$ and $J \subseteq M$ with $f(S) > F(J)$, the inequality

$$\sum_{j \in J} \sum_{i \in S} x_{ij} \leq |S| - 1$$

is valid for the polytope MK. If $|J| \geq 2$ a set of items S with the property $f(S) > F(J)$ is called a *multiple cover* with respect to J and the corresponding inequality a *multiple cover inequality.* A set of items S is called a *minimal multiple cover* with respect to J if $f(S) > F(J)$ and, for all $s \in S$, there exists a valid assignment of all items in $S \setminus \{s\}$ to the knapsacks in J.

One can easily convince oneself that the multiple cover inequality does not always define a facet of MK. Thus, the following question arises: What are (necessary and sufficient) conditions under which the multiple knapsack inequality defines a facet? We have investigated this question and it turns out that, in general, these conditions are rather complicated and involve many (probably) unavoidable technicalities. We do not present the details here. However, the conditions simplify if we consider the special case of the multiple knapsack problem where the knapsack capacity values are all equal.

THEOREM 3.1. *Let (N, M, f, F) be an instance of the multiple knapsack problem where $F_k = F_l$ for all $k, l \in M$. Let $S \subseteq N$ be a minimal multiple cover for some $J \subseteq M$. Then the multiple cover inequality*

$$\sum_{i \in S} \sum_{j \in J} x_{ij} \leq |S| - 1$$

defines a facet of $MK(S \times J, f, F)$ if and only if there exists an item $i \in S$ and a valid assignment x' of the items in $S \setminus \{i\}$ to the knapsacks in J such that $|B_k(x')| \neq |B_l(x')|$ for some $k, l \in J, k \neq l$.

Proof. We first prove that the condition is sufficient. Set $a := \sum_{i \in S} \sum_{j \in J} e_{ij}$ and $\alpha := |S| - 1$. The inequality $a^T x \leq \alpha$ is clearly valid. Let us prove that it defines a facet of $MK(S \times J, f, F)$. Suppose that $b^T x \leq \beta$ defines a facet of $MK(S \times J, f, F)$ such that $EQ(a^T x \leq \alpha) \subseteq EQ(b^T x \leq \beta)$. Let i_0 be an index such that $f_{i_0} = \min\{f_i \mid i \in S\}$ and let x^1 denote a valid assignment of the items in $S \setminus \{i_0\}$ to the knapsacks in J. Obviously, x^1 is in $EQ(a^T x \leq \alpha)$. Also, notice that for all $k \in J$ and $i \in B_k(x^1)$, the vector $x^1 - e_{ik} + e_{i_0 k}$ is an element of $EQ(a^T x \leq \alpha)$. Thus, $b^T x^1 = b^T(x^1 - e_{ik} + e_{i_0 k})$, yielding $b_{ik} = b_{i_0 k}$. Moreover, since the capacities of the knapsacks are all equal, we can exchange the items of every pair of knapsacks and repeat the same arguments as above. Summing up, we conclude that, for every $k \in J$, there exists a constant c_k such that $b_{ik} = c_k$ for all $i \in S$.

In order to prove that $c_k = c_l$ for $k \neq l, k, l \in J$, let $i \in S$ be an item and let x' denote a valid assignment of the items in $S \setminus \{i\}$ to the knapsacks in J such that $|B_k(x')| \neq |B_l(x')|$ for some $k, l \in J, k \neq l$, as required in the condition. Since all knapsacks have the same capacity, we can construct a valid assignment $x'' = (x''_{ij})$ via

$$x''_{ij} := \begin{cases} x'_{il} & \text{for all } i \in S, j = k, \\ x'_{ik} & \text{for all } i \in S, j = l, \\ x'_{ij} & \text{otherwise.} \end{cases}$$

Clearly, x' and x'' belong to the face $EQ(a^T x \leq \alpha)$. Thus, $b^T x' = b^T x''$, yielding

$$|B_k(x')| c_k + |B_l(x')| c_l = |B_k(x')| c_l + |B_l(x')| c_k.$$

This implies $c_k = c_l$. Because of the uniform knapsack capacities, we can apply this construction for all other knapsacks and finally obtain that $b^T x \leq \beta$ and $a^T x \leq \alpha$ are equal up to multiplication with a scalar, which completes the first part of the proof.

It remains to be shown that the condition is necessary. Suppose it is not satisfied, i.e., for all $x \in EQ(a^T x \leq \alpha)$ and $k, l \in J, k \neq l$, $|B_k(x)| = |B_l(x)|$ holds. In this case, all $x \in EQ(a^T x \leq \alpha)$ satisfy the equation $\sum_{i \in S} x_{ik} = (|S| - 1)/|J|$ for all $k \in J$. Thus, the inequality cannot be facet defining. □

Extension of facet-defining inequalities. In this subsection we present a general procedure that allows the extension of particular classes of inequalities. As we will see, this procedure can be iteratively applied, yielding many new valid and facet-defining inequalities.

THEOREM 3.2. *Let be given an instance (N, M, f, F) of the multiple knapsack problem, sets $A \subseteq N$ and $B \subseteq M$, and an inequality $a^T y \leq \alpha$ that is facet defining for $MK(A \times B, f, F)$ and that satisfies the following additional requirement.*

($\star$) For all $\tilde{A} \subseteq A$ with $|\tilde{A}| \geq 2$ the following holds: every assignment $y \in MK(A \times B, f, F)$ with $y_{ik} = 0$ for all $i \in \tilde{A}$, $k \in B$ satisfies $a^T y \leq \alpha - |\tilde{A}| + 1$.

We choose a positive integer $r \leq \min\{|N \setminus A|, |M \setminus B|\}$, nonempty sets $T_1, \ldots, T_r$ that are mutually disjoint subsets of $N \setminus A$, and a subset $\{k_1, \ldots, k_r\}$ of $M \setminus B$. Let us further define $\beta_v := \max\{|G| \mid G \subseteq T_v, f(G) \leq F_{k_v}\}, v = 1, \ldots, r$. Under these assumptions, the inequality

$$a^T x + \sum_{v=1}^{r} \sum_{i \in A \cup T_v} x_{ik_v} \leq \alpha + \sum_{v=1}^{r} \beta_v \tag{3.1}$$

is valid for the polytope $MK(A \times B \cup (\bigcup_{v=1}^{r} (T_v \cup A) \times \{k_v\}), f, F)$ if and only if $\tilde{T}_v \cup \{i\}$ is a cover with respect to k_v for all $i \in A$, $\tilde{T}_v \subseteq T_v$, $|\tilde{T}_v| = \beta_v$, and $v = 1, \ldots, r$.

Proof. For ease of notation let us refer to inequality (3.1) by $b^T x \leq \beta$ and set $Q := A \times B$.

We first prove that the condition is necessary. Suppose there exists an index $v \in \{1, \ldots, r\}$, an item $i_0 \in A$, and a set $\tilde{T}_v \subseteq T_v$, $|\tilde{T}_v| = \beta_v$ such that $\tilde{T}_v \cup \{i_0\}$ is not a cover with respect to k_v. Let $\tilde{T}_w \subseteq T_w, w \in \{1, \ldots, r\} \setminus \{v\}$, $|\tilde{T}_w| = \beta_w$, $f(\tilde{T}_w) \leq F_{k_w}$.

Since $a^T y \leq \alpha$ defines a facet, there exists an assignment $y' \in EQ(MK(Q, f, F), a^T y \leq \alpha)$ with $y'_{i_0 k} = 0$ for all $k \in B$. Set $x' = (x'_{ik})$ via

$$x'_{ik} := \begin{cases} y'_{ik} & \text{for all } (i,k) \in Q, \\ 1 & \text{for all } i \in \tilde{T}_w, k = k_w, w \in \{1, \ldots, r\} \setminus \{v\}, \\ 1 & \text{for all } i \in \tilde{T}_v, k = k_v, \\ 0 & \text{otherwise.} \end{cases}$$

Since $\tilde{T}_v \cup \{i_0\}$ does not define a cover with respect to knapsack k_v, $x' + e_{i_0 k_v}$ is a valid assignment yielding $b^T(x' + e_{i_0 k_v}) = a^T y' + \sum_{w \in \{1,\ldots,r\} \setminus \{v\}} \beta_w + \beta_v + 1 > \alpha + \sum_{v=1}^{r} \beta_v$. This implies that the condition is necessary, indeed.

In order to prove the converse direction, let us assume that the inequality is not valid for the polytope $MK(Q \cup \bigcup_{v=1}^{r} (T_v \cup A) \times \{k_v\}, f, F)$, i.e., there exists an assignment $x \in MK(Q \cup \bigcup_{v=1}^{r} ((T_v \cup A) \times \{k_v\}), f, F)$ with $b^T x > \beta$. Set $\tilde{T}_v := \{i \in T_v \mid x_{ik_v} = 1\}$ and $A_v := \{i \in A \mid x_{ik_v} = 1\}, v = 1, \ldots, r$. Since the inequality $a^T y \leq \alpha$ holds for all $y \in MK(Q, f, F)$, there exists some $v \in \{1, \ldots, r\}$ satisfying $|A_v \cup \tilde{T}_v| > \beta_v$. Let $V \subseteq \{1, \ldots, r\}$ denote the subset of knapsacks with $|A_v \cup \tilde{T}_v| > \beta_v$

for all $v \in V$. Because of the condition that every subset of T_v of cardinality β_v and one element from A defines a cover with respect to k_v ($v \in V$), we have that $|\tilde{T}_v| \leq \beta_v - 1$, thus forcing A_v to be greater than or equal to two, which holds for all $v \in V$. Moreover, $A_v \cap A_w = \emptyset$ for all $v, w \in V, v \neq w$. Summing up, we obtain with ($\star$)

$$\begin{aligned} b^T x &\leq \alpha - \sum_{v \in V} |A_v| + 1 + \sum_{v \in \{1,\ldots,r\} \setminus V} \beta_v + \sum_{v \in V} (|\tilde{T}_v| + |A_v|) \\ &\leq \alpha - \sum_{v \in V} |A_v| + 1 + \sum_{v \in \{1,\ldots,r\} \setminus V} \beta_v + \sum_{v \in V} (\beta_v + |A_v|) - |V| \\ &= \beta + 1 - |V| \leq \beta. \end{aligned}$$

This contradicts the assumption that x is a point violating the inequality. Thus, the inequality is valid, which completes the proof. □

If we strengthen the requirements and conditions of Theorem 3.2, the extended inequality will also be facet defining. The subsequent theorem (which we give without a proof) states necessary and sufficient conditions for the extended inequality to be facet defining.

THEOREM 3.3. *Under the assumptions of Theorem* 3.2 *and the condition that* $T_v \cup \{i\}$ *is a minimal cover with respect to knapsack* k_v *for all* $i \in A$ *and* $v = 1, \ldots, r$, *the inequality*

$$a^T x + \sum_{v=1}^{r} \sum_{i \in A \cup T_v} x_{ik_v} \leq \alpha + \sum_{v=1}^{r} |T_v| \tag{3.2}$$

defines a facet for $MK(A \times B \cup \bigcup_{v=1}^{r} (T_v \cup A) \times \{k_v\}, f, F)$ *if and only if, for every* $v \in \{1, \ldots, r\}$, *there exist a subset* $\tilde{A} \subseteq A$ *with* $|\tilde{A}| \geq 2$, *an item* $t_v \in T_v$, *and an assignment* $y \in MK(A \times B, f, F)$ *such that* $a^T y = \alpha - |\tilde{A}| + 1$, $y_{ik} = 0$ *for all* $i \in \tilde{A}$, $k \in B$, *and* $f(T_v \setminus \{t_v\}) + f(\tilde{A}) \leq F_{k_v}$. □

Theorems 3.2 and 3.3 can easily be generalized by using instead of $A \times B$ any arbitrary "starting set" $T \subseteq N \times M$. At first sight, the requirement ($\star$) seems to be quite restrictive, but it turns out that many inequalities satisfy this condition. For example, the minimal cover and multiple cover inequalities are such inequalities. Moreover, when we apply Theorem 3.3 to the minimal cover or multiple cover inequality, respectively, the resulting extended inequality still satisfies requirement ($\star$). Thus, a repeated extension in the "spirit" of Theorem 3.3 is possible. Moreover, one can convince oneself that, for example, the r-fold repetition of this "extension procedure" (i.e., each time we extend the original inequality by one set of items T_v and one knapsack k_v) leads to an inequality that is different from the one obtained if the simultaneous extension by the sets $T_1, \ldots, T_r$ and knapsacks $k_1, \ldots, k_r$ is applied to the original inequality.

4. Separation. In this section we deal with the separation problem for the classes of inequalities presented in §§2 and 3. Formally, the separation problem for a given class of inequalities can be stated as follows.

> Given an instance (N, M, f, F) of the multiple knapsack problem, a vector $x' \in [0,1]^{N \times M}$, and a class of valid inequalities for $MK(N \times M, f, F)$, decide whether x' satisfies all inequalities of the given class. If not, find an inequality of that class which x' violates.

All the classes of inequalities shown in the previous section are generalizations of cover inequalities. Thus, we consider the separation problem for the minimal cover inequalities first. Obviously, the problem of finding a minimum (with respect to some weighting of the items) cover is $\mathcal{NP}$-hard because it reduces to the single knapsack problem. It also turns out that the separation problem for the minimal cover inequality reduces to the single knapsack problem, implying that it is $\mathcal{NP}$-hard (see [6]). Although this result does not prove that all the other separation problems are $\mathcal{NP}$-hard as well, we strongly conjecture that they are. Thus, we put our emphasis on developing separation heuristics, which we will discuss in the following. We conclude this section by describing the main ideas of lifting and complementing inequalities.

Separating minimal cover inequalities. In [3] a pseudopolynomial algorithm was presented that solves the separation problem for minimal cover inequalities. The algorithm is based on a transformation of the problem into a network flow problem with a pseudopolynomial number of vertices and edges. By applying network flow techniques, the problem can be solved in pseudopolynomial time and space complexity.

In our code we have integrated a heuristic to separate minimal cover inequalities which was introduced in [5]. This procedure can be described as follows.

Separation heuristic for minimal cover inequalities
Choose a knapsack $k \in M$.
Solve the following linear program:

$$\begin{array}{ll} \min & \sum_{i \in N}(1 - x'_{ik})\tilde{s}_i \\ \text{s.t.} & \sum_{i \in N} f_i \tilde{s}_i \geq F_k + 1, \\ & 0 \leq \tilde{s}_i \leq 1 \text{ for all } i \in N. \end{array}$$

Set $S := \{i \in N \mid \tilde{s}_i > 0\}$.
Reduce S to a minimal cover.
Lift the corresponding inequality to a facet of $SK(N, f, F_k)$.
If the resulting inequality is violated add it to the list of violated inequalities.

We store all minimal covers we obtain by applying this procedure in a certain structure, called "pool." We use these minimal covers as substructures for some other separation routines (see the next subsection). Note that the minimal cover S found by the heuristic defines a facet for the polytope $SK(S, f, F_k)$, but not necessarily for $SK(N, f, F_k)$. In order to obtain a facet-defining inequality for $SK(N, f, F_k)$ (and thus for $MK(N \times M, f, F)$) the inequality must be lifted. This can be done in polynomial time, as we will see in the last subsection.

The linear program that appears in the heuristic can be solved efficiently by using Dantzig's procedure. For ease of exposition suppose that $N = \{1, \ldots, n\}$ and that $(1 - x'_{1k})/f_1 \leq \cdots \leq (1 - x'_{nk})/f_n$. Set $c := \min\{i \in \{1, \ldots, n\} \mid \sum_{j=1}^{i} f_j > F_k\}$. Then the solution of the linear program is given by $s^* = (s_i^*)_{i \in N}$, where

$$s_i^* = \begin{cases} 1, & 1 \leq i \leq c-1, \\ 0, & c+1 \leq i \leq n, \\ \frac{F_k + 1 - \sum_{j=1}^{c-1} f_j}{f_c}, & i = c. \end{cases}$$

Separating $(1, d)$-configuration inequalities. In order to find violated $(1, d)$-configuration inequalities we have been implementing and evaluating two heuristics. One of these was introduced in [5] and can be described as follows.

Heuristic 1
Choose a minimal cover S for some knapsack k from the pool.
Let $z \in S$ be an item with maximum weight.
Set $N' := S \setminus \{z\}$ and $d := |N'|$.
For every item $i \in N \setminus S$ perform the following steps
if $\sum_{j \in N' \cup \{i\}} f_j \leq F_k$ and if for all $K \subseteq N' \cup \{i\}$ with $|K| = d$,
the set $K \cup \{z\}$ is a cover with respect to k,
set $N' := N' \cup \{i\}$,
lift the corresponding $(1,d)$-configuration inequality, and
check whether it is violated.

The second heuristic for finding $(1, d)$-configuration inequalities uses a threshold parameter t and partitions the set of items N into two sets L and S. For the items in S the corresponding weight is smaller than or equal to t and the items in L have weight greater than t. The special element of the $(1, d)$-configuration is now chosen among the items in L and the other elements in the $(1, d)$-configuration are selected in S by solving a linear program. More precisely, the algorithm can be described as follows.

Heuristic 2
Set $L := \{i \in N \mid f_i > t\}$ and $S := \{i \in N \mid f_i \leq t\}$.
Choose a knapsack $k \in M$.
Let $z \in L$ be an item such that $x'_{zk} = \max\{x'_{ik} \mid i \in L\}$.
Solve the following linear program:

$$\begin{array}{ll} \max & \sum_{i \in S} x'_{ik} s_i \\ \text{s.t.} & \sum_{i \in S} f_i s_i \leq F_k, \\ & 0 \leq s_i \leq 1 \text{ for all } i \in S. \end{array}$$

Set $N' := \{i \in S \mid s_i = 1\}$.
If $N' \cup \{z\}$ is a $(1, d)$-configuration,
lift the corresponding inequality and check whether it is violated.

TABLE 1
Comparison of heuristics for $(1, d)$-configuration separation.

Problem	Heuristic 1	Heuristic 2
cl2 0% red.	3	104
cl2 4% red.	4	148
dm1 36.8% red.	1	9
dm2 30 % red.	0	22
sa2	2	95

Table 1 subsumes the performance of both heuristics on some problem instances (see §6 for more details on these instances). In column 2 the total number of violated inequalities found by the first procedure is given. Accordingly, the numbers shown in column 3 correspond to the number of violated $(1, d)$-configuration inequalities that were obtained by applying Heuristic 2. On all these examples the second heuristic performs much better than Heuristic 1 does. Our explanation is that Heuristic 1 does not take into account the current fractional point, but starts with a cover that was stored in the pool. However, minimal covers in the pool correspond to inequalities that were violated in some previous iteration, but are mostly satisfied by the current fractional point. Hence, choosing one item from the cover as the special element of the

configuration and adding some other items that do not belong to the cover may not give rise to a violated inequality. In contrast, the performance of Heuristic 2 depends only on the fractional point x' and the parameter t. In our implementation we sort the items in increasing order according to their weights, without loss of generality $N = \{1, \ldots, n\}$, $f_1 \leq f_2 \leq \cdots \leq f_{|N|}$, and run the heuristic with all possible sets L and S such that $S = \{1, \ldots, i\}$, $L = \{i+1, \ldots, |N|\}$, and $f_i < f_{i+1}$.

Separating multiple cover inequalities. Our procedure for separating multiple cover inequalities is a two-stage process. First, we determine a set $\mathcal{M}$ of candidate sets $M' \subseteq M$. For each of these sets $M' \in \mathcal{M}$ we then try to determine a set of items that defines a multiple cover. The latter problem is solved, in principle, by applying the above minimal cover separation routine to the "aggregated" knapsack with capacity $F(M')$. More precisely, suppose $x' \in \mathbb{R}^{N \times M}$ is the current fractional point and $M' \subseteq M$ is the set of knapsacks for which we want to find a multiple cover. We define the instance (N, f, C) of the single knapsack problem by setting $C := F(M')$. Moreover, we define a new fractional point, $y \in \mathbb{R}^N$ say, where $y_i := \sum_{k \in M'} x'_{ik}$ for all $i \in N$. With input (N, f, C) and y, the routines for finding minimal cover inequalities are called. If these calls yield a violated minimal cover inequality, it is transformed back to the original space and, hence, corresponds to a violated multiple cover inequality.

Unfortunately, there are $2^{|M|} - (|M| + 2)$ different sets of knapsacks for which we could perform the algorithm just described. This would, even for small $|M|$, result in an immense running time. Therefore, we generate a set $\mathcal{M}$ of candidates $M' \subseteq M$ according to some heuristic rules which we briefly explain now. For every item $i \in N$ we determine $B_i := \{k \in M \mid x'_{ik} > 0\}$, where x' is the current LP solution. If $2 \leq |B_i| \leq |M| - 1$, we consider B_i as one candidate set for which we try to find a multiple cover, as described before. In case $B_i = M$, we simply add all subsets of M of cardinality $|M| - 1$ to the set $\mathcal{M}$ of candidate sets. The idea for this (heuristic) selection of the candidate sets is that B_i is a subset of knapsacks where still (at least) one item is in conflict with its "correct position." Thus, additional inequalities that provide further information about how to handle this conflict are needed. Proceeding this way, the number of different candidate sets that are generated does not exceed the number $|N| + |M| - 1$.

Separating extended cover inequalities. Let S be a cover with respect to some knapsack k and let $l \in M \setminus \{k\}$ and $T \subseteq N \setminus S$. Obviously, the minimal cover inequality $\sum_{i \in S} x_{ik} \leq |S| - 1$ satisfies property $(\star)$ of Theorem 3.2. Thus, due to Theorem 3.2, the inequality

$$\sum_{i \in S} (x_{ik} + x_{il}) + \sum_{i \in T} x_{il} \leq |S| + |T| - 1$$

is valid for the polytope MK if and only if $T \cup \{i\}$ is a cover with respect to l for all $i \in S$. We call this inequality an *extended cover inequality.* We have been implementing a heuristic procedure for finding extended cover inequalities, which we outline below.

Extended Cover Heuristic
Choose two knapsacks $k, l \in M$, $k \neq l$ and solve the following linear program:

$$\begin{array}{ll} \min & \sum_{i \in N} (1 - x'_{ik} - x'_{il}) s_i \\ \text{s.t.} & \sum_{i \in N} f_i s_i > F_k, \\ & 0 \leq s_i \leq 1 \text{ for all } i \in N. \end{array}$$

Set $S := \{i \in N \mid s_i > 0\}$.
Reduce S to a minimal cover.
Let $s_{\min}$ be an item in S with minimum weight.
Solve the following linear program:

$$\begin{array}{ll} \min & \sum_{i \in N \setminus S}(1 - x'_{il})t_i \\ \text{s.t.} & \sum_{i \in N \setminus S} f_i t_i > F_l - f_{s_{\min}}, \\ & 0 \leq t_i \leq 1 \text{ for all } i \in N \setminus S. \end{array}$$

Set $T := \{i \in N \setminus S \mid t_i > 0\}$.
Reduce T so that $T \cup \{s_{\min}\}$ is a minimal cover.
Lift the extended cover inequality corresponding to S, T, k, and l.

Sequential lifting and complementing. Finally, let us give some remarks on the sequential lifting and complementing of inequalities. The concept of lifting is due to Padberg [14] and consists of the following idea. Given a polytope $P \subseteq [0,1]^N$ with $0/1$ vertices, let N denote the index set of variables. Let i be an element of N and suppose $a^T x \leq \alpha$ is a facet-defining inequality for $P \cap \{x \in \mathbb{R}^N \mid x_i = 0\}$. Set

$$\gamma_i := \max\{a^T x \mid x \in P, x_i = 1\}.$$

Then the inequality $(a + (\alpha - \gamma_i)e_i)^T x \leq \alpha$ defines a facet for the polytope P. In the particular cases of minimal cover inequalities and $(1,d)$-configuration inequalities, the coefficient γ_i can be computed in polynomial time. This result is due to Zemel [18] and is obtained by dualizing the maximization problem above, slightly modifying it, and solving this modified program by applying dynamic programming techniques. Zemel shows that the running time of the dynamic program is bounded by $O(\alpha|N|)$, which is polynomial in the cases of the minimal cover and $(1,d)$-configuration inequalities.

We have been implementing this idea and applied it to the classes of minimal cover and $(1,d)$-configuration inequalities. For the joint inequalities we cannot calculate the exact lifting coefficients efficiently. In the case of extended cover inequalities, we simply determine an upper bound, u_i say, on the value γ_i and determine the (approximate) lifting coefficient by setting $\overline{a}_i := \alpha - u_i$. This approach guarantees the validity of the generated inequality. To determine u_i we solve the linear relaxation of the maximization problem above and define u_i as the objective function value rounded down. In order to lift multiple cover inequalities we proceed in a slightly different way. Roughly speaking, the lifting of multiple cover inequalities is heuristically performed by lifting a cover inequality. Suppose $S \subseteq N$ is a multiple cover with respect to $M' \subseteq M$. We associate with the multiple cover inequality corresponding to S and M' a cover inequality $\sum_{i \in S} y_i \leq |S| - 1$ which is valid for the single knapsack polytope $SK(N, f, F(M'))$. The latter inequality is now lifted by using Zemel's procedure. Let $b^T y \leq |S| - 1$ denote this lifted inequality and define $\overline{a}_{ik} := b_i$ for all $k \in M'$ and $i \in N$. Then it is easy to see that the inequality $\overline{a}^T x \leq |S| - 1$ is a lifted multiple cover inequality corresponding to S and M' which is valid for the multiple knapsack polytope MK.

To the best of our knowledge the idea of complementing inequalities (lifting the complemented variables) was first suggested in [16]. Let $i \in N$ and $a^T x \leq \alpha$ be a facet-defining inequality for $P \cap \{x \in \mathbb{R}^N \mid x_i = 1\}$. Set

$$\gamma_i := \max\{a^T x \mid x \in P, x_i = 0\}.$$

Then the inequality $(a + (\gamma_i - \alpha)e_i)^T x \leq \gamma_i$ defines a facet for the polytope P. For complementing minimal cover and $(1,d)$-configuration inequalities, we again use Zemel's

procedure. But it is not clear that Zemel's worst-case analysis yields a polynomial time bound, since the (new) right-hand side γ_i (in the case of lifting this was α) may be exponential in the size of the input. However, in the cases of the minimal cover and $(1, d)$-configuration inequalities we could show that γ_i is bounded by $|N|$, thus yielding again a polynomial time algorithm (cf. [6]). This type of approach no longer works for complementing joint inequalities. Hence, we determine complementing coefficients by applying the same techniques as in the case of lifting.

Let us conclude this section with some remarks on the interplay among lifting, complementing, and separation routines. Suppose $x' \in \mathbb{R}^{N \times M}$ is a fractional solution that is obtained during the run of the cutting plane algorithm. In our implementation all variables that are close to zero or one will be fixed temporarily. More precisely, there is a threshold parameter, t say, and a variable x_{ik} is fixed if $x'_{ik} < t$ or $x'_{ik} > 1 - t$ holds. Initially, t is set to 0.05. This way, we reduce the size of the original problem instance drastically. Thereafter, the separation routines are called in order to find violated inequalities for the subpolytope $P := MK \cap \{x_{ik} = 0 \text{ if } x'_{ik} < t,\ x_{ik} = 1 \text{ if } x'_{ik} > 1 - t\}$. If the separation algorithms succeed in finding violated inequalities for P, the appropriate complementing and lifting procedures are applied. Table 2 shows the number of violated inequalities that are obtained by fixing variables to one (as described above), calling the separation routines, applying lifting, and complementing the corresponding inequalities. For more details on the problem instances we refer to §6.

TABLE 2
Separation in subproblems using the complementing procedure.

Problem	# viol. ineq.
cl2 0% red.	51
cl2 4% red.	138
dm1 36.8% red.	14
dm2 30 % red.	65
sa2	7

5. Further implementation details. At the end of the introduction we discussed the general outline of a cutting plane based algorithm. In this section we want to point out several aspects that were important in making our algorithm effective and applicable for the solution of practical problem instances.

First let us mention that we used the CPLEX Callable Library [4] for solving the linear programs, a very fast and robust LP solver written and supported by R.E. Bixby. The initial linear program in our algorithm is composed of the knapsack constraints, the SOS constraints, and the trivial inequalities $0 \leq x_{ik} \leq 1$ for all $i \in N$, $k \in M$. For the practical applications we have in mind all items must be assigned to the knapsacks, i.e., the SOS constraints must be satisfied with equality. Thus, instead of using the SOS constraints we add the corresponding equalities $\sum_{k \in M} x_{ik} = 1$ $(i \in N)$ to the initial linear program. From now on, we will call a solution for the multiple knapsack problem *feasible* only if all items are assigned to the knapsacks, i.e., if all SOS constraints are satisfied with equality.

One main aspect in the design of a cutting plane algorithm is to keep the actual linear program of moderate size. To this end, we eliminate inequalities from the LP that are not tight at the current linear programming solution x'. In each iteration we call all separation algorithms discussed in the previous section, but we control the

number of violated inequalities added to the LP by a parameter. If we find more violated inequalities than specified by this parameter, we take the most "violated" inequalities, i.e., the inequalities $a^T x \leq \alpha$ whose value $a^T x' - \alpha$ is smallest. We make sure that no redundant inequalities are added to the linear program.

An important issue in our algorithm is the fixing of variables by reduced costs. Let z' be the objective function value of the current LP solution, z^* be an upper bound on the value of the optimum solution, $x' = (x'_{ik})$ the current LP solution, and $d = (d_{ik})$ the corresponding reduced cost vector. Then each nonbasic variable x_{ik} with $x'_{ik} = 0$ and $z^* - z' \leq d_{ik}$ can be fixed to zero. Similarly, each nonbasic variable x_{ik} with $x'_{ik} = 1$ and $z^* - z' \leq -d_{ik}$ can be fixed to one. Moreover, further variables can be fixed by logical implications; for example, if some variable x_{ik} is fixed to one by the reduced cost criterion, then all variables x_{il}, $l \in M \setminus \{k\}$ can be fixed to zero. Table 3 shows for some problem instances the number of variables (in percentages) that are fixed by this procedure.

TABLE 3
Variables fixed by reduced costs.

Problem	Variables fixed	
cl2 0% red.	2565	(23.54%)
cl2 4% red.	2135	(19.59%)
dm1 36.8% red.	89	(8.79%)
dm2 30 % red.	1256	(27.12%)
sa2	1072	(72.43%)

Clearly, we cannot guarantee that our cutting plane algorithm finds an integer optimum solution, since a complete description of the multiple knapsack polytope is not known and exact separation algorithms for the known classes of inequalities are not at hand. Because of this fact, we have also implemented several LP based heuristics to obtain good upper bounds on the value of the optimal solution. These heuristics are based on rounding the current linear programming solution x'.

In the first heuristic we proceed as follows. We set $F'_k := F_k$ for all $k \in M$ and determine an item i whose value $\max\{x'_{ik} \mid k \in M \text{ and } f_i \leq F'_k\}$ is maximum. Let i^* be a "maximum" item and k^* the corresponding knapsack. We assign i^* to knapsack k^* and update the value F'_{k^*} by setting $F'_{k^*} := F'_{k^*} - f_{i^*}$. We continue this way until a feasible solution is found (i.e., all items are assigned) or no further item can be assigned.

The second heuristic differs from the first one in that it randomly selects the item that is assigned next. More precisely, we determine a random sequence of the items and, according to this sequence, we compute $\max\{x'_{ik} \mid k \in M \text{ and } f_i \leq F'_k\}$. If no more feasible knapsack for item i exists we stop. Otherwise, we assign item i to a knapsack k^* where the maximum value is attained, update F'_{k^*} accordingly, and continue.

In the third heuristic, we interpret each vector $(x'_{ik})_{k \in M}$, for $i \in N$, as a probability distribution, i.e., we toss a dial that assigns item i to knapsack k with probability x'_{ik}. The sequence according to which the items are chosen is again randomly determined. This procedure is performed several times depending on a parameter that is a multiple of the number of fractional variables.

It turns out that none of these heuristics is superior to the others. In most examples, the first two procedures are more successful at the very beginning, whereas the third one performs better later.

Moreover, we have implemented an improvement heuristic that is based on the ideas of [8]. The procedure applied to our problem is described in the following.

Improvement Heuristic
Initialize z' with z, where $z \in \{0,1\}^{N \times M}$ is a given feasible solution.
For $p := 1$ to *number-of-passes*
  Initialize z^p with z'.
  Set $F'_k := F_k - \sum_{i \in N} f_i z^p_{ik}$ for all $k \in M$.
  All items are supposed to be "unlocked."
  While $\{i \in N \mid i \text{ unlocked}, z^p_{ik} = 0, f_i \leq F'_k \text{ for some } k \in M\} \neq \emptyset$
    Choose an unlocked item i^* and a knapsack k^* and $z^p_{i^*k^*} = 0$
    and $c_{i^*k^*} = \min\{c_{ik} \mid i \text{ is unlocked}, z^p_{ik} = 0, \text{ and } f_i \leq F'_{k^*}\}$.
    Move item i^* to knapsack k^*.
    Update z^p and F'.
    Lock item i^* (in order to avoid cycling).
  Let z^* be the best solution found during this pass.
  If $c^T z^* < c^T z'$, set $z' := z^*$.
  Else return z' and STOP.
Return z'.

It turns out that these three primal heuristics together with the improvement heuristic perform quite well and even find the optimal solution in many cases (see the next section).

Finally, let us note that we have embedded our cutting plane algorithms in an enumeration scheme. Here, if we find no further violated inequality and the current linear programming solution is not an integer, we select a variable that is closest to 0.5 and create two subproblems, one where the selected variable is set to zero and one where this variable is set to one. The resulting branching tree is worked out in a depth first search manner.

6. Computational results. In this section we report on computational experiences with our cutting plane based algorithm. We have tested the algorithm on multiple knapsack problem instances arising in the design of main frame computers, in the layout of electronic circuits, and in the production of sugar cane alcohol.

Before presenting the results in detail, let us make some general comments on our experiences with the problem instances. As described in §4, we have separated cover, (1,d)-configuration, multiple cover, and extended cover inequalities. The number of cover and (1,d)-configuration inequalities we found are approximately the same for all examples. For joint inequalities, however, the situation is different. Almost all violated joint inequalities are multiple covers. In fact, extended covers were found only for one example. We do not know whether the extended cover inequalities are indeed rare in these particular problem instances or whether the performance of our separation heuristic is too poor. In the following tables we will simply distinguish between individual and joint inequalities.

Let us first focus on the class of problems that arise in the design of main frame computers. Table 4 summarizes the data. (Instances coming from this application are abbreviated by dm in the tables.) Columns 2 and 3 give the number of items and knapsacks, respectively, whereas in columns 4 and 5 the total weight of the items and the total sum of the capacities are shown.

Table 5 presents the solutions obtained by our algorithm. Neither individual nor joint inequalities were necessary to obtain the objective function value of an optimal

TABLE 4
Description of examples dm.

Problem	$\|N\|$	$\|M\|$	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
dm1	257	4	83827	132704
dm2	772	6	284608	423972

TABLE 5
Computational results for examples dm.

Problem	Opt. sol.	Ind. ineq.	Joint ineq.	CPU time
dm1	236250	0	0	1.62
dm2	81120	0	0	5.92

solution, which was already found by the primal heuristics in the first iteration. The CPU times are in seconds and were obtained on a Sun Sparc 4/50.

The fact that both problem instances are trivial is not surprising, since the total sum of the knapsack capacities is much bigger than the sum of the weights of the items. The reason for this is that after assigning the items to the devices the nets must be connected by wires, which requires a certain amount of space. The real amount of space that is necessary for connecting the wires is not available in advance. Thus, the numbers in column 5 of Table 4 are only a rough estimate. A usual procedure in practice is to start with some initial capacities of the devices and try to find a solution that assigns the modules to the devices and connects the nets by wires. If this succeeds, the capacities of the devices are reduced, the whole problem is solved again, and it is continued in this way until no further area reduction is possible. In fact, one of the main goals in the design of main frame computers is to reduce the available amount of space as far as possible. So, from a practical point of view a very interesting question is how far the capacities of the devices can be reduced. We followed this question and iteratively reduced the total amount of the knapsack capacities.

Table 6 summarizes our results. A reduction of 36.85% in example dm1 and a reduction of 33% in example dm2 leads to infeasibility, since the total available knapsack capacity is less than the sum of the weights of the items. Column 2 shows the amount of reduction and columns 3 to 6 present, respectively, the value of the optimal solution, the number of individual and joint inequalities found by our separation algorithms, and the total CPU time (min:sec). The percentual separation time is between 20% and 50% for these examples. All problem instances in Table 6 are solved to optimality without branching.

TABLE 6
Solutions of the reduced instances of dm *examples.*

Problem	Red.	Opt. sol.	Ind. ineq.	Joint ineq.	CPU
dm1	36.75%	236250	18	15	8.78
dm1	36.8%	236250	18	15	8.87
dm2	27%	81134	9	16	40.75
dm2	28%	81176	42	41	5:22.25
dm2	29%	81204	25	29	1:08.30
dm2	30%	81302	33	15	36:41.92

Two problems could not be solved to optimality by our algorithm without applying branching. Table 7 shows the results for these two examples. Column 6 shows the number of branch and cut nodes used to solve the problems.

TABLE 7
Difficult dm *instances.*

Problem	Red.	Opt. sol.	Ind. ineq.	Joint ineq.	BC	CPU
dm2	31%	81486	2547	3308	105	485:22.03
dm2	32%	81736	44	62	5	56:35.20

Let us now turn to the examples arising in the layout of electronic circuits. Table 8 summarizes the data. (Here, the instances are abbreviated by the symbol cl in the tables.) The ratio between the total weight of the items and the total available knapsack capacity is much closer to one than it is for the case of the instances in Table 4. One might expect that these instances are more difficult than the examples dm with 0% area reduction.

TABLE 8
Description of the examples cl.

Problem	$\|N\|$	$\|M\|$	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
cl1	2292	16	9522	10000
cl2	681	16	2571	2704
cl3	2669	16	6762	7104
cl4	1021	16	4031	4240
cl5	68	16	260	288
cl6	6112	16	25392	26672

However, the results are very similar. The first lower bound already provides the value of the optimal solution in almost all examples. A possible explanation for this fact is that in these examples the weights of the items are similar and there are many items with small weights and identical objective function values for all knapsacks. In all cases for which the value of the first LP is already equal to the value of the optimal solution our primal heuristics find an optimal solution in the first iteration of the cutting plane algorithm. The only nontrivial example is cl2, where individual and joint inequalities were indeed necessary to find the optimal solution. In Table 9 we show the value of an optimal solution, the number of individual and joint inequalities found by our separation algorithms, and the total CPU time (min:sec). Here, about 50% of the CPU time was spent in the separation routines.

TABLE 9
Solutions of examples cl.

Problem	Opt. solution	Ind. ineq.	Joint ineq.	CPU time
cl1	2292	0	0	3:58.53
cl2	939.99	143	46	11:40.97
cl3	2669	0	0	3:26.42
cl4	1021	0	0	28.43
cl5	472	0	0	1.73
cl6	6112	0	0	25:28.62

Again, as in the design of main frame computers, an interesting problem is to determine the minimum area for which the problems still have a feasible solution. We created some problem instances by reducing the total amount of the capacities for the knapsacks until the total weight of the items exceeds the total available knapsack capacities. Even here, we solve all reduced examples, except the reduced instances of cl2 in the first iteration, which is a very astonishing fact. The reduced instances of cl2 are much more difficult. In all but one example the first lower bound does not give the optimal objective function value. To solve these problems to optimality not only individual inequalities but also joint inequalities were necessary. Table 10 presents the results. (Note that a reduction of 5% leads to infeasibility, since in this case the total sum of the knapsack capacities is less than the total weight of the items.)

TABLE 10
Solutions of the reduced instances of cl2.

Problem	Red.	Opt. sol.	Ind. ineq.	Joint ineq.	CPU time
cl2	1%	946.99	164	94	11:18.48
cl2	2%	946.99	88	44	8:26.70
cl2	3%	960.99	163	119	14:33.45
cl2	4%	967.99	204	157	19:12.63

Our third set of instances comes from the application in sugar cane alcohol production. In Table 11 we show a description of the instances (they are abbreviated by sa in the tables). Columns 2 and 3 give the number of plantations and factories, respectively, whereas in columns 4 and 5 the total weight of the items (plantation production) and the total sum of the capacities of the factories are shown.

TABLE 11
Description of examples sa.

Problem	$\|N\|$	$\|M\|$	$\sum_{i\in N} f_i$	$\sum_{k\in M} F_k$
sa1	450	4	97714	106346
sa2	370	4	82914	97224

In contrast to the other two applications, different knapsacks have different capacities. Moreover, the weights of the items (the production of the plantations) cover a wide range of numbers. For our code these instances appear more difficult than the previous ones and we cannot solve these problems without branching. Our results are summarized in Table 12. About 20% of the CPU time (measured on a Sun Sparc 4/50 in minutes) was spent in the separation routines.

TABLE 12
Computational results for examples sa.

Problem	Opt. sol.	Ind. ineq.	Joint ineq.	BC	CPU time
sa1	14592921	110	14	121	57:28.68
sa2	11733362	15	9	48	19:33.73

The number of cutting planes compared to the number of branch and cut nodes is quite small. So, the following question may arise: why not use just branch and bound? However, by adding no cutting planes we were unable to prove optimality within several hundreds of branch and bound nodes. Thus, the individual and joint

inequalities (though quite few) are important for solving these instances in reasonable time. Certainly the numbers also indicate that there is still a lack of good inequalities for these instances.

An interesting peculiarity of most practical problem instances is that the first lower bound is quite close to the value of the optimal solution. The reason for this is that, for many items $i \in N$, the objective function coefficients c_{ik}, $k \in M$ are similar. However, the gap between the lower bound and the upper bound after the first iteration is larger by far. Only when individual and especially joint inequalities are added does the linear programming solution provide structural information such that the primal heuristics find good upper bounds or even an optimal solution. This fact can also be observed by running random examples.

TABLE 13
Random examples.

$\|N\|$	$\|M\|$	Average gap after first LP	Average gap after ind. ineq.		Average gap after joint ineq.	
50	4	561.0	231.4	(41.2%)	36.6	(6.5%)
100	4	265.4	164.0	(61.8%)	78.6	(29.6%)
150	4	355.2	149.2	(42.0%)	114.2	(32.3%)
200	4	335.4	117.0	(34.9%)	75.2	(22.4%)
300	4	332.0	112.4	(33.9%)	100.6	(30.3%)
400	4	210.5	154.8	(73.5%)	90.0	(42.8%)
500	4	171.0	41.5	(24.3%)	38.2	(22.4%)

Table 13 provides typical results for random problems. The first two columns give the number of items and the number of knapsacks. The weights of the items are randomly chosen from the interval $[5, 300]$, the objective function coefficients are random numbers in the interval $[1, 1000]$, and the knapsack capacities are randomly computed such that $\sum_{k \in M} F_k \leq \alpha \sum_{i \in N} f_i$, where α is a random number chosen from $[1.05, 1.3]$. We have created four different problems with the same number of knapsacks and items. The numbers in columns 3 to 5 show the average over the absolute values of the gaps between the upper bounds and the lower bounds after the first iteration, after no further violated individual inequalities were found, and after no further violated joint inequalities were found. The numbers in parentheses correspond to the ratio of the gap (columns 4 and 5) and the gap of the first LP (column 3). Although we cannot solve most of these random examples to optimality without branching, the results confirm that the gap between the lower and upper bounds is substantially decreased by using individual and joint inequalities.

REFERENCES

[1] E. BALAS, *Facets of the knapsack polytope*, Math. Programming, 8 (1975), pp. 146–164.
[2] E. BALAS AND E. ZEMEL, *Facets of the knapsack polytope from minimal covers*, SIAM J. Appl. Math., 34 (1978), pp. 119–148.
[3] E. A. BOYD, *A pseudo-polynomial network flow formulation for exact knapsack separation*, Networks, 22 (1992), pp. 503–514.
[4] *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, CPLEX Optimization, 1992.
[5] H. CROWDER, E. L. JOHNSON, AND M. W. PADBERG, *Solving large-scale zero-one linear programming problems*, Oper. Res., 31 (1983), pp. 803–844.
[6] C. E. FERREIRA, *On Combinatorial Optimization Problems Arising in Computer Systems Design*, Ph.D. thesis, Technische Universität Berlin, 1994.

[7] C. E. FERREIRA, M. GRÖTSCHEL, S. KIEFL, C. KRISPENZ, A. MARTIN, AND R. WEISMANTEL, *Some integer programs arising in the design of main frame computers*, ZOR, 38 (1993), pp. 77–100.

[8] C. M. FIDUCCIA AND R. M. MATTHEYSES, *A linear-time heuristic for improving network partitions*, in Proc. 19th DAC, ACM/IEEE, 1982, pp. 175–181.

[9] E. S. GOTTLIEB AND M. R. RAO, *The generalized assignment problem: Valid inequalities and facets*, Math. Programming, 46 (1990), pp. 31–52.

[10] ———, *$(1,k)$-configuration facets for the generalized assignment problem*, Math. Programming, 46 (1990), pp. 53–60.

[11] P. L. HAMMER, E. L. JOHNSON, AND U. N. PELED, *Facets of regular 0-1 polytopes*, Math. Programming, 8 (1975), pp. 179–206.

[12] R. M. KARP, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.

[13] S. MARTELLO AND P. TOTH, *Knapsack Problems—Algorithms and Computer Implementations*, John Wiley, New York, 1990.

[14] M. W. PADBERG, *A note on 0-1 programming*, Oper. Res., 23 (1975), pp. 833–837.

[15] ———, *$(1,k)$-configurations and facets for packing problems*, Math. Programming 18 (1980), pp. 94–99.

[16] L. A. WOLSEY, *Faces of linear inequalities in 0-1 variables*, Math. Programming, 8 (1975), pp. 165–178.

[17] ———, *Valid inequalities for 0-1 knapsacks and MIPS with generalised upper bound constraints*, Discrete Appl. Math., 29 (1990), pp. 251–261.

[18] E. ZEMEL, *Easily computable facets of the knapsack polytope*, Math. Oper. Res., 14 (1989), pp. 760–764.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 879–912, November 1996
© 1996 Society for Industrial and Applied Mathematics

CONDITION NUMBERS, THE BARRIER METHOD, AND THE CONJUGATE-GRADIENT METHOD*

JAMES RENEGAR†

Abstract. We show that the condition numbers of the systems of linear equations arising during application of an interior-point method can be nicely bounded in terms of a quantity which is naturally interpreted as the condition number of the system of inequalities to which the method is being applied. We rely on the bound to analyze the efficiency of the conjugate-gradient method in approximately solving the linear equations.

Key words. condition number, interior-point method, barrier method, conjugate-gradient method, linear programming, complexity theory

AMS subject classifications. 15A12, 65F35, 68Q25, 90C05, 90C25

1. Introduction.

1.1. We consider the (theoretical) efficiency of the conjugate-gradient method in approximately solving the linear equations which arise during application of a particular interior-point method. The interior-point method (approximately) solves systems of constraints of the form

$$\begin{aligned} Ax &= b, \\ x &\geq 0, \end{aligned} \tag{1}$$

where $x \geq 0$ is interpreted quite generally. The interior-point method we consider is simply the barrier method applied to a carefully chosen reformulation of the constraints (1).

Actually, our main results are nice relations between the condition numbers of the systems of linear equations which arise during application of the barrier method and a particular quantity which is naturally interpreted as the condition number of the constraints (1). These relations have ramifications well beyond our analysis of the conjugate-gradient method. Readers interested in condition numbers, but not interested in our conjugate-gradient method analysis, should read §§1.4–1.8, 1.10, and 1.12 (through Corollary 1.7), ignoring sentences which refer to the conjugate-gradient method.

This paper continues in the conceptual vein developed in the introductory section of [6]. To avoid devoting numerous journal pages to redundant exposition, we assume the reader to have browsed that introductory section (but not the other sections of [6]); that introductory section provides crucial motivation as well as development of the abstract Hilbert-space setting (i.e., coordinate-free setting) underlying our exposition.

We rely extensively on order notation[1] throughout this paper.

* Received by the editors October 7, 1994; accepted for publication (in revised form) August 5, 1995.

† College of Engineering, School of Operations Research, Cornell University, E&TC Building, Ithaca, NY 14853-3801 (renegar@orie.cornell.edu).

[1] We rely extensively on order notation (i.e., O, Ω, and Θ) throughout this work, supressing constants partly in hope of achieving more transparent proofs. Thus, for example, a statement of the form

"If $\alpha = O(\beta)$, then $\gamma = \Theta(\delta)$"

means that there exist universal positive constants κ_1, κ_2, and κ_3 such that

if α and β satisfy $\alpha \leq \kappa_1 \beta$, then γ and δ satisfy $\kappa_2 \delta \leq \gamma \leq \kappa_3 \delta$.

We begin by recalling a few important, widely known, theoretical features of the conjugate-gradient method (cgm); some of our results will be seen as analogues to these.

1.2. Assume that X is a Hilbert space, $A : X \to X$ is a strictly positive-definite[2] linear operator, and $b \in X$. Define the inner product $\langle\ ,\ \rangle_A$ and norm $\|\ \|_A$ by

$$\langle u, v\rangle_A := \langle u, Av\rangle, \quad \|u\|_A := \langle u, u\rangle_A^{1/2},$$

where $\langle\ ,\ \rangle$ is the (original) inner product on X.

Let x^* denote the solution of the system $Ax = b$. Provided with an initial iterate x_0, the cgm can be applied to solve the system; the cgm produces the sequence of iterates $x_1, x_2 \ldots$, where x_i is the optimal solution for the following problem:

$$\begin{array}{ll} \min & \|x^* - x\|_A \\ \text{s.t.} & x - x_0 \in \text{Span}\{Ae_0, A^2e_0, \ldots, A^ie_0\}; \end{array} \tag{2}$$

here, $e_0 := x^* - x_0$. In the literature, one often finds this optimality property possessed by the cgm expressed somewhat differently, as we now recall.

Clearly, the vectors in $\text{Span}\{Ae_0, A^2e_0, \ldots, A^ie_0\}$ are precisely the vectors of the form $p(A)e_0$, where p ranges over polynomials of degree at most i which satisfy $p(0) = 0$. Since

$$x^* - x = e_0 - (x - x_0),$$

one thus sees that the optimal value of (2) (i.e., $\|x^* - x_i\|_A$) equals the optimal value of the following problem:

$$\begin{array}{ll} \min & \|p(A)e_0\|_A \\ \text{s.t.} & p \in \mathcal{P}_i, \end{array} \tag{3}$$

where $\mathcal{P}_i$ is the set consisting of those polynomials p of degree at most i which satisfy $p(0) = 1$. (It follows, for example, that if A has at most k distinct values in its spectrum, $\text{Spec}(A)$, then $x_i = x^*$ whenever $i \geq k$.)

As is well known, the fact that $\|x^* - x_i\|_A$ equals the optimal value of (3) allows one to deduce, via Chebyshev polynomials, a bound on $\|x^* - x_i\|_A$ in terms of the relative condition number of $\sqrt{A}$:

$$\text{relcond}(\sqrt{A}) := \|\sqrt{A}\|\,\|\sqrt{A^{-1}}\| \ (= \sqrt{\text{relcond}(A)}).$$

Precisely, one can prove that

$$\|x^* - x_i\|_A \leq 2\left(\frac{\text{relcond}(\sqrt{A}) - 1}{\text{relcond}(\sqrt{A}) + 1}\right)^i \|x^* - x_0\|_A; \tag{4}$$

cf. [1, Appendix B and (5.30)]. From this it is easily deduced that if $0 < \epsilon \leq 1$ then the cgm produces x_i satisfying

$$\|x^* - x_i\|_A \leq \epsilon\|x^* - x_0\|_A$$

Similarly,

"If $\alpha = \Theta(\beta)$, then $\gamma = \Omega(\delta)$"

means that there exist universal positive constants κ_1, κ_2, and κ_3, where $\kappa_1 < \kappa_2$, such that

if α and β satisfy $\kappa_1\beta \leq \alpha \leq \kappa_2\beta$, then γ and δ satisfy $\kappa_3\delta \leq \gamma$.

And so on.

[2] A is selfadjoint and $\langle x, Ax\rangle > 0$ for all nonzero $x \in X$.

within

$$O\left(\text{relcond}(\sqrt{A})\log_2 \tfrac{2}{\epsilon}\right)$$

iterations, a fact we express in shorthand as

$$i = \Omega\left(\text{relcond}(\sqrt{A})\log_2 \tfrac{2}{\epsilon}\right) \Rightarrow \|x^* - x_i\|_A \leq \epsilon\|x^* - x_0\|_A. \tag{5}$$

Although (4) does not in general imply

$$\|x^* - x_i\|_A \leq \|x^* - x_0\|_A \text{ for all } i, \tag{6}$$

the relation (6) is indeed valid (virtually by definition of the cgm).

1.3. If one wishes to solve a least-squares problem

$$\min \|Ax - b\|,$$

where $A : X \to Y$ and X, Y are Hilbert spaces, one can apply the cgm to solve the system

$$A^T Ax = A^T b$$

if $0 \notin \text{Spec}(A^T A)$. Letting $x_1, x_2, \ldots$ denote the resulting sequence of iterates, as an immediate corollary to (5) one obtains

$$i = \Omega\left(\text{relcond}(\sqrt{A^T A})\log_2 \tfrac{2}{\epsilon}\right) \Rightarrow \|x^* - x_i\|_{A^T A} \leq \epsilon\|x^* - x_0\|_{A^T A},$$

where x^* is the least-squares solution. To understand the relevance of the norm $\| \ \|_{A^T A}$ in this context, note that

$$\|x^* - x_i\|^2_{A^T A} = \|Ax_i - b\|^2 - \min\{\|Ax - b\|^2;\ x \in X\}.$$

In a similar vein, if $0 \notin \text{Spec}(AA^T)$ then one can use the cgm to solve $Ax = b$ where $A : X \to Y$. Given initial $y_0 \in Y$, one applies the cgm to the system

$$AA^T y = b$$

to obtain a sequence $y_1, y_2, \ldots$. As an immediate corollary to (5) one obtains

$$i = \Omega\left(\text{relcond}(\sqrt{AA^T})\log_2 \tfrac{2}{\epsilon}\right) \Rightarrow \|x^* - A^T y_i\| \leq \epsilon\|x^* - A^T y_0\|, \tag{7}$$

where x^* is the minimal norm solution for the system $Ax = b$. This is the application of the cgm that we will refer to later.

For future reference we also record the inequality analogous to (6); namely,

$$\|x^* - A^T y_i\| \leq \|x^* - A^T y_0\| \text{ for all } i. \tag{8}$$

1.4. We now begin considering (very general) systems of constrained linear equations to be solved via the barrier method.

Consider a system of constraints

$$\begin{aligned} Ax &= b, \\ x &\geq 0, \end{aligned} \tag{9}$$

where

(i) $A : X \to Y$ is a continuous linear operator,
(ii) X and Y are real Hilbert spaces,
(iii) $x \geq 0$ means $x \in C$ where $C \subseteq X$ is a fixed, closed, convex cone with vertex at the origin.

Such very general definitions of constraints date at least to Duffin [2].

By specializing the cone C, one obtains customary forms of constraints appearing in the linear programming literature. For example, if one lets $X = \mathbb{R}^n$ and wishes to impose nonnegativity constraints on the first J variables, then one lets

$$C := \{x \in \mathbb{R}^n;\ x_j \geq 0 \text{ for all } j = 1, \ldots, J\}. \tag{10}$$

We consider perturbations in A and b but we always consider the cone C as fixed (once it has been chosen); we will impose only minor restrictions on the choice of C.

1.5. Much of our recent research (cf. [6]) examines the efficiency of algorithms for solving general constraints like (9), where the efficiency is measured in terms of a parameter which is a type of condition number; the parameter is the reciprocal of the minimal relative amount by which the system $Ax = b$, $x \geq 0$, can be perturbed to obtain a system which is inconsistent. Formally, letting (A, b) denote the system $Ax = b$, $x \geq 0$, and defining

$$\|(A, b)\| := \max\{\|A\|, \|b\|\},$$

(where $\|A\|$ is the operator norm induced by the norms on X and Y), we define

$$\text{relcond}(A, b) := \frac{\|(A, b)\|}{\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\}},$$

where "$(A + \Delta A, b + \Delta b)$ is inconsistent" means that the system

$$\begin{gathered}(A + \Delta A)x = b + \Delta b, \\ x \geq 0\end{gathered}$$

is inconsistent. As is discussed at great length in [6, §1], $\text{relcond}(A, b)$ is a naturally defined condition number for systems of constraints.

It might be useful for the reader to think of the constraints as being scaled so that $\|A\| \approx \|b\|$.

It is not difficult to verify that in the case of general systems of linear equations $Ax = b$ (i.e., in the case that $C = X$) where $\|b\| \leq \|A\|$, one has

$$\text{relcond}(A, b) = \text{relcond}(\sqrt{AA^T}), \tag{11}$$

precisely the principal parameter in (7); moreover, in the case of "square" systems of linear equations (i.e., when, in addition, $X = Y$), the values in (11) equal $\|A\|\,\|A^{-1}\|$, the traditional condition number.

1.6. We make minor assumptions regarding the cone C so that now-standard barrier method theory is readily applicable.

We assume that the interior C° of the cone C is dense in C. Moreover, in the terminology of [5] and [6, §1], we assume C° to be the domain of a functional $\phi \in \mathcal{F}'(K)$, where $K \geq 0$; in the terminology of Nesterov and Nemirovskii [3], "$\phi \in \mathcal{F}'(K)$" essentially means that ϕ is a self-concordant barrier functional whose barrier parameter is of magnitude $O(K^2)$.

For example, if $X := \mathbb{R}^n$ and C is the cone (10) then one may let

$$\phi_x := -12 \sum_{j=1}^{J} \ln x_j \quad \text{and} \quad K = \sqrt{12\,J}. \tag{12}$$

The factor 12 is not crucial.[3] What is important is the order of K, i.e., $K = O(\sqrt{J})$.

We write $x > 0$ to mean $x \in C^\circ$.

Let $\breve{x}$ denote the point which minimizes the functional

$$x \mapsto \phi_x - 12 \ln(1 - \|x\|^2),$$

a functional whose domain is

$$\{x;\ x > 0 \ \text{ and } \ \|x\| < 1\}, \tag{13}$$

i.e., $\breve{x}$ is the analytic center of the region (13). For simplicity, we assume that $\breve{x}$ is known; this assumption can be relaxed at the price of increased technicalities.[4]

For example, if $X := \mathbb{R}^n$ is equipped with the Euclidean inner product, C is the cone (10), and ϕ is defined by (12), then $\breve{x}$ is the vector with coordinates

$$\breve{x}_j := \begin{cases} \sqrt{J/(J^2+2)} & \text{if } j \le J, \\ 0 & \text{if } j > J. \end{cases}$$

1.7. We now reformulate the constraints $Ax = b$, $x \ge 0$ so that they might be approximately solved by the barrier method. Our reformulation depends on two additional variables, s and t. Roughly, s is a homogenization variable used to transform the (relative interior of the) possibly unbounded feasible region for the constraints into a bounded set, and t is a homotopy parameter. Thus, the barrier method iterates will lie in the vector space

$$\mathcal{H} := X \times \mathbb{R} \times \mathbb{R}$$

which is a real Hilbert space (since X is a real Hilbert space) if we define an inner product by

$$\langle (x_1, s_1, t_1), (x_2, s_2, t_2) \rangle := \langle x_1, x_2 \rangle + s_1 s_2 + t_1 t_2.$$

[3] The factor 12 is simply chosen so that the theory described in [5] and [6] is immediately applicable and hence we can avoid proving standard details concerning the barrier method; all of the details we need are undoubtedly contained somewhere in the fundamental work of Nesterov and Nemirovskii [3] and elsewhere. The theory in [5] and [6] involves a natural normalization constant (namely, 1) which leads to the rather unnatural factor 12 above. Instead, one can incorporate the unnatural constant 12 into the theory, thus eliminating the need for the factor 12 above; but since the theory applies beyond the natural log function, it seems preferable to have the natural constant 1 in the theory. However, all of this discussion is moot, for in the present paper we hide all constants in big-O's; consequently, the normalization is actually irrelevant—all of our results are in fact true if the above factor 12 is replaced by 1 (likewise with *all* of the 12's one encounters in this paper). Not wanting to go through the labor of making this claim rigorous, we leave the factor 12 so that the theory described in [5] and [6] is indeed immediately applicable.

[4] For example, if one wishes to assume that *some* point

$$\breve{x} \in \{x \in C^\circ;\ \|x\| < 1\}$$

is known, then, in the terminology of [6, §1], one can develop our results in terms of the quantity $\mathrm{sym}(\breve{x}, C(1))$.

We apply the barrier method to approximately solve the following optimization problem:

$$\begin{array}{ll} \inf & t \\ \text{s.t.} & \bar{A}(x,s,t) = 0, \\ & x > 0, \\ & \|x\| < 1, \\ & 0 < s < 1, \\ & 0 < t < 2, \end{array} \tag{14}$$

where $\bar{A} : \mathcal{H} \to Y$ is the continuous linear operator defined by

$$\bar{A}(x,s,t) := Ax - sb - t(A\check{x} - \tfrac{1}{2}b). \tag{15}$$

To understand the relevance of the optimization problem (14), observe that if $(x,s,0)$ is the limit of a sequence of points which are feasible for the problem and if $s > 0$, then $\frac{1}{s}x$ it satisfies the original constraints to be solved, i.e., it satisfies $Ax = b$, $x \geq 0$.

Let val_{inf} denote the optimal value of (14) and let val_{sup} denote the optimal value of the problem obtained by replacing "inf" in the objective of (14) with "sup." Observe that the point

$$w_0 := (\check{x}, \tfrac{1}{2}, 1)$$

satisfies the constraints of (14); hence,

$$1 \leq \text{val}_{\text{sup}} \leq 2.$$

Moreover, observe that the feasible region for (14) is closed under scaling by positive constants less than unity (i.e., if (x,s,t) satisfies the constraints and $0 < r < 1$, then (rx, rs, rt) satisfies the constraints); consequently,

$$\text{val}_{\text{inf}} = 0.$$

In applying the barrier method to approximately solve (14), we rely on the functional

$$f_{x,s,t} := \phi_x - 12\left[\ln(1 - \|x\|^2) + \ln(s) + \ln(1-s) + \ln(t) + \ln(2-t)\right], \tag{16}$$

whose domain is

$$D_f := \{(x,s,t);\ x > 0,\ \ \|x\| < 1,\ \ 0 < s < 1 \text{ and } 0 < t < 2\}.$$

In the terminology of [5] and [6] we have that $f \in \mathcal{F}(\sqrt{\bar{K}^2 + 72})$; that is, $f \in \mathcal{F}(\bar{K})$, where

$$\bar{K} = K + O(1);$$

in the terminology of Nesterov and Nemirovskii [3], $f \in \mathcal{F}(\bar{K})$ essentially means that f is a nondegenerate self-concordant barrier functional whose barrier parameter is of magnitude $O(\bar{K}^2)$.

Defining $c := (\vec{0}, 0, 1) \in \mathcal{H}$, we find it convienent to rewrite the optimization problem (14) in the following general format:

$$\begin{array}{ll} \inf & \langle c, w \rangle \\ \text{s.t.} & \bar{A}w = 0, \\ & w \in D_f. \end{array} \tag{17}$$

1.8. Due to our assumptions concerning the point $\check{x}$, it is not difficult to show that $w_0 := (\check{x}, \frac{1}{2}, 1)$ minimizes f. Consequently, as arguments which are well known to the interior-point method (ipm) community show, if $\mu_1 = \mu$ is a constant satisfying

$$0 < \mu = O\left(\frac{1}{\mathrm{val}_{\mathrm{sup}} - \mathrm{val}_{\mathrm{inf}}}\right), \tag{18}$$

then w_0 is "close" to the unique optimal solution $z_{\mu_1} = z_\mu$ of the following linearly constrained optimization problem:

$$\begin{array}{ll} \min_w & \mu\langle c, w\rangle + f_w \\ \text{s.t.} & \bar{A}w = 0. \end{array} \tag{19}$$

Being close implies that one iteration of Newton's method, initiated at w_0, will yield a point $w_1 \in D_f$ which is closer than w_0 to z_{μ_1}. The barrier method computes w_1 and then replaces the constant μ_1 by a larger value μ_2. One iteration of Newton's method, initiated at w_1, then yields w_2. And so on, with an increasing sequence of constants μ_i. If all of the constants μ_i are chosen to satisfy

$$\frac{\mu_{i+1}}{\mu_i} - 1 = O\left(\frac{1}{\overline{K}}\right), \tag{20}$$

then the barrier method stays on track, i.e., w_i will be close to $z_{\mu_{i+1}}$ for all i. Moreover, if all of the constants μ_i are chosen to satisfy

$$\frac{\mu_{i+1}}{\mu_i} - 1 = \Theta\left(\frac{1}{\overline{K}}\right), \tag{21}$$

then for each ϵ satisfying $0 < \epsilon \leq 1$,

$$i = \Omega\left(\overline{K}\log_2(\overline{K} + 1 + \tfrac{1}{\epsilon})\right) \;\Rightarrow\; \frac{\langle c, w_i\rangle - \mathrm{val}_{\mathrm{inf}}}{\mathrm{val}_{\mathrm{sup}} - \mathrm{val}_{\mathrm{inf}}} < \epsilon. \tag{22}$$

Letting $w_i := (x_i, s_i, t_i)$ and recalling that in our application $\mathrm{val}_{\mathrm{sup}} \geq 1$ and $\mathrm{val}_{\mathrm{inf}} = 0$, (22) can be rewritten as

$$i = \Omega\left((K+1)\log_2(K + 1 + \tfrac{1}{\epsilon})\right) \;\Rightarrow\; t_i < \epsilon. \tag{23}$$

Close is defined in terms of a particular norm. Precisely, for each $w \in D_f$, one defines an inner product on $\mathcal{H}$ by

$$\langle u, v\rangle_w := \langle u, H_w v\rangle,$$

where H_w is the Hessian of f at w. In turn, this inner product gives rise to the norm

$$\|v\|_w := \langle v, v\rangle_w^{1/2}.$$

To say that w is close to z_μ is to say that $\|z_\mu - w\|_w$ is less than a particular universal positive constant, say, κ. The particular constant κ is not particularly important and, in fact, differs from paper to paper; that the constant is universal (i.e., problem instance independent) *is* important. (Moreover, it is also important that the theory allows one to replace the universal constant κ defining "close" with any smaller positive constant *if* one alters the other universal constants occurring in the theory

appropriately; e.g., if one decreases κ then one must also decrease the universal constants hidden in the big-O notation of (18), (20), and (21) and increase the universal constant hidden in (22).)

1.9. As is well known, to ensure that the barrier method stays on track, it is sufficient for the Newton steps to be accurately approximated rather than computed exactly. More precisely, letting n_{w_i} denote the exact *Newton step* at w_i for the linearly constrained optimization problem (19) when $\mu = \mu_{i+1}$, it is sufficient to compute an approximation $n_{w_i} + \Delta n_{w_i}$, where $\|\Delta n_{w_i}\|_{w_i}$ is small; one then lets the next iterate be

$$w_{i+1} := w_i + n_{w_i} + \Delta n_{w_i}.$$

If the iterate w_i is close to z_{μ_i} and if $\|\Delta n_{w_i}\|_{w_i}$ is indeed small, then the iterate w_{i+1} will be close to $z_{\mu_{i+1}}$ (assuming (20)).

To say that $\|\Delta n_{w_i}\|_{w_i}$ is small is to say that $\|\Delta n_{w_i}\|_{w_i}$ is less than a particular universal positive constant, say, γ. The determination of an appropriate universal constant γ is intertwined with the determination of an appropriate universal constant κ (defining "close") and the determination of the appropriate universal constant hidden in (20). Although somewhat involved, such analysis is now commonplace; cf. [3], [5].

Now, for general problems of the form (17), when one only accurately approximates the Newton steps in computing the sequence $\{w_i\}$, the implication (22) need not be valid; it can happen that w is close to z_μ and $\langle c, z_\mu \rangle \approx \text{val}_{\text{inf}}$ but $\langle c, w \rangle$ is nowhere near val_{inf} *if* $\bar{A}w \neq 0$. However, if the optimal value val_{inf} of a general problem of the form (17) satisfies

$$\text{val}_{\text{inf}} = \inf\{\langle c, w \rangle;\ w \in D_f\}, \tag{24}$$

then indeed the implication (22) is valid even when one only accurately approximates the Newton steps rather than computing them exactly; this is not difficult to prove, cf. [4].

Recalling that $\text{val}_{\text{inf}} = 0$ for the problem (14) in which we are interested and noting the constraint $t > 0$ in defining the domain D_f for that problem, we see that for us the identity (24) is indeed fulfilled; hence, (22) (equivalently, (23)) is valid even when the Newton steps are accurately approximated rather than computed exactly.

1.10. Since the theory only requires that the Newton steps be accurately approximated, it is natural to investigate the efficiency of the cgm in computing accurate approximations.

As is explained in §2, if $w \in D_f$ then the Newton step n_w for the linearly constrained optimization problem (19) is

$$n_w := -H_w^{-1}(\mu c + g_w + \bar{A}^T y),$$

where y solves

$$\bar{A}H_w^{-1}\bar{A}^T y = \bar{A}(w - H_w^{-1}(\mu c + g_w)); \tag{25}$$

here, g_w denotes the gradient of f at w and H_w denotes the Hessian. The operator H_w^{-1} is easily computed for the cones C of common interest, e.g., (10); in fact, H_w (hence H_w^{-1}) is typically a diagonal matrix. Thus, assuming H_w^{-1} to be readily available, the main work in computing the Newton step lies in solving the linear equations (25); it is to these equations that one applies the cgm.

Of course when one applies the cgm to the linear equations (25) one does not form the matrix product $\bar{A}H_w^{-1}\bar{A}^T$ but instead limits oneself to matrix-vector multiplications so as to keep the number of arithmetic operations low and so as to take full advantage of sparsity in $\bar{A}$ and H_w^{-1}; note that any sparsity in A and b is reflected in $\bar{A}$ (by choice of $\bar{A}$).

Our discussion of the cgm indicates that the efficiency of the cgm in approximately solving (25) is largely governed by the condition number

$$\text{relcond}\left(\sqrt{\bar{A}H_w^{-1}\bar{A}^T}\right). \tag{26}$$

In order to measure the efficiency of the cgm in terms that are relevant to the original constraints $Ax = b,\ x \geq 0$, it is thus natural to attempt to relate the condition number (26) to the condition number $\text{relcond}(A, b)$. The following theorem and proposition yield a corollary which provides such a relation.

Recall that the *central trajectory* of the optimization problem (14) is the path

$$\text{CentTraj} := \{z_\mu;\ \mu \geq 0\}.$$

For $w \in D_f$, define

$$\text{dist}(w, \text{CentTraj}) := \inf\{\|z - w\|_w;\ z \in \text{CentTraj}\}.$$

As with all of our results, the universal constants hidden in the order notation of the following theorem are independent of X, Y, C, A, and b, i.e, they are truly universal. The fact that the constants appearing in our results are indeed universal is perhaps the most important and surprising (to the author, at least) aspect of this work. With additional effort, and much longer and more obscure proofs, one could determine specific universal constants. Our research goal was to prove that such universal constants exist, i.e., provide only foundational theory.

THEOREM 1.1. *Assume* $\text{relcond}(A, b) < \infty$. *If* $w \in D_f$ *satisfies*

$$\text{dist}(w, \text{CentTraj}) = O(1),$$

then

$$\left\|\sqrt{(\bar{A}H_w^{-1}\bar{A}^T)^{-1}}\right\| \leq \frac{O(K^2+1)}{\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}}.$$

Proof. See §5. □

PROPOSITION 1.2. *If* $w \in D_f$, *then*

$$\left\|\sqrt{\bar{A}H_w^{-1}\bar{A}^T}\right\| = O(\|(A, b)\|).$$

Proof. See §3. □

COROLLARY 1.3. *Assume* $\text{relcond}(A, b) < \infty$. *If* $w \in D_f$ *satisfies*

$$\text{dist}(w, \text{CentTraj}) = O(1),$$

then

$$\text{relcond}\left(\sqrt{\bar{A}H_w^{-1}\bar{A}^T}\right) \leq O(K^2+1)\,\text{relcond}(A, b).$$

Proof. The proof follows immediately from both Theorem 1.1 and Proposition 1.2. □

In the author's opinion, Corollary 1.3 is one of the two most interesting results in this paper, both of which have ramifications beyond the analysis of the cgm. Our discussion of the cgm serves primarily as a showcase for the two results.

It is worth noting that Corollary 1.3 cannot be established by bounding the condition numbers of, say, $\sqrt{\bar{A}\bar{A}^T}$ and $\sqrt{H_w^{-1}}$ separately because

$$\lim_{\mu \to \infty} \text{relcond}\left(\sqrt{H_w^{-1}}\right) = \infty.$$

One of the main tools in proving Theorem 1.1 is, not surprisingly, the Hahn–Banach theorem.

1.11. It is not immediately evident how Corollary 1.3 provides an upper bound on the number of cgm iterations which are sufficient to compute an appropriately accurate approximation $\tilde{y}$ to the exact solution of the linear equations (25), that is, a vector $\tilde{y}$ such that if

$$\Delta n_w := -H_w^{-1}(\mu c + g_w + \bar{A}^T \tilde{y}) - n_w$$

then $\|\Delta n_w\|_w < \gamma$ (i.e., $\|\Delta n_w\|_w$ is small) and hence

$$-H_w^{-1}(\mu c + g_w + \bar{A}^T \tilde{y})$$

is an appropriately accurate approximation to the Newton step n_w. In §2 we derive an upper bound on the number of cgm iterations which are indeed sufficient, letting the initial cgm iterate be $y_0 := 0$. Assuming $w \in D_f$ satisfies $\|z_\mu - w\|_w = O(1)$ (i.e., w is close to z_μ), we use Corollary 1.3 to show that

$$O\left((K^2 + 1)\,\text{relcond}(A, b)\,\log_2(K + 2)\right) \tag{27}$$

cgm iterations suffice.

The factor $(K^2 + 1)\log_2(K + 2)$ appearing in the iteration bound (27) certainly limits the interest of that bound. In particular, if Y is finite-dimensional, i.e., $Y = \mathbb{R}^m$, then the bound is of no interest if $K^2 \geq m$ (or $\text{relcond}(A, b) \geq m$) because the linear equations (25) will involve only m variables and hence will be solved exactly by the cgm within m iterations. The bound is only of interest if K^2 is small relative to m. For example, the bound is of interest if $X := \mathbb{R}^n$ is equipped with the Euclidean inner product, C is the cone (10) where $J << m$, and ϕ is defined by (12); then $K^2 = 12\,J << m$.

1.12. Of course the efficient approximation of the Newton steps is only a means to an end, the goal being the computation of an accurate approximation to a solution of the constraints $Ax = b$, $x \geq 0$. One would like to know something about the overall amount of work which is sufficient to achieve the goal, where the overall amount of work is measured in terms of $\text{relcond}(A, b)$. Toward providing a bound on the overall amount of work, we describe an approach to computing an accurate approximation to a solution of the constraints $Ax = b$, $x \geq 0$ when given a barrier method iterate $(x, s, t) \in D_f$.

Let $f' : X \times \mathbb{R} \to \mathbb{R}$ denote the functional

$$f'_{x,s} := \phi_x - 12\left[\ln(1 - \|x\|^2) + \ln(s) + \ln(1 - s)\right],$$

with domain

$$D_{f'} := \{(x, s);\ x > 0,\ \ \|x\| < 1, \text{ and } 0 < s < 1\}.$$

Note that f' is identical to f except that in the definition of f' there is no reference to the homotopy parameter t. If $(x, s, t) \in D_f$, then $(x, s) \in D_{f'}$.

We let H'_w denote the Hessian of f' at $w = (x, s) \in D_{f'}$.

In the terminology of [5] and [6] we have that $f' \in \mathcal{F}(\sqrt{K^2 + 48})$, that is, $f' \in \mathcal{F}(K')$, where

$$K' = K + O(1);$$

in the terminology of Nesterov and Nemirovskii [3], $f' \in \mathcal{F}(K')$ essentially means that f' is a nondegenerate self-concordant barrier functional whose barrier parameter is of magnitude $O((K')^2)$.

Let $\hat{A} : X \times \mathbb{R} \to Y$ denote the linear operator defined by

$$\hat{A}(x, s) := Ax - sb.$$

Given $w = (x, s) \in D_{f'}$ and assuming $A\hat{A}^T$ is invertible, one approach to solving the linear equations $Ax = b$ is to apply the cgm to the linear equations (in the variable y)

$$\hat{A}(H'_w)^{-1}\hat{A}^T y = -\hat{A}w,$$

thus obtaining a convergent sequence $y_0 := 0, y_1, \ldots$. Defining

$$w_i = (x_i, s_i) := w + (H'_w)^{-1}\hat{A}y_i, \tag{28}$$

the sequence $\{w_i\}$ then converges to a point $w_\infty = (x_\infty, s_\infty)$ satisfying $\hat{A}w_\infty = 0$. Under certain conditions one will have $s_i, s_\infty > 0$ and hence the sequence $\{\frac{1}{s_i}x_i\}$ will then converge to $\frac{1}{s_\infty}x_\infty$ which will be a solution of $Ax = b$. We now describe such certain conditions which are relevant to solving $Ax = b,\ x \geq 0$.

If $\operatorname{relcond}(A, b) < \infty$, then the constraints $Ax = b,\ x > 0$ are consistent; cf. [6, Cor. 3.7]. Consequently, if $\operatorname{relcond}(A, b) < \infty$, then the constraints

$$\begin{aligned} &\hat{A}w = 0, \\ &w \in D_{f'} \end{aligned} \tag{29}$$

are consistent. Since $D_{f'}$ is bounded and $f' \in \mathcal{F}(K')$, it follows that the convex optimization problem

$$\begin{aligned} \inf\ & f'_w \\ \text{s.t.}\ & \hat{A}w = 0, \\ & w \in D_{f'} \end{aligned}$$

has an optimal solution z', i.e., the constraints (29) have an analytic center.

Now, in the same manner that the Hessian H_w of f at $w = (x, s, t) \in D_f$ induces the inner product $\langle\ ,\ \rangle_w$ and norm $\|\ \|_w$ on $X \times \mathbb{R} \times \mathbb{R}$, the Hessian H'_w of f' at $w = (x, s) \in D_{f'}$ induces an inner product $\langle\ ,\ \rangle'_w$ and norm $\|\ \|'_w$ on $X \times \mathbb{R}$.

THEOREM 1.4. *Assume* $\operatorname{relcond}(A, b) < \infty$. *If* $w \in D_{f'}$ *satisfies*

$$\|z' - w\|'_w = O(1),$$

then

$$(30)\quad \left\|\sqrt{(\hat{A}(H'_w)^{-1}\hat{A}^T)^{-1}}\right\| \leq \frac{O(K^2+1)}{\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}}.$$

Proof. See §4. □

THEOREM 1.5. *Assume* $\operatorname{relcond}(A,b) < \infty$. *If* $w \in D_{f'}$ *and* $z \in X \times \mathbb{R}$ *satisfy*

$$\hat{A}z = 0 \ \text{ and } \ \|z-w\|'_w = O(1),$$

then

$$(31)\quad \left\|\sqrt{(\hat{A}(H'_w)^{-1}\hat{A}^T)^{-1}}\right\| \geq \frac{\Omega(1)}{\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}}.$$

Proof. See §4. □

PROPOSITION 1.6. *If* $w \in D_{f'}$, *then*

$$\left\|\sqrt{\hat{A}(H'_w)^{-1}\hat{A}^T}\right\| = O(\|(A,b)\|).$$

Proof. See §3. □

COROLLARY 1.7. *If* $\operatorname{relcond}(A,b) < \infty$ *and* $w \in D_{f'}$ *satisfies*

$$\|z'-w\|'_w = O(1),$$

then

$$\operatorname{relcond}\left(\sqrt{\hat{A}(H'_w)^{-1}\hat{A}^T}\right) = O(K^2+1)\operatorname{relcond}(A,b),$$

and

$$\begin{aligned}\Omega(1)\operatorname{relcond}(A,b) &\leq \|(A,b)\| \left\|\sqrt{(\hat{A}(H'_w)^{-1}\hat{A}^T)^{-1}}\right\| \\ &\leq O(K^2+1)\operatorname{relcond}(A,b).\end{aligned}$$

Proof. The proof follows immediately from Theorems 1.4 and 1.5 and Proposition 1.6. □

The author considers Corollary 1.7 to be the other of the two most interesting results in this paper.

Corollary 1.7 is used to prove the following theorem.

THEOREM 1.8. *Assume* $\operatorname{relcond}(A,b) < \infty$ *and* $w = (x,s) \in D_{f'}$ *satisfies* $\|z'-w\|'_w = O(1)$. *Then the sequence* $\{w_i\}$ *defined by* (28) *and the limit point* w_∞ *satisfy* $w_i, w_\infty \in D_{f'}$; *in particular,* $\frac{1}{s_i}x_i, \frac{1}{s_\infty}x_\infty > 0$, *and* $A(\frac{1}{s_\infty}x_\infty) = b$.

Moreover, for all ϵ *satisfying* $0 < \epsilon \leq 1$,

$$i = \Omega((K^2+1)\operatorname{relcond}(A,b)\ \log_2 \tfrac{K+2}{\epsilon}) \ \Rightarrow\ \frac{\|\frac{1}{s_\infty}x_\infty - \frac{1}{s_i}x_i\|}{\max\{\|\frac{1}{s_\infty}x_\infty\|, 1\}} \leq \epsilon.$$

Also,

$$\begin{aligned}\|\tfrac{1}{s_\infty}x_\infty\| &= O(K^2+1)(1+\inf\{\|x\|;\ Ax=b \text{ and } x\geq 0\})\\ &\leq O(K^2+1)\ \mathrm{relcond}(A,b),\end{aligned}$$

i.e., $\frac{1}{s_\infty}x_\infty$ *is a not-too-far-from-minimal norm solution for the constraints* $Ax = b,\ x \geq 0$.

Proof. See §6. □

We would like to be in position to apply Theorem 1.8 to (x, s) where (x, s, t) is a point computed by the barrier method. To this end, we have the following theorem.

THEOREM 1.9. *Assume* $\mathrm{relcond}(A, b) < \infty$. *If* $w = (x, s, t) \in D_f$ *satisfies*

$$\mathrm{dist}(w, \mathrm{CentTraj}) = O(1) \ \text{ and } \ t \leq \frac{O(1)}{(K^2+1)\,\mathrm{relcond}(A,b)},$$

then

$$\|z' - (x,s)\|'_{x,s} = O(\mathrm{dist}(w, \mathrm{CentTraj}) + t(K^2+1)\mathrm{relcond}(A,b)).$$

Proof. See §7. □

Theorems 1.8 and 1.9 and (23) readily yield bounds on the number of barrier method iterations sufficient to obtain (x, s, t) for which (x, s) yields a sequence $\{\frac{1}{s_i}x_i\}$ converging to a solution $\frac{1}{s_\infty}x_\infty$ of $Ax = b,\ x > 0$; precisely,

$$O((K+1)\ \log_2(K+1+\mathrm{relcond}(A,b)))$$

barrier method iterations suffice. Hence, if the Newton step at each barrier method iteration is approximated using the cgm then (27) and Theorem 1.8 imply that the total number of cgm iterations (including those used in constructing the sequence $\{\frac{1}{s_i}x_i\}$) sufficient to obtain a point $\frac{1}{s_i}x_i$ which accurately approximates a solution of $Ax = b,\ x > 0$ is roughly

$$O\left((K^3+1)\,\mathrm{relcond}(A,b)\ \log_2(K+2)\ \log_2(K+1+\mathrm{relcond}(A,b))\right). \tag{32}$$

It is instructive to consider the iteration bound (32) in the case that $C = X$, that is, in the case of linear equations $Ax = b$, assuming $\|b\| \leq \|A\|$. (Of course one would never use the barrier method to solve linear equations.) An appropriate functional ϕ is then the functional whose value is identically zero; for this functional, $K = 0$. Hence, the bound (32) becomes

$$O\left(\mathrm{relcond}(A,b)\log(1+\mathrm{relcond}(A,b))\right),$$

which, upon substitution of (11), gives

$$O\left(\mathrm{relcond}(\sqrt{AA^T})\ \log(1+\mathrm{relcond}(\sqrt{AA^T}))\right).$$

Up to a logarithmic factor, this is consistent with the well-known bound (7); the logarithmic factor is precisely the number of barrier method iterations.

If Y is finite-dimensional, i.e., $Y = \mathbb{R}^m$, then since the cgm solves each of the systems of linear equations within m iterations, the total number of cgm iterations can be bounded by

$$O\left((K+1)m\ \log_2(K+1+\mathrm{relcond}(A,b))\right).$$

If $(K^2+1)\mathrm{relcond}(A,b) << m$, then (32) provides a better bound.

2. The Newton step.

2.1. In this section we explain why, assuming $w \in D_f$, the Newton step n_w for the linearly constrained optimization problem

$$\begin{array}{ll} \min_w & \mu\langle c, w\rangle + f_w \\ \text{s.t.} & \bar{A}w = 0 \end{array} \tag{33}$$

satisfies

$$n_w = -H_w^{-1}(\mu c + g_w + \bar{A}^T y), \tag{34}$$

where y solves

$$\bar{A}H_w^{-1}\bar{A}^T y = \bar{A}(w - H_w^{-1}(\mu c + g_w)). \tag{35}$$

Our explanation follows a rather unconventional route, the advantage of this route being that once one understands it, one also understands how the cgm fits into the customary analysis of the barrier method. In particular, one understands how to proceed to bound the number of cgm iterations which are sufficient to obtain an appropriately accurate approximation $\tilde{y}$ to the exact solution of the linear equations (35), that is, a vector $\tilde{y}$ such that if

$$\Delta n_w := -H_w^{-1}(\mu c + g_w + \bar{A}^T\tilde{y}) - n_w,$$

then $\|\Delta n_w\|_w < \gamma$ (i.e., $\|\Delta n_w\|_w$ is small) and hence

$$-H_w^{-1}(\mu c + g_w + \bar{A}^T\tilde{y})$$

is an appropriately accurate approximation to the Newton step n_w.

Our explanation is somewhat unconventional in that it focuses on the inner product $\langle\ ,\ \rangle_w$ rather than the original inner product $\langle\ ,\ \rangle$ on $\mathcal{H}$. Since from a theoretical perspective the natural way to measure the proximity of w to z_μ is in terms of the norm $\|\ \|_w$ which arises from the inner product $\langle\ ,\ \rangle_w$, it is not surprising that an explanation which focuses on the inner product $\langle\ ,\ \rangle_w$ provides deeper insight than an explanation which focuses on the inner product $\langle\ ,\ \rangle$.

2.2. In terms of the inner product $\langle\ ,\ \rangle_w$, the linearly constrained optimization problem (33) becomes

$$\begin{array}{ll} \min_v & \mu\langle \tilde{c}, v\rangle_w + f_v \\ \text{s.t.} & \bar{A}v = 0, \end{array} \tag{36}$$

where $\tilde{c} := H_w^{-1}c$.

Now, H_v denotes the Hessian of f at $v \in D_f$ with respect to the original inner product $\langle\ ,\ \rangle$. However, the Hessian of f at v with respect to the inner product $\langle\ ,\ \rangle_w$ is

$$\tilde{H}_v := H_w^{-1}H_v;$$

in particular, $\tilde{H}_w = I$, the identity operator. Similarly, the gradient of f at v with respect to the inner product $\langle\ ,\ \rangle_w$ is

$$\tilde{g}_v := H_w^{-1}g_v;$$

in particular, $\tilde{g}_w := H_w^{-1} g_w$.

Just as the choice of inner product affects gradients and Hessians, it also affects adjoint operators. In particular, whereas the adjoint of $\bar{A}$ is denoted $\bar{A}^T$ when $\mathcal{H}$ is equipped with the original inner product $\langle\ ,\ \rangle$, if $\mathcal{H}$ is equipped with the inner product $\langle\ ,\ \rangle_w$ then the adjoint is

$$\bar{A}^{\tilde{T}} := H_w^{-1} \bar{A}^T.$$

(We never vary the inner product on Y.)

Finally, of course, the particular inner product affects what is meant by (orthogonal) projection.

Throughout the remainder of §2, *whenever we speak of a gradient, a Hessian, an adjoint, or (orthogonal) projection, we mean with respect to the inner product* $\langle\ ,\ \rangle_w$. *Similarly, whenever a norm on* $\mathcal{H}$ *is referred to, it is the norm* $\|\ \|_w$.

2.3. Let $\tilde{P}$ denote the operator projecting $\mathcal{H}$ onto the nullspace of $\bar{A}$ and let $\tilde{P}_\perp$ denote the operator projecting $\mathcal{H}$ onto the orthogonal complement of the nullspace, that is, onto the range space of $\bar{A}^{\tilde{T}}$.

Since the gradient of the objective functional of (36) at w is $\mu\tilde{c} + \tilde{g}_w$ and since the Hessian is $\tilde{H}_w = I$, the Newton model functional at w for the objective functional of (36) is

$$v \mapsto (\mu\langle \tilde{c}, w\rangle_w + f_w) + \langle \mu\tilde{c} + \tilde{g}_w, v - w\rangle_w + \tfrac{1}{2}\|v - w\|_w^2.$$

Consequently, when the objective functional in (36) is replaced by the Newton model function, the optimal solution of the resulting linearly constrained optimization problem is

$$\begin{aligned} w + n_w &:= \tilde{P}(w - \mu\tilde{c} - \tilde{g}_w) \\ &= w - \mu\tilde{c} - \tilde{g}_w - \tilde{P}_\perp(w - \mu\tilde{c} - \tilde{g}_w). \end{aligned}$$

Hence

$$n_w = -\mu\tilde{c} - \tilde{g}_w - w^*, \tag{37}$$

where

$$w^* := \tilde{P}_\perp(w - \mu\tilde{c} - \tilde{g}_w). \tag{38}$$

Observe that w^* is the minimal norm solution to the following system of linear equations in the variable v:

$$\bar{A}v = \bar{A}(w - \mu\tilde{c} - \tilde{g}_w). \tag{39}$$

Consequently,

$$w^* = \bar{A}^{\tilde{T}} y, \tag{40}$$

where y solves

$$\bar{A}\bar{A}^{\tilde{T}} y = \bar{A}(w - \mu\tilde{c} - \tilde{g}_w). \tag{41}$$

The linear equations (41) are exactly the same as the linear equations (35) and the identity (37) is exactly the same as the identity (34); i.e., we have now derived

the linear equations underlying Newton's method. Moreover, we are in position to understand a bit about the behavior of the cgm when applied to the linear equations (35) (or, equivalently, the linear equations (41)) in order to approximate n_w.

2.4. Recall that for the barrier method theory, one wants to approximate n_w to within error γ (as measured in the norm $\| \ \|_w$), where γ is a certain universal positive constant. To obtain such an approximation, (37) shows that it is sufficient to approximate w^* to within error γ and then to subtract the resulting approximation from $-\mu\tilde{c} - \tilde{g}_w$.

As we observed, w^* is the minimal norm solution to the linear equations (39). Our discussion of the cgm in §1.3 showed that the minimal norm solution for a system of linear equations can be approximated using the cgm. Translating that discussion to the present context, to solve (39) one applies the cgm to solve (41), thus obtaining a sequence $\{y_i\}$; the sequence $\{\bar{A}^{\tilde{T}} y_i\}$ then converges to the minimal norm solution. In other words, when one thinks in terms of the inner product $\langle \ , \ \rangle_w$, our application of the cgm is exactly as prescribed in §1.3. Moreover, in terms of the present context, (7) is rewritten as

$$(42) \quad i = \Omega\left(\text{relcond}\left(\sqrt{\bar{A}\bar{A}^{\tilde{T}}}\right)\log_2 \tfrac{2}{\epsilon}\right) \ \Rightarrow \ \|w^* - \bar{A}^{\tilde{T}} y_i\|_w < \epsilon \|w^* - \bar{A}^{\tilde{T}} y_0\|_w.$$

If we take our initial iterate as $y_0 := 0$, (42) yields

$$(43) \quad i = \Omega\left(\text{relcond}\left(\sqrt{\bar{A}\bar{A}^{\tilde{T}}}\right)\log_2\left(2 + \tfrac{\|w^*\|_w}{\gamma}\right)\right) \ \Rightarrow \ \|w^* - \bar{A}^{\tilde{T}} y_i\|_w < \gamma.$$

To prove the claimed upper bound (27) on the number of cgm iterations which are sufficient to compute an appropriately accurate approximation to the Newton step during an iteration of the barrier method, (43) shows it is sufficient to prove that if

$$\|z_\mu - w\|_w = O(1),$$

then

$$(44) \qquad \text{relcond}\left(\sqrt{\bar{A}\bar{A}^{\tilde{T}}}\right) = O((K^2+1)\,\text{relcond}(A,b)),$$

and

$$(45) \qquad \|w^*\|_w = O(K^2+1).$$

The relation (44) is precisely the content of Corollary 1.3. The relation (45) is established via now-standard arguments; the proof of the following proposition provides an outline of those arguments.

PROPOSITION 2.1. *If* $\|z_\mu - w\|_w = O(1)$ *and* w^* *is the minimal norm solution of the linear equations* (39), *then*

$$\|w^*\|_w = O(K^2+1).$$

Proof. We have

$$\begin{aligned}\|w^*\|_w &= \|\tilde{P}_\perp(w - \mu\tilde{c} - \tilde{g}_w)\|_w \\ &\le \|\tilde{P}_\perp w\|_w + \mu\|\tilde{c}\|_w + \|\tilde{g}_w\|_w \\ &\le \|z_\mu - w\|_w + \mu\|\tilde{c}\|_w + O(\bar{K})\end{aligned}$$

and thus, assuming $\|z_\mu - w\|_w = O(1)$, it is sufficient to show that

$$\mu\|\tilde{c}\|_w = O(K^2 + 1). \tag{46}$$

It follows from the definitions of f (i.e., (16)), $\tilde{c} := H_w^{-1}c$, and $\tilde{c} := (\vec{0}, 0, 1)$ that if $w = (x, s, t)$ then

$$\|\tilde{c}\|_w = \tfrac{1}{3}\left(\tfrac{1}{t^2} + \tfrac{1}{(2-t^2)}\right)^{-1/2}. \tag{47}$$

Also, since

$$\forall v, \quad \|v\|_w = O(1) \quad \Rightarrow \quad w + v \in D_f$$

and

$$(\bar{x}, \bar{s}, \bar{t}) \in D_f \quad \Rightarrow \quad \bar{t} > 0,$$

we have that if $z_\mu = (x_\mu, s_\mu, t_\mu)$ then

$$\|z_\mu - w\|_w = O(1) \quad \Rightarrow \quad t_\mu = \Theta(t). \tag{48}$$

The relations (47) and (48) imply that to prove (46) it is sufficient to show

$$t_\mu = O(\tfrac{1}{\mu}\bar{K}^2),$$

that is,

$$\langle c, z_\mu \rangle = O(\tfrac{1}{\mu}\bar{K}^2). \tag{49}$$

However, for general problems of the form

$$\begin{array}{ll} \inf_v & \langle c, v \rangle \\ \text{s.t.} & \bar{A}v = \bar{b}, \\ & v \in D_f, \end{array} \tag{50}$$

where $f \in \mathcal{F}(\bar{K})$, one has that

$$\langle c, z_\mu \rangle - \inf_{u \in D_f} \langle c, u \rangle = O(\tfrac{1}{\mu}\bar{K}^2)$$

(cf. [5]). Since in our setting we have that

$$\inf_{u \in D_f} \langle c, u \rangle = 0,$$

the relation (49) is indeed valid, completing the proof. □

We now have our cgm iteration bound (27).

3. Proofs of Propositions 1.2 and 1.6.

Proof of Proposition 1.2. First recall that if $w \in D_f$ then

$$\forall v, \quad \|v - w\|_w = O(1) \quad \Rightarrow \quad v \in D_f$$

and hence, by definition of the norm $\| \ \|_w$,

$$\forall u, \quad \|u\| = O(1) \quad \Rightarrow \quad w + \sqrt{H_w^{-1}}u \in D_f.$$

It follows that

$$\left\|\sqrt{H_w^{-1}}\right\| = O(\text{diam}(D_f)),$$

where

$$\text{diam}(D_f) := \sup\{\|u - v\|;\ u, v \in D_f\}.$$

Since, for the particular domain D_f of interest to us,

$$D_f \subseteq \{(x, s, t);\ \|x\| < 1,\ 0 < s < 1 \text{ and } 0 < t < 2\},$$

we thus have

$$\left\|\sqrt{H_w^{-1}}\right\| = O(1)$$

and hence

$$\|H_w^{-1}\| = O(1).$$

Consequently,

$$\begin{aligned}\left\|\sqrt{\bar{A}H_w^{-1}\bar{A}^T}\right\| &= \sqrt{\|\bar{A}H_w^{-1}\bar{A}^T\|} \\ &\leq \sqrt{\|\bar{A}\|\,\|H_w^{-1}\|\,\|\bar{A}^T\|} \\ &= O\left(\sqrt{\|\bar{A}\|\,\|\bar{A}^T\|}\right) \\ &\leq O(\|\bar{A}\|) \\ &\leq O(\|(A, b)\|),\end{aligned}$$

the final equality being a simple consequence of the definition of $\bar{A}$ (and the bound $\|\check{x}\| < 1$). □

Proof of Proposition 1.6. The proof of this proposition is entirely analogous to the proof of Proposition 1.2 and is thus left to the reader. □

4. Proofs of Theorems 1.4 and 1.5.

4.1. In proving Theorems 1.4 and 1.5 we rely on the inner product $\langle\ ,\ \rangle'_w$ for which the adjoint of $\hat{A}$ is

$$\hat{A}^{T'} := (H'_w)^{-1}\hat{A}^T.$$

A system $Ax = b$, $x \geq 0$ and vector $\bar{b} \in Y$ give rise to the following constraints in the variables $(x, s) \in X \times \mathbb{R}$:

$$\begin{aligned} Ax &= sb + \bar{b}, \\ x &> 0, \\ s &> 0, \\ \|x\| + s &< 1. \end{aligned} \tag{51}$$

The proofs of Theorems 1.4 and 1.5 depend heavily on the following theorem regarding the different systems (51) obtained by varying $\bar{b}$.

THEOREM 4.1.

$$\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}$$
$$= \inf\{\|\bar{b}\|;\ \bar{b} \in Y \text{ and the constraints (51) are inconsistent}\}.$$

Proof. This is a special case of [6, Thm. 3.5]; the main ingredient in the proof is the Hahn–Banach theorem. □

4.2. Proof of Theorem 1.5. We wish to prove that if $w \in D_{f'}$ and z satisfy

$$\hat{A}z = 0 \text{ and } \|z-w\|'_w = O(1), \tag{52}$$

then

$$\left\|\sqrt{(\hat{A}\hat{A}^{T'})^{-1}}\right\| \geq \frac{\Omega(1)}{\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}}.$$

Recalling the well-known (and easily proven) Hilbert-space relation (which is based solely on the fact that $\hat{A}^{T'}$ is the adjoint of $\hat{A}$)

$$\frac{1}{\left\|\sqrt{(\hat{A}\hat{A}^{T'})^{-1}}\right\|} = \inf\{\|\bar{b}\|;\ \|v\|'_w < 1 \Rightarrow \hat{A}v \neq \bar{b}\},$$

we see that it is sufficient to show that if w and z satisfy (52) then

$$\inf\{\|\bar{b}\|;\ \|v\|'_w < 1 \Rightarrow \hat{A}v \neq \bar{b}\}$$
$$\leq O(1)\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}. \tag{53}$$

In proving the inequality (53) we may assume that $w = z$. For if

$$\|z-w\|'_w = O(1),$$

then $z \in D_{f'}$ and for all v we have

$$\|v\|'_z = O(\|v\|'_w);$$

thus

$$\inf\{\|\bar{b}\|;\ \|v\|'_w < 1 \Rightarrow \hat{A}v \neq \bar{b}\} = O(\inf\{\|\bar{b}\|;\ \|v\|'_z < 1 \Rightarrow \hat{A}v \neq \bar{b}\}).$$

Consequently, if the inequality (53) is valid for $w = z$ then it is valid for all $w \in D_{f'}$ satisfying $\|z-w\|'_w = O(1)$ (but perhaps with a larger universal constant hidden in the big-O).

Since $\hat{A}z = 0$, when $w = z$ the quantity on the left of the inequality (53) equals

$$\inf\{\|\bar{b}\|;\ \|u-z\|'_z < 1 \Rightarrow \hat{A}u \neq \bar{b}\}. \tag{54}$$

Since

$$\|u-z\|'_z = O(1) \Rightarrow u \in D_{f'},$$

the quantity (54) is bounded above by

$$O(1)\inf\{\|\bar{b}\|;\ u \in D_{f'} \Rightarrow \hat{A}u \neq \bar{b}\}.$$

Consequently, to establish the inequality (53) when $w = z$ it is sufficient to show that

$$\begin{aligned} &\inf\{\|\bar{b}\|;\ u \in D_{f'} \ \Rightarrow\ \hat{A}u \neq \bar{b}\} \\ &\qquad \leq O(1)\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\}. \end{aligned} \tag{55}$$

For $\bar{b} \in Y$, consider the following constraints in the variables (x, s):

$$\begin{aligned} Ax &= sb + \bar{b}, \\ x &> 0, \\ s &> 0, \\ \|x\| + s &< 2. \end{aligned} \tag{56}$$

Note that each element of $D_{f'}$ satisfies the constraints $x > 0$, $s > 0$, and $\|x\| + s < 2$. Hence,

$$\begin{aligned} &\inf\{\|\bar{b}\|;\ u \in D_{f'} \ \Rightarrow\ \hat{A}u \neq \bar{b}\} \\ &\qquad \leq \inf\{\|\bar{b}\|;\ \bar{b} \in Y \text{ and the constraints (56) are inconsistent}\}. \end{aligned} \tag{57}$$

Observe that the quantity on the right of the inequality (57) is exactly twice the quantity

$$\inf\{\|\bar{b}\|;\ \bar{b} \in Y \text{ and the constraints (51) are inconsistent}\}.$$

Hence, (57) and Theorem 4.1 give the desired inequality (55). □

4.3. The proof of Theorem 1.4 depends on the following proposition.

PROPOSITION 4.2. *Let $z' = (x', s')$. If $\bar{b} \in Y$ satisfies*

$$\forall v,\ \|v\|'_{z'} < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}$$

and if $r \in \mathbb{R}$ satisfies $r = \Omega(K^2 + 1)$, then either

$$\hat{A}w \neq r\bar{b} \text{ for all } w \in D_{f'}$$

or

$$\hat{A}w \neq r\bar{b} \text{ for all } w \in D_{f'}.$$

Proof. See §4.4. □

Proof of Theorem 1.4. We wish to prove that if $\|z' - w\|'_w = O(1)$ then

$$\left\|\sqrt{(\hat{A}\hat{A}^{T'})^{-1}}\right\| \leq \frac{O(K^2 + 1)}{\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\}}.$$

Recalling the well-known (and easily proven) Hilbert-space relation (which is based solely on the fact that $\hat{A}^{T'}$ is the adjoint of $\hat{A}$)

$$\frac{1}{\left\|\sqrt{(\hat{A}\hat{A}^{T'})^{-1}}\right\|} = \inf\{\|\bar{b}\|;\ \|v\|'_w < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}\},$$

we see that it is sufficient to show that if $\|z' - w\|'_w = O(1)$ then

$$\begin{aligned} &\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\} \\ &\qquad \leq O(K^2 + 1)\inf\{\|\bar{b}\|;\ \|v\|'_w < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}\}. \end{aligned} \tag{58}$$

In proving the inequality (58) we may assume that $w = z'$. For if $\|z' - w\|'_w = O(1)$, then for all v we have $\|v\|'_w = O(\|v\|'_{z'})$ and hence

$$\begin{aligned} &\inf\{\|\bar{b}\|;\ \|v\|'_{z'} < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}\} \\ &\quad = O(\inf\{\|\bar{b}\|;\ \|v\|'_w < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}\}). \end{aligned} \tag{59}$$

Consequently, if the inequality (58) is valid for $w = z'$ then it is valid for all w satisfying $\|z' - w\|'_w = O(1)$ (but perhaps with a larger universal constant hidden in the big-O).

Proposition 4.2 implies that there exists $\hat{b} \in Y$ such that

$$\hat{A}w \neq \hat{b} \text{ for all } w \in D_{f'} \tag{60}$$

and

$$\|\hat{b}\| \leq O(K^2 + 1) \inf\{\|\bar{b}\|;\ \|v\|'_{z'} < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}\}. \tag{61}$$

The definitions of $\hat{A}$ and $D_{f'}$, along with (60), imply that the following system of constraints in the variables $(x, s) \in X \times \mathbb{R}$ is inconsistent:

$$\begin{gathered} Ax = sb + \hat{b}, \\ x > 0, \\ \|x\| < 1, \\ 0 < s < 1. \end{gathered}$$

Consequently, Theorem 4.1 gives

$$\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\} \leq \|\hat{b}\|. \tag{62}$$

Together, (61) and (62) yield the inequality (58) when $w = z'$, which completes the proof. □

4.4. Proof of Proposition 4.2. Letting $g'_{z'}$ denote the gradient of f' at z', consider the functional

$$f''_w := f'_w - 12\ln(\langle g'_{z'}, w - z'\rangle + 12),$$

whose domain is the set

$$D_{f''} := \{w \in D_{f'};\ \langle g'_{z'}, w - z'\rangle > -12\}. \tag{63}$$

Recalling that $f' \in \mathcal{F}(K')$ (where $K' = K + O(1)$), in the terminology of [5] and [6] we have $f'' \in \mathcal{F}(\sqrt{(K')^2 + 12})$; in the terminology of Nesterov and Nemirovskii [3], f'' is essentially a nondegenerate self-concordant barrier functional whose barrier parameter is of magnitude $O(K^2 + 1)$.

We let g''_w, H''_w, $\langle\ ,\ \rangle''_w$, and $\|\ \|''_w$ denote, respectively, the gradient, the Hessian, the inner product, and the norm at w with respect to the functional f''.

We claim that to prove the proposition it is sufficient to prove the following statement (which we will refer to simply as Statement 1): *If $\bar{b} \in Y$ satisfies*

$$\forall v,\ \ \|v\|''_{z'} < 1 \ \Rightarrow\ \hat{A}v \neq \bar{b}$$

and if $r \in \mathbb{R}$ satisfies $r = \Omega(K^2 + 1)$, then

$$\hat{A}w \neq r\bar{b} \text{ for all } w \in D_{f''}.$$

To understand why it is sufficient to prove Statement 1, assume that $\bar{b} \in Y$ satisfies the assumption of the proposition, that is,

$$\forall v, \quad \|v\|'_{z'} < 1 \quad \Rightarrow \quad \hat{A}v \neq \bar{b}.$$

Since $\|v\|'_{z'} \leq \|v\|''_{z'}$ for all v, we then have

$$\forall v, \quad \|v\|''_{z'} < 1 \quad \Rightarrow \quad \hat{A}v \neq \bar{b}.$$

Hence, if $r = \Omega(K^2 + 1)$ then according to Statement 1,

$$\hat{A}w \neq r\bar{b} \text{ for all } w \in D_{f''}. \tag{64}$$

Now, since z' minimizes f' subject to the constraints $\hat{A}x = 0$, we have that $g'_{z'}$ lies in the range space of $\hat{A}^T$. Hence, for any $r \in \mathbb{R}$, either

$$\forall w, \quad \hat{A}w = r\bar{b} \quad \Rightarrow \quad \langle g'_{z'}, w - z' \rangle \geq 0 \tag{65}$$

or

$$\forall w, \quad \hat{A}w = -r\bar{b} \quad \Rightarrow \quad \langle g'_{z'}, w - z' \rangle \geq 0. \tag{66}$$

If (65) is valid then (63) and (64) imply

$$\hat{A}w \neq r\bar{b} \text{ for all } w \in D_{f'}.$$

Similarly, if (66) is valid then (63) and (64) imply

$$\hat{A}w \neq -r\bar{b} \text{ for all } w \in D_{f'}.$$

Consequently, the proposition does indeed follow from Statement 1.

To prove Statement 1, first observe that $g''_{z'} = 0$. Hence z' minimizes the strictly convex functional f''; i.e., z' is the analytic center of the domain $D_{f''}$. Thus, since $f'' \in \mathcal{F}((K')^2 + 12)$ and $K' = K + O(1)$, we have

$$w \in D_{f''} \quad \Rightarrow \quad \|w - z'\|''_{z'} = O(K^2 + 1). \tag{67}$$

Assume that $\bar{b}$ satisfies the assumption of Statement 1, that is,

$$\forall v, \quad \|v\|''_{z'} < 1 \quad \Rightarrow \quad \hat{A}v \neq \bar{b}. \tag{68}$$

Of course if $r \in \mathbb{R}$ then (68) implies

$$\forall v, \quad \|v\|''_{z'} < r \quad \Rightarrow \quad \hat{A}v \neq r\bar{b}. \tag{69}$$

Finally, since $\hat{A}z' = 0$, (67) and (69) imply that if $r = \Omega(K^2 + 1)$, then

$$\hat{A}w \neq r\bar{b} \text{ for all } w \in D_{f''},$$

concluding the proof of Statement 1 and hence the proof of the proposition. □

5. Proof of Theorem 1.1.

5.1. The proof of Theorem 1.1 is very similar to, but slightly more complicated than, the proof of Theorem 1.4 given in §4. The proof of Theorem 1.1 depends on Theorem 4.1 and the following proposition (which plays a role analogous to that of Proposition 4.2 in the proof of Theorem 1.4).

PROPOSITION 5.1. *Let* $z_\mu = (x_\mu, s_\mu, t_\mu)$. *If* $\bar{b} \in Y$ *satisfies*

$$\forall v, \quad \|v\|_{z_\mu} < 1 \quad \Rightarrow \quad \bar{A}v \neq \bar{b}$$

and if $r \in \mathbb{R}$ *satisfies* $r = \Omega(K^2 + 1)$, *then either*

$$\bar{A}w \neq r\bar{b} \text{ for all } w = (x, s, t) \in D_f \text{ satisfying } t \leq t_\mu$$

or

$$\bar{A}w \neq -r\bar{b} \text{ for all } w = (x, s, t) \in D_f \text{ satisfying } t \leq t_\mu.$$

Proof. See §5.2. □

Proof of Theorem 1.1. We rely on the inner product $\langle\ ,\ \rangle_w$ for which the adjoint of $\bar{A}$ is

$$\bar{A}^{\tilde{T}} := H_w^{-1}\bar{A}^T.$$

We wish to prove that if $\operatorname{dist}(w, \text{CentTraj}) = O(1)$ then

$$\left\|\sqrt{(\bar{A}\bar{A}^{\tilde{T}})^{-1}}\right\| = \frac{O(K^2+1)}{\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\}}.$$

Recalling the well-known (and easily proven) Hilbert-space relation (which is based solely on the fact that $\bar{A}^{\tilde{T}}$ is the adjoint of $\bar{A}$)

$$\frac{1}{\|\sqrt{(\bar{A}\bar{A}^{\tilde{T}})^{-1}}\|} = \inf\{\|\bar{b}\|;\ \|v\|_w < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}\},$$

we see that it is sufficient to show that if $\operatorname{dist}(w, \text{CentTraj}) = O(1)$ then

$$\begin{aligned} &\inf\{\|(\Delta A, \Delta b)\|;\ (A+\Delta A, b+\Delta b) \text{ is inconsistent}\} \\ &\quad \leq O(K^2+1)\ \inf\{\|\bar{b}\|;\ \|v\|_w < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}\}. \end{aligned} \tag{70}$$

In proving the inequality (70) we may assume that $w = z_\mu \in \text{CentTraj}$. For if $\|z_\mu - w\|_w = O(1)$, then for all v we have $\|v\|_w = O(\|v\|_{z_\mu})$; hence,

$$\begin{aligned} &\inf\{\|\bar{b}\|;\ \|v\|_{z_\mu} < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}\} \\ &\quad = O(\inf\{\|\bar{b}\|;\ \|v\|_w < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}\}). \end{aligned} \tag{71}$$

Consequently, if the inequality (70) is valid for $w = z_\mu \in \text{CentTraj}$, then it is valid for all w satisfying $\operatorname{dist}(w, \text{CentTraj}) = O(1)$ (but perhaps with a larger universal constant hidden in the big- O).

Proposition 5.1 implies that there exists $\hat{b} \in Y$ such that

$$\bar{A}w \neq \hat{b} \text{ for all } w = (x, s, t) \in D_f \text{ satisfying } t \leq t_\mu \tag{72}$$

and

$$\|\hat{b}\| \leq O(K^2+1)\ \inf\{\|\bar{b}\|;\ \|v\|_{z_\mu} < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}\}. \tag{73}$$

The definitions of $\bar{A}$ and D_f, along with (72), imply that the following system of constraints in the variables $(x,s) \in X \times \mathbb{R}$ is inconsistent if $0 < t \leq t_\mu$:

$$\begin{gathered} Ax = sb + t(A\check{x} - \tfrac{1}{2}b) + \hat{b}, \\ x > 0, \\ \|x\| < 1, \\ 0 < s < 1. \end{gathered}$$

Consequently, Theorem 4.1 shows that if $0 < t \leq t_\mu$, then

$$\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\} \leq \|t(A\check{x} - \tfrac{1}{2}b) + \hat{b}\|;$$

thus, letting $t \downarrow 0$,

$$\inf\{\|(\Delta A, \Delta b)\|;\ (A + \Delta A, b + \Delta b) \text{ is inconsistent}\} \leq \|\hat{b}\|. \tag{74}$$

Together, (73) and (74) yield the inequality (70) when $w = z_\mu$, which completes the proof. □

5.2. Proof of Proposition 5.1. Consider the functional

$$f''_w := f_w - 12\ln(-\langle c, w - z_\mu\rangle + \tfrac{12}{\mu})$$

whose domain is the set

$$D_{f''} := \{w = (x,s,t) \in D_f;\ t < t_\mu + \tfrac{12}{\mu}\}. \tag{75}$$

Recalling that $f \in \mathcal{F}(\bar{K})$ (where $\bar{K} = K + O(1)$), in the terminology of [5] and [6] we have $f'' \in \mathcal{F}(\sqrt{\bar{K}^2 + 12})$; in the terminology of Nesterov and Nemirovskii [3], f'' is essentially a nondegenerate self-concordant barrier functional whose barrier parameter is of magnitude $O(K^2+1)$.

We let g''_w, H''_w, $\langle\ ,\ \rangle''_w$, and $\|\ \|''_w$ denote, respectively, the gradient, the Hessian, the inner product, and the norm at w with respect to the functional f''.

Observe that if $w \in D_{f''}$, then $\|v\|_w \leq \|v\|''_w$ for all v. Together with (75) this implies that to prove the proposition, it is sufficient to prove the following statement (which we will refer to as Statement 2): *If $\bar{b} \in Y$ has the property that*

$$\forall v,\ \ \|v\|''_{z_\mu} < 1 \ \ \Rightarrow\ \ \bar{A}v \neq \bar{b}$$

and if $r \in \mathbb{R}$ satisfies $r = \Omega(K^2+1)$, then either

$$\bar{A}w \neq r\bar{b} \ \textit{for all}\ w \in D_{f''}$$

or

$$\bar{A}w \neq -r\bar{b} \ \textit{for all}\ w \in D_{f''}.$$

Note that $g''_{z_\mu} = \mu c + g_{z_\mu}$, where g_{z_μ} denotes the gradient of f at z_μ. Consequently, since z_μ is optimal for the linearly constrained convex optimization problem

$$\begin{array}{ll} \min & \mu\langle c, w\rangle + f_w \\ \text{s.t.} & \bar{A}w = 0, \end{array}$$

we have that g''_{z_μ} lies in the range space of $\bar{A}^T$.

Now consider the functional

$$f'''_w := f''_w - 12\ln(\langle g''_{z_\mu}, w - z_\mu\rangle + 12)$$

whose domain is the set

$$D_{f'''} := \{w \in D_{f''};\ \langle g''_{z_\mu}, w - z_\mu\rangle > -12\}. \tag{76}$$

In the terminology of [5] and [6], $f''' \in \mathcal{F}(\sqrt{K^2 + 24})$; in the terminology of Nesterov and Nemirovskii [3], f''' is essentially a nondegenerate self-concordant barrier functional whose barrier parameter is $O(K^2 + 1)$.

We let g'''_w, H'''_w, $\langle\ ,\ \rangle'''_w$, and $\|\ \|'''_w$ denote, respectively, the gradient, the Hessian, the inner product, and the norm at w with respect to the functional f'''.

We claim that to prove Statement 2 it is sufficient to prove the following statement (which we will refer to as Statement 3): *If $\bar{b} \in Y$ satisfies*

$$\forall v,\ \|v\|'''_{z_\mu} < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}$$

and if $r \in \mathbb{R}$ satisfies $r = \Omega(K^2 + 1)$, then

$$\bar{A}w \neq t\bar{b} \text{ for all } w \in D_{f'''}.$$

To understand why Statement 3 implies Statement 2, assume that $\bar{b} \in Y$ satisfies the assumption of Statement 2, that is,

$$\forall v,\ \|v\|''_{z_\mu} < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}.$$

Since $\|v\|'''_{z_\mu} \leq \|v\|''_{z_\mu}$ for all v, we then have

$$\forall v,\ \|v\|'''_{z_\mu} < 1 \ \Rightarrow\ \bar{A}v \neq \bar{b}.$$

Hence, if $r = \Omega(K^2 + 1)$, then according to Statement 3,

$$\bar{A}w \neq t\bar{b} \text{ for all } w \in D_{f'''}. \tag{77}$$

Now, since g''_{z_μ} lies in the range space of $\bar{A}^T$, given $r \in \mathbb{R}$ we have that either

$$\forall w,\ \bar{A}w = r\bar{b} \ \Rightarrow\ \langle g''_{z_\mu}, w - z_\mu\rangle \geq 0 \tag{78}$$

or

$$\forall w,\ \bar{A}w = -r\bar{b} \ \Rightarrow\ \langle g''_{z_\mu}, w - z_\mu\rangle \geq 0. \tag{79}$$

If (78) is valid then (76) and (77) imply

$$\bar{A}w \neq r\bar{b} \text{ for all } w \in D_{f''}.$$

Similarly, if (79) is valid then (76) and (77) imply

$$\bar{A}w \neq -r\bar{b} \text{ for all } w \in D_{f'''}.$$

Consequently, Statement 2 does indeed follow from Statement 3.

To prove Statement 3, first observe that $g'''_{z_\mu} = 0$. Hence z_μ minimizes the strictly convex functional f'''; i.e., z_μ is the analytic center of the domain $D_{f'''}$. Thus, since $f \in \mathcal{F}(\bar{K}^2 + 24)$ and $\bar{K} = K + O(1)$, we have

$$w \in D_{f'''} \Rightarrow \|w - z_\mu\|'''_{z_\mu} = O(K^2 + 1). \tag{80}$$

Assume that $\bar{b}$ satisfies the assumption of Statement 3, that is,

$$\forall v, \ \|v\|'''_{z_\mu} < 1 \Rightarrow \bar{A}v \neq \bar{b}. \tag{81}$$

Of course if $r \in \mathbb{R}$ then (81) implies

$$\forall v, \ \|v\|'''_{z_\mu} < r \Rightarrow \bar{A}v \neq r\bar{b}. \tag{82}$$

Finally, since $\bar{A}z_\mu = 0$, (80) and (82) imply that if $r = \Omega(K^2 + 1)$ then

$$\bar{A}w \neq r\bar{b} \text{ for all } w \in D_{f'''},$$

concluding the proof of Statement 3 and hence the proof of the proposition. □

6. Proof of Theorem 1.8.

6.1. Recall that in the context of Theorem 1.8, we are given $w = (x, s) \in X \times \mathbb{R}$ and are considering the sequence

$$w_i = (x_i, s_i) := w + (H'_w)^{-1}\hat{A}y_i, \tag{83}$$

where $y_0 := 0, y_1, \ldots$ is the sequence obtained by applying the cgm to the linear equations (in the variable y)

$$\hat{A}(H'_w)^{-1}\hat{A}^T y = -\hat{A}w. \tag{84}$$

As discussed in §1, if $\text{relcond}(A, b) < \infty$ then the constraints

$$\begin{aligned} &\hat{A}v = 0, \\ &v \in D_{f'} \end{aligned}$$

have an analytic center, that is, a solution z' of the following optimization problem:

$$\begin{aligned} \inf \quad & f'_v \\ \text{s.t.} \quad & \hat{A}v = 0, \\ & v \in D_{f'}. \end{aligned} \tag{85}$$

Our proof of Theorem 1.8 depends on the following two propositions.

PROPOSITION 6.1. *Assume that $w = (x, s) \in D_{f'}$ and the linear operator*

$$\hat{A}(H'_w)^{-1}\hat{A}^T$$

is invertible; hence the sequence $\{w_i\}$ given by (83) is well defined and has limit w_∞ satisfying $\hat{A}w_\infty = 0$.

If z is any point satisfying $\hat{A}z = 0$ and $\|z - w\|'_w = O(1)$, then

$$w_i, w_\infty, z \in D_{f'}, \quad \|w_\infty - z\|'_z = O(\|z - w\|'_w)$$

and

$$i = \Omega\left(\operatorname{relcond}\left(\sqrt{\hat{A}(H'_w)^{-1}\hat{A}^T}\right)\log_2 \tfrac{2}{\epsilon}\right) \Rightarrow \|\tfrac{1}{s_\infty}x_\infty - \tfrac{1}{s_i}x_i\| \leq \tfrac{\epsilon}{s_\infty}\|z - w\|'_w$$

for all ϵ satisfying $0 < \epsilon \leq 1$.

Proof. See §6.2. □

PROPOSITION 6.2. *Assume* $\operatorname{relcond}(A, b) < \infty$ *and let z' denote the optimal solution for the optimization problem* (85). *If $\bar{w} = (\bar{x}, \bar{s})$ is an arbitrary point satisfying $\|\bar{w} - z'\|'_{z'} = O(1)$, then*

$$\|\tfrac{1}{\bar{s}}\bar{x}\| \leq O(K^2 + 1)(1 + \inf\{\|x\|;\ Ax = b \text{ and } x \geq 0\}) \tag{86}$$

$$\leq O(K^2 + 1)\operatorname{relcond}(A, b) \tag{87}$$

and

$$\tfrac{1}{\bar{s}} \leq O(K^2 + 1)\,(\|\tfrac{1}{\bar{s}}\bar{x}\| + 1). \tag{88}$$

Proof. See §6.5. □

Proof of Theorem 1.8. The proof of this theorem follows immediately from Corollary 1.7 and Propositions 6.1 and 6.2, letting z in Proposition 6.1 be z' and letting $\bar{w}$ in Proposition 6.2 be w_∞. □

6.2. The proof of Proposition 6.1 depends on the following proposition and lemma; the lemma will also be used in the proof of Proposition 6.2.

PROPOSITION 6.3. *Assume the same conditions as in Proposition* 6.1 *but require no bound on $\|z - w\|'_w$. Then for all i,*

$$\|w_\infty - w_i\|'_w \leq \|z - w\|'_w \tag{89}$$

and for all ϵ satisfying $0 < \epsilon \leq 1$,

$$i = \Omega\left(\operatorname{relcond}\left(\sqrt{\hat{A}(H'_w)^{-1}\hat{A}^T}\right)\log_2 \tfrac{2}{\epsilon}\right) \Rightarrow \|w_\infty - w_i\|'_w \leq \epsilon\|z - w\|'_w. \tag{90}$$

Proof. See §6.3. □

LEMMA 6.4. *If $\bar{w} = (\bar{x}, \bar{s})$ and $\hat{w} = (\hat{x}, \hat{s})$ satisfy*

$$\|\bar{w} - w\|'_w = O(1) \text{ and } \|\hat{w} - w\|'_w = O(1),$$

where $w = (x, s) \in D_{f'}$, then $\bar{w}, \hat{w} \in D_{f'}$,

$$\|\tfrac{1}{\bar{s}}\bar{x} - \tfrac{1}{\hat{s}}\hat{x}\| = O(\tfrac{1}{\bar{s}}\|\bar{w} - \hat{w}\|'_w), \text{ and } \hat{s} = \Theta(\bar{s}).$$

Proof. See §6.4. □

Proof of Proposition 6.1. First note that since $w_0 = w$, (89) implies

$$\|w_\infty - w\|'_w \leq \|z - w\|'_w \text{ and } \|w_i - w\|'_w \leq 2\|z - w\|'_w. \tag{91}$$

Consequently, if $\|z - w\|'_w = O(1)$ then $w_i, w_\infty \in D_{f'}$.

Observe that (91) implies

$$\|w_\infty - z\|'_w = O(\|z - w\|'_w).$$

Since

$$\|z - w\|'_w = O(1) \;\Rightarrow\; \|v\|'_z = O(\|v\|'_w) \text{ for all } v,$$

it follows that if $\|z - w\|'_w = O(1)$ then

$$\|w_\infty - z\|'_z = O(\|z - w\|'_w).$$

For the remaining assertions of the proposition, combine Proposition 6.3 and Lemma 6.4, replacing the points $\bar{w}$, $\hat{w}$ in the lemma with w_∞, w_i, respectively. □

6.3. Proof of Proposition 6.3. Let w^* denote the minimal norm solution (with respect to the norm $\| \ \|'_w$) to the linear equations (in the variable v)

$$\hat{A}v = -\hat{A}w.$$

Note that $\|w^*\|_w \leq \|z - w\|_w$ since $\hat{A}(z - w) = -\hat{A}w$.

The adjoint of $\hat{A}$ with respect to the inner product $\langle \ , \ \rangle'_w$ is

$$\hat{A}^{T'} := (H'_w)^{-1}\hat{A}^T.$$

Consequently, the linear equations (84) can be rewritten as

$$\hat{A}\hat{A}^{T'}y = -\hat{A}w.$$

By considering the linear equations rewritten in this form we see that (8) implies that for all i,

$$\|w^* - \hat{A}^{T'}y_i\|'_w \leq \|w^*\|'_w. \tag{92}$$

Similarly, (7) gives

$$i = \Omega\left(\text{relcond}\left(\sqrt{\hat{A}\hat{A}^{T'}}\right)\log_2 \tfrac{2}{\epsilon}\right) \;\Rightarrow\; \|w^* - \hat{A}^{T'}y_i'\|'_w < \epsilon\|w^*\|'_w \tag{93}$$

for all ϵ satisfying $0 < \epsilon \leq 1$.

Observe that (93) implies $w_\infty = w + w^*$ and hence (92) implies

$$\|w_\infty - w_i\|'_w \leq \|w^*\|'_w,$$

whereas (93) gives

$$i = \Omega\left(\text{relcond}\left(\sqrt{\hat{A}\hat{A}^{T'}}\right)\log_2 \tfrac{2}{\epsilon}\right) \;\Rightarrow\; \|w_\infty - w_i\|'_w < \epsilon\|w^*\|'_w.$$

Since $\|w^*\|'_w \leq \|z - w\|'_w$, the proof is complete. □

6.4. Proof of Lemma 6.4. Since $f' \in \mathcal{F}(K')$ and $w \in D_{f'}$ we have that for all u,

$$\|u - w\|'_w = O(1) \;\Rightarrow\; [u \in D_{f'} \text{ and for all } v,\ \|v\|'_u = \Theta(\|v\|'_w)]\,.$$

Consequently, in proving the lemma we may assume that $\hat{w} = w$.

Since

$$\|v\|'_w = O(1) \quad \Rightarrow \quad w + v \in D_{f'}$$

and

$$(x', s') \in D_{f'} \quad \Rightarrow \quad [\|x'\| < 1 \text{ and } 0 < s' < 1],$$

one can easily prove that for all $\tilde{v} = (\tilde{x}, \tilde{s})$,

$$\|\tilde{x}\| = O(\|\tilde{v}\|'_w) \text{ and } \tilde{s} = O(s\|\tilde{v}\|'_w).$$

Hence,

$$\|\bar{x} - x\| = O(\|\bar{w} - w\|'_w) \text{ and } |\bar{s} - s| = O(s\|\bar{w} - w\|'_w).$$

Thus, if $\|\bar{w} - w\|'_w = O(1)$ then

$$\begin{aligned} \|\tfrac{1}{\bar{s}}\bar{x} - \tfrac{1}{s}x\| &= \tfrac{1}{\bar{s}}\|\bar{x} - x + \tfrac{\bar{s}-s}{s}x\| \\ &\le \tfrac{1}{\bar{s}}\left(\|\bar{x} - x\| + \tfrac{\bar{s}-s}{s}\right) \\ &= O(\tfrac{1}{\bar{s}}\|\bar{w} - w\|'_w), \end{aligned}$$

completing the proof. □

6.5. The proof of Proposition 6.2 depends on the following lemma.

LEMMA 6.5. *Assume* $w = (x, s) \in D_{f'}$ *and* $\frac{1}{s} = O(\|\frac{1}{s}x\| + 1)$. *If* $\hat{w} = (\hat{x}, \hat{s})$ *satisfies* $\|\hat{w} - w\|_w = O(1)$, *then*

$$\|\tfrac{1}{\hat{s}}\hat{x}\| = O(\|\tfrac{1}{s}x\| + 1) \tag{94}$$

and

$$\tfrac{1}{\hat{s}} = O(\|\tfrac{1}{\hat{s}}\hat{x}\| + 1). \tag{95}$$

Proof. See §6.6. □

Proof of Proposition 6.2. Lemma 6.5, applied with $w = z'$ and $\hat{w} = \bar{w}$, implies that in proving the proposition we may assume $\bar{w} = z'$. Thus, letting $z' = (x', s')$, it is sufficient to show

$$\|\tfrac{1}{s'}x'\| \le O(K^2 + 1)(1 + \inf\{\|x\|;\ Ax = b \text{ and } x \ge 0\}), \tag{96}$$

$$\tfrac{1}{s'} \le O(K^2 + 1)\,(\|\tfrac{1}{s'}x'\| + 1), \tag{97}$$

and (since $\text{relcond}(A, b) \ge 1$)

$$\inf\{\|x\|;\ Ax = b \text{ and } x \ge 0\}) \le \text{relcond}(A, b). \tag{98}$$

The inequality (98) is a special case of [4, Thm. 1.1(1)]. Hence we need only prove (96) and (97).

The assumption $\text{relcond}(A, b) < \infty$ implies that the set of points which satisfies the strict constraints $Ax = b$, $x > 0$ is dense in the set of points satisfying the

constraints $Ax = b,\ x \geq 0$; cf. [6, Lem. 3.4]. Consequently, assuming that $\tilde{x}$ is an arbitrary point satisfying the strict constraints, to prove (96) it is sufficient to show

$$\|\tfrac{1}{s'}x'\| \leq O(K^2+1)\,(\|\tilde{x}\|+1). \tag{99}$$

Observe that

$$\hat{w} = (\hat{x}, \hat{s}) := \tfrac{1}{2(\|\tilde{x}\|+1)}(\tilde{x}, 1)$$

satisfies the constraints

$$\begin{aligned} &\hat{A}w = 0, \\ &w \in D_{f'}; \end{aligned} \tag{100}$$

i.e., $\hat{w}$ is an element of the region for which z' is the analytic center. Since $s > 0$ is among the constraints defining (100), it follows that

$$\hat{s} = O((K')^2 s').$$

Thus,

$$\|\tfrac{1}{s'}x'\| \leq \tfrac{1}{s'} = O((K')^2 \tfrac{1}{\hat{s}}).$$

Since $\frac{1}{\hat{s}} = 2(\|\tilde{x}\|+1)$ and $K' = K + O(1)$, the proof of (99) is complete; thus, the proof of (96) is complete.

Let $z'' = (x'', s'')$ be a positive multiple of z' with the property that z'' lies in the bounded region defined by the constraints (100) but $2z''$ does not. Since z' is the analytic center for the region and $s > 0$ is among the constraints defining the region, we have

$$s'' = O((K')^2 s') \tag{101}$$

and hence

$$\|x''\| = O((K')^2 \|x'\|). \tag{102}$$

Since $2z''$ does not lie in the region, we have

$$\|x''\| \geq \tfrac{1}{2} \ \text{ or } \ s'' \geq \tfrac{1}{2}. \tag{103}$$

Combining (101–103) gives

$$(K')^2\|x'\| = \Omega(1) \ \text{ or } \ (K')^2 s' = \Omega(1).$$

Since $K' = K + O(1)$ and $\|x'\| < 1$, the bound (97) follows. □

6.6. Proof of Lemma 6.5. Lemma 6.4 (applied with $\bar{w} = w$) implies that if $\|\hat{w} - w\|'_w = O(1)$, then

$$\|\tfrac{1}{\hat{s}}\hat{x}\| \leq \|\tfrac{1}{s}x\| + O(\tfrac{1}{s}).$$

Since $\frac{1}{s} = O(\|\frac{1}{s}x\| + 1)$ (by assumption), the bound (94) follows.

Lemma 6.4 also implies that if $\|\hat{w} - w\|'_w = O(1)$, then

$$\|\tfrac{1}{s}x\| \leq \|\tfrac{1}{\hat{s}}\hat{x}\| + O(\tfrac{1}{s}\|\hat{w} - w\|'_w) \tag{104}$$

and

$$\tfrac{1}{\hat{s}} = O(\tfrac{1}{s}). \tag{105}$$

If $\|\hat{w} - w\|'_w = O(1)$ and $\frac{1}{s} = O(\|\frac{1}{s}x\| + 1)$, then (104) implies

$$\|\tfrac{1}{s}x\| = O(\|\tfrac{1}{\hat{s}}\hat{x}\| + 1)$$

and hence (again using $\frac{1}{s} = O(\|\frac{1}{s}x\| + 1)$),

$$\tfrac{1}{s} = O(\|\tfrac{1}{\hat{s}}x\| + 1). \tag{106}$$

Together, (105) and (106) yield (95). □

7. Proof of Theorem 1.9.

7.1. In proving Theorem 1.9 it is useful to rely on the functional

$$f''_{x,s,t} := \phi_x - 12\left[\ln(1 - \|x\|^2) + \ln(s) + \ln(1 - s) + \ln(1 + t) + \ln(2 - t)\right]$$

whose domain is

$$D_{f''} := \{(x, s, t);\ x > 0,\ \ \|x\| < 1,\ \ 0 < s < 1,\ \text{ and }\ -1 < t < 2\}.$$

Observe that the only difference between f'' and f is that the summand $\ln(1 + t)$ in the definition of f'' is replaced by $\ln t$ in the definition of f. Similarly, the only difference between $D_{f''}$ and D_f is that the constraint $-1 < t$ in the definition of $D_{f''}$ is replaced by $0 < t$ in the definition of D_f; thus $D_f \subseteq D_{f''}$. Also note that

$$(x, s, t) \in D_{f''} \ \Rightarrow\ (x, s) \in D_{f'}.$$

The functional f'' is an element of $\mathcal{F}(K'')$ where $K'' = \sqrt{K^2 + 72}$; recall that in the terminology of Nesterov and Nemirovskii [3], $f'' \in \mathcal{F}(K'')$ essentially means that f'' is a nondegenerate self-concordant barrier functional with barrier parameter of magnitude $O((K'')^2)$.

In the same manner that the Hessian H_w of f at $w \in D_f$ induces the inner product $\langle\ ,\ \rangle_w$ and norm $\|\ \|_w$, the Hessian H''_w of f'' at $w \in D_{f''}$ induces an inner product $\langle\ ,\ \rangle''_w$ and norm $\|\ \|''_w$.

Let $\text{val}''_{\text{inf}}$ denote the optimal value of the problem (in the variable $w = (x, s, t)$),

$$\begin{array}{ll} \inf & t \\ \text{s.t.} & \bar{A}w = 0, \\ & w \in D_{f''} \end{array} \tag{107}$$

and let $\text{val}''_{\text{sup}}$ denote the optimal value of the problem obtained by replacing "inf" in (107) with "sup."

Our proof of Theorem 1.9 depends on the corollary to the following two propositions.

PROPOSITION 7.1. *If* $\text{relcond}(A, b) < \infty$, *then*

$$\text{val}''_{\text{inf}} \le \frac{-1}{2\,\text{relcond}(A, b)}$$

and

$$\text{val}''_{\text{sup}} \ge 1.$$

Proof. The bound on $\text{val}''_{\text{inf}}$ is a special case of [6, Prop. 3.2]. The bound on $\text{val}''_{\text{sup}}$ follows from the fact that $(\check{x}, \frac{1}{2}, 1)$ satisfies the constraints of (107). □

For a constant $\mu \in \mathbb{R}$, let z''_μ denote the optimal solution of the problem

$$\begin{array}{ll} \inf & \mu t + f''_w \\ \text{s.t.} & \bar{A}w = 0, \\ & w \in D_{f''}. \end{array} \tag{108}$$

(A unique optimal solution exists because $f'' \in \mathcal{F}(K'')$, $D_{f''}$ is bounded, and the feasible region for (108) is nonempty (in particular, $(\check{x}, \frac{1}{2}, 1)$ is feasible); cf. [5].)

Define

$$\text{CentTraj}''_{+,-} := \{z''_\mu;\ \mu \in \mathbb{R}\}.$$

The subscripts $+, -$ are meant to hightlight the fact that all values of μ are allowed, not just nonnegative values as in the definition of CentTraj in §1.

PROPOSITION 7.2. *If*

$$z''_{\mu_1} = (x''_{\mu_1}, s''_{\mu_1}, t''_{\mu_1}) \in \text{CentTraj}''_{+,-} \quad \textit{and} \quad z''_{\mu_2} = (x''_{\mu_2}, s''_{\mu_2}, t''_{\mu_2}) \in \text{CentTraj}''_{+,-}$$

satisfy

$$|t''_{\mu_2} - t''_{\mu_1}| \leq \frac{O(\min\{\text{val}''_{\text{sup}} - t_{\mu_2},\ t_{\mu_2} - \text{val}''_{\text{inf}}\})}{(K'')^2},$$

then

$$\|z''_{\mu_2} - z''_{\mu_1}\|''_{z''_{\mu_1}} \leq \frac{O((K'')^2 |t''_{\mu_2} - t''_{\mu_1}|)}{\min\{\text{val}''_{\text{sup}} - t_{\mu_2},\ t_{\mu_2} - \text{val}''_{\text{inf}}\}}.$$

Proof. This proposition follows from standard (but somewhat involved) arguments in the ipm literature; it is solely a consequence of f'' being an element of $\mathcal{F}(K'')$; for a detailed proof, cf. [5]. □

Recall that z' denotes the analytic center for the constraints

$$\begin{array}{l} \hat{A}(x, s) = 0, \\ (x, s) \in D_{f'}. \end{array}$$

COROLLARY 7.3. *Assume* $\text{relcond}(A, b) < \infty$. *If* $z''_\mu = (x''_\mu, s''_\mu, t''_\mu) \in \text{CentTraj}''_{+,-}$ *satisfies*

$$|t''_\mu| \leq \frac{O(1)}{(K^2 + 1)\,\text{relcond}(A, b)},$$

then

$$\|(z', 0) - z''_\mu\|''_{z''_\mu} \leq O(|t''_\mu|)(K^2 + 1)\,\text{relcond}(A, b).$$

Proof. This result follows easily from Propositions 7.1 and 7.2 because

$$(z', 0) \in \text{CentTraj}''_{+,-}$$

and $K'' = K + O(1)$. □

7.2. Proof of Theorem 1.9. Observe that if $w = (x, s, t) \in D_f$, then for all $(\bar{x}, \bar{s}, \bar{t})$,

$$\langle (\bar{x}, \bar{s}, \bar{t}), H_w(\bar{x}, \bar{s}, \bar{t}) \rangle = \langle (\bar{x}, \bar{s}, 0), H_w(\bar{x}, \bar{s}, 0) \rangle + \langle (0, 0, \bar{t}), H_w(0, 0, \bar{t}) \rangle$$

and

$$\langle (\bar{x}, \bar{s}, \bar{t}), H''_w(\bar{x}, \bar{s}, \bar{t}) \rangle = \langle (\bar{x}, \bar{s}, 0), H''_w(\bar{x}, \bar{s}, 0) \rangle + \langle (0, 0, \bar{t}), H''_w(0, 0, \bar{t}) \rangle.$$

Moreover,

$$\langle (\bar{x}, \bar{s}, 0), H_w(\bar{x}, \bar{s}, 0) \rangle = \langle (\bar{x}, \bar{s}), H'_{x,s}(\bar{x}, \bar{s}) \rangle$$

and

$$\langle (\bar{x}, \bar{s}, 0), H''_w(\bar{x}, \bar{s}, 0) \rangle = \langle (\bar{x}, \bar{s}), H'_{x,s}(\bar{x}, \bar{s}) \rangle.$$

Hence,

$$\|(\bar{x}, \bar{s}, \bar{t})\|_w^2 = (\|(\bar{x}, \bar{s})\|'_{x,s})^2 + \|(0, 0, \bar{t})\|_w^2 \tag{109}$$

and

$$(\|(\bar{x}, \bar{s}, \bar{t})\|''_w)^2 = (\|(\bar{x}, \bar{s})\|'_{x,s})^2 + (\|(0, 0, \bar{t})\|''_w)^2. \tag{110}$$

If $w = (x, s, t) \in D_f$ and $\hat{w} = (\hat{x}, \hat{s}, \hat{t})$, then (109) implies

$$\|(\hat{x}, \hat{s}) - (x, s)\|'_{x,s} \leq \|\hat{w} - w\|_w. \tag{111}$$

Consequently,

$$\|\hat{w} - w\|_w = O(1) \;\Rightarrow\; \left[\|v\|'_{\hat{x},\hat{s}} = \Theta(\|v\|'_{x,s}) \text{ for all } v\right]. \tag{112}$$

Note that if $w = (x, s, t) \in D_f$ and $z_\mu = (x_\mu, s_\mu, t_\mu) \in \text{CentTraj}$, then (111) and (112) yield

$$\|z_\mu - w\|_w = O(1) \;\Rightarrow\; \|z' - (x, s)\|'_{x,s} = O(\|z_\mu - w\|_w + \|z' - (x_\mu, s_\mu)\|'_{x_\mu,s_\mu}).$$

Consequently, to prove the theorem it is sufficient to show that if

$$\|z_\mu - w\|_w = O(1) \text{ and } t \leq \frac{O(1)}{(K^2 + 1)\,\text{relcond}(A, b)}, \tag{113}$$

then

$$\|z' - (x_\mu, s_\mu)\|'_{x_\mu,s_\mu} \leq O(t)(K^2 + 1)\,\text{relcond}(A, b). \tag{114}$$

Since

$$\|v\|_w = O(1) \;\Rightarrow\; w + v \in D_f$$

and since

$$(\bar{x}, \bar{s}, \bar{t}) \in D_f \;\Rightarrow\; \bar{t} > 0,$$

we have

$$\|z_\mu - w\|_w = O(1) \quad \Rightarrow \quad t_\mu = O(t).$$

Consequently, to prove that (113) implies (114), it is sufficient to show that if

$$t_\mu \leq \frac{O(1)}{(K^2+1)\,\mathrm{relcond}(A,b)}, \tag{115}$$

then

$$\|z' - (x_\mu, s_\mu)\|'_{x_\mu, s_\mu} \leq O(t_\mu)(K^2+1)\,\mathrm{relcond}(A,b). \tag{116}$$

Since (110) implies

$$\|z' - (x_\mu, s_\mu)\|'_{x_\mu, s_\mu} \leq \|(z', 0) - z_\mu\|''_{z_\mu},$$

to prove that (115) implies (116) it is sufficient to show that (115) implies

$$\|(z', 0) - z_\mu\|''_{z_\mu} = O(t_\mu)(K^2+1)\,\mathrm{relcond}(A,b). \tag{117}$$

However, since CentTraj $\subseteq$ CentTraj$''_{+,-}$, the fact that (115) does indeed imply (117) follows immediately from Corollary 7.3. □

REFERENCES

[1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.

[2] R. J. DUFFIN, *Infinite programs*, in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 157–170.

[3] YU. E. NESTEROV AND A. S. NEMIROVSKII, *Interior Point Methods in Convex Optimization: Theory and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1993.

[4] J. RENEGAR, *Some perturbation theory for linear programming*, Math. Programming, 65 (1994), pp. 73–92.

[5] ———, *Lecture Notes on the Efficiency of Newton's Method for Convex Optimization in Hilbert Spaces*, preprint, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1993.

[6] ———, *Linear programming, complexity theory and elementary functional analysis*, Math. Programming, 70 (1995), pp. 279–351.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 913–932, November 1996
© 1996 Society for Industrial and Applied Mathematics

AN INTERIOR POINT METHOD FOR BORDERED BLOCK-DIAGONAL LINEAR PROGRAMS*

MICHAEL D. GRIGORIADIS† AND LEONID G. KHACHIYAN†

Abstract. This paper presents an interior point method for solving a *bordered block-diagonal* linear program which consists of a number of disjoint blocks coupled by a total of p variables and constraints. This structure includes the well-known block-angular and dual block-angular structures, as well as their special cases, such as staircase problems, generalized bounds, and multicommodity flows. When p is small relative to the total dimension of the problem, the method achieves a substantial speedup relative to other general-purpose methods.

Key words. structured linear programs, bordered block-diagonal problems, interior point methods

AMS subject classifications. 68Q25, 90C05, 90C30

1. Introduction.

We consider the following *bordered* (or L-*shaped*) *block-diagonal* linear programming problem:

$$\text{minimize} \quad a_0^T x_0 + \sum_{k=1}^{K} a_k^T x_k \tag{1.1}$$

$$\text{subject to} \quad A_0 x_0 + \sum_{k=1}^{K} A_k x_k = b_0, \tag{1.2}$$

$$C_k x_0 + B_k x_k = b_k, \qquad k = 1, \ldots, K, \tag{1.3}$$

$$x = (x_0, x_1, \ldots, x_K) \geq 0,$$

where A_k, B_k, and C_k are arbitrary real (M, n_k)-, (m_k, n_k)-, and (m_k, N)-matrices, respectively; A_0 is an (M, N)-matrix; and $a_0 \in \mathbb{R}^N$, $b_0 \in \mathbb{R}^M$, $a_k \in \mathbb{R}^{n_k}$, and $b_k \in \mathbb{R}^{m_k}$ are given vectors for each k. Equations (1.2) are called *coupling constraints*, variables $x_0 \in \mathbb{R}^N$ are called *coupling variables*, and $x_k \in \mathbb{R}^{n_k}$ are called *block variables* for each $k = 1, \ldots, K$. The entire constraint matrix consists of $m = M + \sum_{k=1}^{K} m_k$ rows and $n = N + \sum_{k=1}^{K} n_k$ columns, $p = N + M$ columns and rows of which "couple" the otherwise disjoint block problems $\min \{ a_k^T x_k \mid B_k x_k = b_k,\ x_k \geq 0 \}$, $k = 1, \ldots, K$.

The above general bordered block-diagonal structure includes the well-known *block-angular* ($N = 0$) and *dual block-angular* ($M = 0$) structures, as well as their special cases, such as multistage or staircase problems, generalized bounds, and multicommodity flows, recently studied in the context of interior point methods [3, 4, 11, 12, 13, 26, 25, 29, 28]. But unlike these special cases, the dual of the bordered block-diagonal form (1.1)–(1.3) exhibits the same structure as the primal. This property is important for solving (1.1)–(1.3) by primal-dual interior point methods since the optimization variant of the problem can be reduced by duality to a similarly structured

*Received by the editors February 23, 1994; accepted for publication (in revised form) June 7, 1995. This research was supported by National Science Foundation grant CCR-9208539.

†Department of Computer Science, Hill Center, Rutgers University, New Brunswick, NJ 08903 (grigoriadis@cs.rutgers.edu, leonid@cs.rutgers.edu).

feasibility problem with a total of $O(p)$ coupling rows and columns and $O(K)$ blocks, each of dimension at most $(n_k, n_k + m_k)$. In view of this observation, we shall focus on the feasibility variant of the bordered block-diagonal optimization problem: compute a nonnegative vector $x = (x_0, x_1, \ldots, x_K)$ satisfying (1.2)–(1.3).

We shall first state an $O(nL)$-iteration, structure-preserving extension of Karmarkar's projective interior point method for linear programming to this overall *feasibility* problem. For the case of dense A_k, B_k, and C_k of (1.2)–(1.3), each iteration of this method requires

$$O\left(p^3 + p^2 \sum_{k=1}^{K} (n_k + m_k) + p \sum_{k=1}^{K} n_k m_k + \sum_{k=1}^{K} n_k m_k^2\right) \tag{1.4}$$

arithmetic operations. When p is small compared to the size of the overall problem, a substantial speedup is achieved relative to other nonstructured interior point methods, even if standard heuristics to minimize the fill-in in the Cholesky factors of the scaled least-squares matrix of the entire system (1.2)–(1.3) are used. Our approach is similar to that presented in [12] for multicommodity flows—the feasibility problem is treated as a convex quadratic programming problem which minimizes the l_2-norm of the M-dimensional error vector of the coupling constraints (1.2) and maintains the feasibility of the constraints (1.3). As in [12, 32, 15, 16], this quadratic problem is then reduced to a sequence of local minimizations of the scaled error vector over an ellipsoid, circumscribed about the current approximation. However, by considering the dual of the error-minimization problem and by adjusting the circumscribed ellipsoid, each step of the algorithm can be modified so that it becomes equivalent to solving a least-squares system of M linear equations. This avoids the solution of the $O(\log n)$ or $O(L)$ linear least-squares systems within the binary search used in each iteration of the algorithms of [12] and [32], respectively. The resulting algorithm is also capable of computing an exact solution of (1.2)–(1.3), interior to the block polyhedron (1.3), in a finite number of steps.

The structured projective algorithm and its modification which avoids the binary search mentioned above are described in §2. Section 4 presents preliminary computational results with an affine variant of the method which is outlined in §3. Finally, §5 shows that by using compact inverse techniques, the complexity of the bordered block-diagonal problem (1.2)–(1.3) can be reduced to $O(\sqrt{n}L)$ iterations, each of which still requires a number of arithmetic operations bounded by (1.4). This implies the bound of $O\left(\sqrt{n+m}L\right)$-iterations for the bordered block-diagonal *optimization* problem (1.1)–(1.3) and its dual, where each iteration requires

$$O\left(p^3 + p^2 \sum_{k=1}^{K} (n_k + m_k) + p \sum_{k=1}^{K} n_k m_k + \sum_{k=1}^{K} n_k^2 m_k\right)$$

arithmetic operations for dense matrices A_k, B_k, and C_k of (1.2)–(1.3).

We shall use the following notation: lower case letters for vectors and capitals for matrices, the subscript k for the block index, e for vectors of all ones, I_q for the q-order unit matrix, $X = \operatorname{diag}(x)$ for the diagonal matrix whose diagonal is x, $\Pi(x)$ for the product of the components of x, and " ^{T} " for the transpose.

2. Projective interior point method for bordered block-diagonal problems. We start with the following *homogeneous* bordered block-diagonal (strict) fea-

sibility problem:[1] compute a nontrivial solution $x^* = (x_0^*, x_1^*, \ldots, x_K^*)$ of the following system:

$\mathbf{P_H}$:

$$A_0 x_0 + \sum_{k=1}^{K} A_k x_k = 0, \tag{2.1}$$

$$C_k x_0 + B_k x_k = 0, \qquad k = 1, \ldots, K, \tag{2.2}$$

$$x_0, x_1, \ldots, x_K \geq 0.$$

We shall assume that we are given a *strictly block-feasible solution*[1] of $\mathbf{P_H}$, that is, an $x = (x_0, x_1, \ldots, x_K) > 0$ that satisfies the constraints (2.2) of $\mathbf{P_H}$. Karmarkar's canonical form for linear programs [14] is a special case of $\mathbf{P_H}$, e.g., for $N = 0$ and $K = M = 1$. We shall also assume without loss of generality that no system (2.2) contains linearly-dependent equations. In fact, we can and shall make the stronger assumption that each matrix B_k in (2.2) is of full row rank.

Remark 1. If some B_k is rank deficient, one or more of its rows can be zeroed out by pivoting on (C_k, B_k). Collecting all such rows for all k results in a system of linear equations, say, $C_{K+1} x_0 = 0$. After discarding any linearly-dependent rows, either $x_0 = 0$ or some components of x_0 can be eliminated, and the dimension of x_0 reduced, at the expense of introducing a new block $B_{K+1} = I$. □

We shall work with the standard logarithmic *potential function* of the following form:

$$\mathcal{K}(x_0, x_1, \ldots, x_K) = n \ln \mathcal{E}(x_0, x_1, \ldots, x_K) - \sum_{k=0}^{K} \ln \Pi(x_k), \tag{2.3}$$

where $\Pi(x_k)$ denotes the product of the n_k elements of x_k and $\mathcal{E}(x_0, x_1, \ldots, x_K) = \|r(x)\|$ denotes the magnitude of the error (infeasibilities)

$$r(x) = r(x_0, x_1, \ldots, x_K) = \sum_{k=0}^{K} A_k x_k$$

in the coupling constraints (2.1) for the ℓ_2-norm.

2.1. The error-reduction subproblem. Let $\hat{x} = (\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_K) > 0$ be the current strictly block-feasible iterate. The standard operation is to introduce the *scaled block variables* $\xi_k \in \mathbb{R}^{n_k}$ defined from the scaling (change of variables):

$$x_k = \hat{X}_k(e + \xi_k), \qquad k = 0, 1, \ldots, K.$$

The potential (2.3) can then be written in terms of the ξ_k as follows:

$$\mathcal{K}(\xi_0, \xi_1, \ldots, \xi_K) = n \ln \mathcal{E}(\xi_0, \xi_1, \ldots, \xi_K) - \sum_{k=0}^{K} \ln \Pi(e + \xi_k) + \mathcal{C}, \tag{2.4}$$

[1]The nonhomogeneous problem (1.2)–(1.3) can be reduced to the homogeneous case $\mathbf{P_H}$ by the same transformations used for Karmarkar's method. Similarly, the requirement that a strictly block-feasible solution of $\mathbf{P_H}$ be given can be removed by appending an artificial column and row, which preserves the bordered block-diagonal structure.

where $\mathcal{E}(\xi_0, \xi_1, \ldots, \xi_K) = \| r + \sum_{k=0}^{K} A_k \hat{X}_k \xi_k \|$, the current error vector $r = \sum_{k=0}^{K} A_k \hat{x}_k$, and $C = \sum_{k=0}^{K} \ln \Pi(\hat{x}_k)$ is a constant of no consequence. As in [12, 32], consider the local *error-reduction subproblem*:

$$\mathbf{R}: \qquad \min \{ \mathcal{E}(\xi_0, \xi_1, \ldots, \xi_K) \mid (\xi_0, \xi_1, \ldots, \xi_K) \in L \cap \mathcal{B}_\alpha \},$$

where

$$L = L(\hat{x}) = \left\{ \xi_0, \xi_1, \ldots, \xi_K \;\middle|\; \begin{array}{c} C_k \hat{X}_0 \xi_0 + B_k \hat{X}_k \xi_k = 0, \quad k = 1, \ldots, K, \\ \sum_{k=0}^{K} e^T \xi_k = 0 \end{array} \right\} \tag{2.5}$$

and

$$\mathcal{B}_\alpha = \{ \xi_0, \xi_1, \ldots, \xi_K \mid \|\xi_0, \xi_1, \ldots, \xi_K\| \leq \alpha \}$$

for some positive constant α. Note that if $\xi \in L \cap \mathcal{B}_\alpha$ and $\alpha < r(n) \triangleq \sqrt{n/(n-1)}$, the radius of the inscribed sphere for the simplex $S_n = \{\xi_0, \xi_1, \ldots, \xi_K \mid \sum_{k=0}^{K} e^T \xi_k = 0, e + \xi_k \geq 0 \text{ for all } k = 1, \ldots, K\}$, then $x = (\hat{X}_0(e + \xi_0), \hat{X}_1(e + \xi_1), \ldots, \hat{X}_K(e + \xi_K))$ is strictly feasible.

THEOREM 1. *Suppose that the original problem* $\mathbf{P_H}$ *has a nontrivial nonnegative solution. Let* $\xi^* = (\xi_0^*, \xi_1^*, \ldots, \xi_K^*)$ *be an optimal solution of the error-reduction subproblem* $\mathbf{R}$ *defined for some* $\alpha \in [\frac{1}{2}\sqrt{n/(n-1)}, \sqrt{n(n-1)}]$. *Then, replacing* $\hat{x}$ *by* $\hat{X}(e + \beta^* \xi^*)$, *where* $\beta^* = \operatorname{argmin}\{\mathcal{K}(\hat{X}(e + \beta\xi^*)) \mid \beta \geq 0\}$, *decreases the potential function* $\mathcal{K}(\cdot)$ *by at least* $1 - \ln 2 \approx 0.31$.

Proof. We shall only sketch the proof, which is standard (see, e.g., [14], [24]). It is based on the observation that for a nontrivially feasible problem $\mathbf{P_H}$ there is a zero η of the function $\mathcal{E}(\xi)$ in the intersection of L and the $(n-1)$-dimensional simplex S_n so that $\eta \in L \cap \mathcal{B}_{R(n)}$, where $R(n) \triangleq \sqrt{n(n-1)}$ is the radius of the circumscribed sphere for the simplex. If $\mathbf{R}$ were solved for $\alpha = R(n)$, the minimum value of $\mathbf{R}$ would be zero, but such long steps are not possible. Nevertheless, the convexity of $\mathcal{E}(\cdot)$ and the fact that $\mathcal{E}(\xi^*) = 0$ for $\|\xi^*\| < \alpha$ imply that $\mathcal{E}(\beta\xi^*) \leq \mathcal{E}(0)\,(1 - \beta\|\xi^*\|/R(n))$ for all $\beta \in [0, 1]$. If $\|\xi^*\| \geq r(n)/2$, we set $\beta = r(n)/2\|\xi^*\|$ and obtain $\mathcal{E}(\beta\xi^*) \leq \mathcal{E}(0)(1 - 1/2(n-1))$. Since $\|\beta\xi^*\| = r(n)/2$, we have $\Pi(e + \beta\xi^*) \geq (1 + 1/2(n-1))^{n-1}/2$ (see [24, p. 192]), and the claim follows. If $\|\xi^*\| < r(n)/2$, we have $\mathcal{E}(\xi^*) = 0$, and the claim follows for $\beta = 1$. □

The above theorem immediately implies the bound of $O(nL)$-iterations for the projective iterative method $x \leftarrow \hat{X}(e + \beta^* \xi^*)$ for any value of α in the range $[\frac{1}{2}\sqrt{n/(n-1)}, \sqrt{n(n-1)}]$. These theoretical results notwithstanding, the number of iterations in practice is almost independent of the size of the problem, and the overall run time heavily depends on how efficiently problem $\mathbf{R}$ can be solved at each iteration.

2.2. The dual of the error-reduction subproblem. The preceding discussion applies to all problems and has little to do with the specific bordered block-diagonal structure of problem $\mathbf{P_H}$. In this section we shall state an algorithm that performs as much independent computation within the blocks of $\mathbf{P_H}$ as possible.

Our approach is based on solving the dual of the error-reduction subproblem $\mathbf{R}$. The relationship between $\mathbf{R}$ and its dual can be characterized by von Neumann's saddlepoint theorem (see, e.g., [23, p. 393]) as follows:

$$\min_{\xi \in L \cap \mathcal{B}_\alpha} \mathcal{E}(\xi_0, \xi_1, \ldots, \xi_K) = \min_{\xi \in L \cap \mathcal{B}_\alpha} \left\| r + \sum_{k=0}^{K} A_k \hat{X}_k \xi_k \right\|$$

$$
\begin{aligned}
&= \min_{\xi \in L \cap \mathcal{B}_\alpha} \max_{y \in U} \left\{ y^T r + y^T \sum_{k=0}^{K} A_k \hat{X}_k \xi_k \right\} \\
&= \max_{y \in U} \min_{\xi \in L \cap \mathcal{B}_\alpha} \left\{ y^T r + y^T \sum_{k=0}^{K} A_k \hat{X}_k \xi_k \right\} \\
&= \max_{y \in U} \mathcal{D}(y),
\end{aligned} \tag{2.6}
$$

where U denotes the Euclidean unit ball in $\mathbb{R}^M$. From the implied definition of $\mathcal{D}(y)$ above, it is convenient to use the notation

$$
\mathcal{D}(y) = r^T y + \min_{\xi \in L \cap \mathcal{B}_\alpha} \{c^T(y)\xi\}, \tag{2.7}
$$

where

$$
c(y) = \left(\hat{X}_0 A_0^T y, \hat{X}_1 A_1^T y, \ldots, \hat{X}_k A_k^T y \right). \tag{2.8}
$$

We see that the computation of $\mathcal{D}(y)$ for a fixed y amounts to projecting $c(y)$ onto the subspace L. This is because the minimizer of $\mathcal{D}(y)$ is given by

$$
\xi(y) = -\alpha P c(y) / \|Pc(y)\|,
$$

where P is the orthogonal projection matrix for L. Upon substitution of this $\xi(y)$ into (2.7), and since P is idempotent, we obtain

$$
\begin{aligned}
\mathcal{D}(y) &= r^T y - \alpha c^T(y) P c(y) / \|Pc(y)\| \\
&= r^T y - \alpha \sqrt{c^T(y) P c(y)} \\
&= r^T y - \alpha \sqrt{y^T G y},
\end{aligned}
$$

where G is an M-order symmetric positive semidefinite matrix. Using the Sherman–Morrison–Woodbury formula (see, e.g., [6]), one can verify that

$$
G = \sum_{k=1}^{K} G_k + S \Gamma S^T - \frac{1}{n} r r^T, \tag{2.9}
$$

where, for each $k = 1, \ldots, K$,

$$
G_k = A_k \hat{X}_k \hat{P}_k \hat{X}_k A_k^T \tag{2.10}
$$

is the Gram matrix of $A_k \hat{X}_k \hat{P}_k$ and

$$
\hat{P}_k = I_k - \hat{X}_k B_k^T (B_k \hat{X}_k^2 B_k^T)^{-1} B_k \hat{X}_k \tag{2.11}
$$

is the n_k-order projection matrix onto the kth block subspace $L_k = \{\xi_k \mid B_k \hat{X}_k \xi_k = 0\}$. The N-order positive definite matrix Γ is given by

$$
\Gamma = \left(I_N + \sum_{k=1}^{K} \hat{X}_0 C_k^T \left(B_k \hat{X}_k^2 B_k^T \right)^{-1} C_k \hat{X}_0 \right)^{-1}, \tag{2.12}
$$

and S is given by

$$S = A_0\hat{X}_0 - \sum_{k=1}^{K} A_k\hat{X}_k^2 B_k^T \left(B_k\hat{X}_k^2 B_k^T\right)^{-1} C_k\hat{X}_0. \tag{2.13}$$

Once G is constructed, the dual of $\mathbf{R}$ can be written from the above as follows:

$$\mathbf{D_y}: \qquad \mathcal{D}^* \equiv \mathcal{D}(y^*) = \max\left\{ r^T y - \alpha\sqrt{y^T G y} \;\middle|\; \|y\| \le 1 \right\}.$$

Suppose that $\mathbf{D_y}$ has a unique solution $y^* \in U$ and let $c^* = c(y^*) \in \mathbb{R}^n$ be given by (2.8). Then the error-reduction subproblem $\mathbf{R}$ also has the unique solution $\xi^* = -\alpha\eta^*/\|\eta^*\|$, where η^* is the projection of c^* onto the subspace L defined for $\mathbf{R}$. This projection can be computed by the formulae

$$\eta_0^* = \Gamma\left(c_0^* - \sum_{k=1}^{K} \hat{X}_0 C_k \left(B_k\hat{X}_k^2 B_k^T\right)^{-1} B_k\hat{X}_k c_k^*\right), \tag{2.14}$$

$$\eta_k^* = P_k c_k^* - \hat{X}_k B_k^T \left(B_k\hat{X}_k^2 B_k^T\right)^{-1} C_k\hat{X}_0\eta_0^*, \tag{2.15}$$

followed by the update operations

$$\eta_0^* \leftarrow \eta_0^* - e\ (e^T c_0^*/n), \quad \eta_k^* \leftarrow \eta_k^* - e\ (e^T c_k^*/n). \tag{2.16}$$

As we shall see in the next section, if the solution of the dual problem given by (2.8) is not unique, the value of α can be adjusted so that either α gets into the working range $\frac{1}{2}\sqrt{n/(n-1)} \le \alpha \le \sqrt{n(n-1)}$ of Theorem 1 and a solution of the new error-reduction subproblem $\mathbf{R}$ is obtained, or an exact solution of the original problem $\mathbf{P_H}$ is computed in one step of the algorithm.

2.3. A direct method for the dual. Several methods can be devised to solve $\mathbf{D_y}$ efficiently for a fixed positive α. We shall confine our discussion here to the simplest direct method based on orthogonal transformations of G. Specifically, we shall represent $G = QDQ^T$, where $QQ^T = I$ and $D = \mathrm{diag}(d_1, \ldots, d_M) \ge 0$, so that the corresponding change of variables $y = Qu$ transforms problem $\mathbf{D_y}$ to the diagonal form

$$\mathbf{D}: \qquad \mathcal{D}^* \equiv \mathcal{D}(u^*) = \max\left\{ g^T u - \alpha\sqrt{u^T D u} \;\middle|\; \|u\| \le 1 \right\},$$

where $g = Q^T r$. In the rest of this section we shall examine how to solve $\mathbf{D}$.

Let $M_0 = \{\, i \mid d_i = 0 \,\}$ and $M_+ = \{\, i \mid d_i > 0 \,\}$ be the partition of the coupling-row indices $i = 1, \ldots, M$ induced by the above transformation, and denote by u_0 and u_+ the corresponding components of u. Problem $\mathbf{D}$ is analogously partitioned into the two problems, in variables u_0 and u_+, conformal to M_0 and M_+, as follows:

$$\mathbf{D_0}: \quad \mathcal{D}_0^* = \mathcal{D}(u_0^*) = \max\left\{ \mathcal{D}(u_0) = g_0^T u_0 \;\middle|\; \|u_0\| \le 1 \right\},$$

$$\mathbf{D_+}: \quad \mathcal{D}_+^* = \mathcal{D}(u_+^*) = \max\left\{ \mathcal{D}(u_+) = g_+^T u_+ - \alpha\sqrt{u_+^T D_+ u_+} \;\middle|\; \|u_+\| \le 1 \right\}.$$

Clearly, $\mathcal{D}_0^* = \|g_0\|$, $\mathcal{D}^* = \sqrt{\|g_0\|^2 + \mathcal{D}_+^{*2}}$, and

$$u^* = (g_0,\ u_+^*\mathcal{D}_+^*)\,/\,\mathcal{D}^*$$

for positive $\mathcal{D}^*$. Moreover, the following lemma shows that $\|g_0\| > 0$ implies that the original problem $\mathbf{P_H}$ has no nontrivial nonnegative solution.

LEMMA 1. *Suppose that problem* $\mathbf{P_H}$ *has a nontrivial nonnegative solution. Then*

$$y^T G y = 0 \quad \Rightarrow \quad r^T y = 0 \quad \textit{for any} \quad y \in \mathbb{R}^M.$$

Proof. Without loss of generality assume that $K = 1$ and $N = 0$ and dispense with the block index k and the coupling variables x_0. Suppose that problem $\mathbf{P_H}$ has a nontrivial solution $\hat{x} \geq 0$. Then for any $y \in \mathbb{R}^n$ the system

$$\begin{aligned} y^T A \hat{X} \xi + y^T r &= 0, \\ B \hat{X} \xi &= 0, \\ e^T \xi &= 0 \end{aligned}$$

is feasible. Let $c(y) = \hat{X} A^T y$ as in (2.8) and rewrite the above system as follows (see (2.5)):

$$c^T(y)\xi + r^T y = 0, \quad \xi \in L = \left\{ \xi \;\middle|\; B\hat{X}\xi = 0, \quad e^T \xi = 0 \right\}.$$

Suppose that $r^T y \neq 0$. Then the linear form $c^T(y)\xi$ is not constant on L, i.e., $Pc(y) \neq 0$, where P is the projection matrix given by (2.11) for L. Thus, $y^T G y = \|Pc(y)\|^2 \neq 0$ for the matrix G defined by (2.10). □

COROLLARY 1. *If* $\mathbf{P_H}$ *has a nontrivial nonnegative solution, then* $\|g_0\| = 0$.

Proof. It follows directly from Lemma 1: $d_i = 0$ implies $g_i = 0$. □

Remark 2. If $d_i = g_i = 0$ for $i = 1, \ldots, s < M$, the number of coupling constraints can be reduced by s: replace the original coupling system $\sum_{k=0}^{K} A_k x_k = 0$ by the equivalent system $Q^T \sum_{k=0}^{K} A_k x_k = 0$. The first s equations of the new system are identically zero for any solution $x = (x_0, x_1, \ldots, x_K)$ satisfying the constraints (2.2) of $\mathbf{P_H}$. These rows of the so-modified system can be discarded and the remaining part used in place of the original coupling system (2.1). In many cases, this is a good way to reduce the number of coupling rows, but if the coupling matrix $A = (A_0, A_1, \ldots, A_K)$ is sparse or if it contains some useful structure, premultiplying A by Q^T will almost certainly disturb these desirable properties. An alternate approach for such cases might be to remove a set of s redundant equations from the original system. □

It remains to show how to solve problem $\mathbf{D}_+$ with all positive d_i's. To this end, define

$$\rho = \sqrt{g_+^T D_+^{-1} g_+} = \max\{ r^T y \mid y^T G y \leq 1 \}. \tag{2.17}$$

The following two-step procedure finds a unique solution of problem $\mathbf{D}_+$ for the case $\rho > \alpha$. Its correctness and the uniqueness of the optimal solution it produces are shown in Lemma 2.

PROCEDURE $\mathcal{U}^*$.

Step i. Compute the unique positive zero t^* of the equation

$$f(t) \triangleq \sqrt{r^T (tI + G)^{-1} G (tI + G)^{-1} r} \equiv \sqrt{\sum_{i \in M_+} \left(\frac{g_i}{t + d_i} \right)^2 d_i} = \alpha. \tag{2.18.a}$$

Step ii. Compute u_+^* as follows:

$$u_+^* = u_+(t^*) = (t^* I + D_+)^{-1} g_+ \ / \ \|(t^* I + D_+)^{-1} g_+\| \ . \qquad \square \tag{2.18.b}$$

LEMMA 2. *The following three conditions are equivalent.*

1. $\rho > \alpha$.
2. *Problem* $\mathbf{D}_+$ *has the unique optimal solution* u_+^* *given by Procedure* $\mathcal{U}^*$.
3. $\mathcal{D}_+^* > 0$.

Proof. First, Part 1 implies that $f(0) = \rho > \alpha$. At the same time, $f(t_{\max}) < \alpha$ for

$$t_{\max} = \alpha^{-1}\sqrt{r^T G r} = \alpha^{-1}\sqrt{g_+^T D_+ g_+} \ . \tag{2.19}$$

Since the continuous function $f(t)$ is strictly decreasing, it has a unique positive zero $t^* \in (0, t_{\max})$, and hence Step (i) of Procedure $\mathcal{U}^*$ is well defined, as is the u_+^* computed by (2.18.b). Second, note that (2.18.a–b) imply that

$$\begin{aligned} \nabla \mathcal{D}(u_+)\big|_{u_+ = u_+^*} &= g_+ - \alpha D_+ u_+^* \Big/ \sqrt{u_+^{*T} D_+ u_+^*} \\ &= (I - D_+(t^* I + D_+)^{-1})\, g_+ \\ &= t^* \left\| (t^* I + D_+)^{-1} \, g_+ \right\| u_+^* . \end{aligned} \tag{2.20.a}$$

Third, by Step (ii) of Procedure $\mathcal{U}^*$,

$$\| u_+^* \| = 1. \tag{2.20.b}$$

We recognize equations (2.20.a–b) as the Karush–Kuhn–Tucker optimality conditions for the concave maximization problem $\mathbf{D}_+$, which confirms the optimality of u_+^*. Its uniqueness follows from the strict convexity of the Euclidean unit ball $\{ u_+ \mid \|u_+\| \le 1 \}$ and the positivity of the Lagrange multiplier

$$\lambda^* = t^* \left\| (t^* I + D_+)^{-1} \, g_+ \right\| \tag{2.21}$$

in (2.20.a). Hence, Part 1 implies Part 2 of the lemma.

Part 2 implies $\mathcal{D}_+^* > 0$ for $\mathbf{D}_+$: $\mathcal{D}(u_+)$ is homogeneous of degree one and therefore

$$\mathcal{D}_+^* = \mathcal{D}(u_+^*) = \nabla \mathcal{D}(u_+^*)^T u_+^* = \lambda^* > 0. \tag{2.22}$$

Finally, suppose that Part 1 is false for $\mathcal{D}_+^* > 0$; that is, $\rho \le \alpha$. Then, by the Cauchy–Schwarz inequality,

$$\begin{aligned} g_+^T u_+ &= \left(g_+ D_+^{-1/2} \right)^T \left(D_+^{1/2} u_+ \right) \\ &\le \sqrt{g_+^T D_+^{-1} g_+} \sqrt{u_+^T D_+ u_+} \\ &= \rho \sqrt{u_+^T D_+ u_+} \\ &\le \alpha \sqrt{u_+^T D_+ u_+} \ . \end{aligned}$$

That is, the homogeneous function $\mathcal{D}(u_+)$ is always nonpositive and hence $\mathcal{D}_+^* = 0$. $\square$

2.4. Computing an exact solution of $\mathbf{P_H}$. An attractive feature of the above projective method for the case $\rho \le r(n) = \sqrt{n/(n-1)}$ is its ability to compute an exact solution of the original problem $\mathbf{P_H}$ in one iteration by adjusting the final step length. We explore this property in the following lemma.

LEMMA 3. *If $\rho \le \sqrt{n/(n-1)}$, then the interior point algorithm obtains an exact nontrivial nonnegative solution of the original problem $\mathbf{P_H}$ in one iteration by applying Procedure $\mathcal{U}^*$ with $\alpha \leftarrow \rho$.*

Proof. If $\rho = 0$, we have a nontrivial nonnegative solution of $\mathbf{P_H}$ since $g = Qr$ implies that the error vector $r = 0$. Otherwise, consider the error-reduction subproblem $\mathbf{R}$ for $\alpha(\delta) = \rho - \delta$, where δ is a small positive number. For this new α we have the condition given in Part 1 of Lemma 2, and thus there is a unique minimizer $u^*(\delta)$ and $\xi^*(\delta)$ by Procedure $\mathcal{U}^*$ and Lemma 2. Since $\mathcal{D}_+ > 0$, from equation (2.18.a) and the definition of ρ in (2.17) it follows that $t^*(\delta)$ tends to zero as $\delta \to 0$. Now, (2.21) and (2.22) imply that $\mathcal{D}_+^*(\delta) = t^*(\delta) \,\|\, (t^*(\delta)I + D_+)^{-1} \, g_+ \,\| \to 0$. Accordingly, when Procedure $\mathcal{U}^*$ is executed with $\alpha = \rho$ and $t = 0$, it yields a nontrivial nonnegative solution of the original problem $\mathbf{P_H}$. That is, the algorithm terminates in one step. □

For $g_0 = 0$ and for the linear subspace L defined by (2.5), the above proof shows that

$$\rho = \min \left\{ \; \|\xi\| \;\; \Big| \;\; r + A\hat{X}\xi = 0, \;\; \xi \in L \; \right\}.$$

Hence, we must have $\rho < R(n) = \sqrt{n(n-1)}$ for a nontrivially feasible problem $\mathbf{P_H}$. On the other hand, if $\mathbf{P_H}$ has no nontrivial nonnegative solution, $\rho \to +\infty$ with the number of iterations; otherwise, the potential $\mathcal{K}(x)$ would decrease by a constant factor at each iteration, and since by the geometric–arithmetic mean inequality

$$e^{\mathcal{K}(x)/n} = \mathcal{E}(x) \Big/ \sqrt[n]{\Pi(x)} \ge \mathcal{E}(x) \sqrt{n} / \|x\|,$$

this would imply that $\mathbf{P_H}$ is nontrivially solvable.

Also consider that if $g_i \neq 0$ for some $d_i = 0$, then $\max\{ r^T y \mid y^T G y \le 1 \}$, the right term in the definition (2.17) of ρ, is $+\infty$. This implies that the feasibility test

$$\rho = \max\{ r^T y \mid y^T G y \le 1 \} \; < \; \sqrt{n(n-1)} \tag{2.23}$$

includes Corollary 1.

2.5. The algorithm. In this section we summarize the above results in a general outline of an algorithm for computing a nontrivial solution of the homogeneous problem $\mathbf{P_H}$. The algorithm is initialized with a strictly block-feasible solution $\hat{x}$, along with the desired feasibility tolerance $\varepsilon_{\text{feas}}$.

ALGORITHM A.

while $\|r\| = \left\| \sum_{k=0}^{K} A_k \hat{x}_k \right\| > \varepsilon_{\text{feas}}$ **do**
  Compute G from (2.9)–(2.13).
  Solve the system $G\psi(0) = r$, e.g., $\psi(0) = G^{\dagger} r$, where $G^{\dagger}$ is the pseudoinverse of G.
  Define $\rho = \sqrt{r^T \psi(0)}$ $\left(= \max\{r^T y \mid y^T G y \le 1\}.\right)$
  if $\rho > \sqrt{n(n-1)}$, **then** $\mathbf{P_H}$ has no nontrivial solution, **stop**
    elseif $\rho < \sqrt{n/(n-1)}$, **then** $y^* := \psi(0)/\|\psi(0)\|$,
    else $\left(\sqrt{n/(n-1)} \le \rho \le \sqrt{n(n-1)} \right)$ select any t such that

8 $\sqrt{\psi(t)^T G\psi(t)} \geq \frac{1}{2}\sqrt{n/(n-1)}$, where $\psi(t) = (tI_M + G)^{-1}r$.
9 $y^* := \psi(t)/\|\psi(t)\|$.
10 **endif**
11 Compute $c^* = c(y^*)$ from (2.8).
12 Use (2.14)–(2.16) to compute η^*, the projection of c^* onto L.
13 $\xi^* := -\eta^*/\|\eta^*\|$.
14 **if** $\rho < \sqrt{n/(n-1)}$ **then** $\beta^* := \rho$,
15 **else** $\beta^* := \operatorname{argmin}\left\{ \mathcal{K}\left(\hat{X}(e+\beta\xi^*)\right) \;\middle|\; 0 \leq \beta \leq \sqrt{n(n-1)} \right\}$ **endif**.
16 $\hat{x}_k := \hat{X}_k(e+\beta^*\xi_k^*)$ for $k = 0, 1, \ldots, K$.
17 **endwhile**.

It is always possible to set $t = 0$ and $y^* = \psi(0)/\|\psi(0)\|$ in lines 7–9 of Algorithm A. However, a natural extension is to look for other, possibly more powerful, values of t. For example, several values of t, as well as of α (see Procedure $\mathcal{U}^*$), can be tried and a line search independently performed for each. There are opportunities for various selection strategies, including the selection of the best one among them.

3. An affine variant. The affine variant of the structured projective Algorithm A works directly with the set of bordered block-diagonal constraints

$$\textbf{PA:} \qquad \begin{aligned} A_0x_0 + \sum_{k=1}^{K} A_kx_k &= b_0, \\ C_kx_0 + \quad B_kx_k &= b_k, \qquad k = 1, \ldots, K, \\ x_0, x_1, \ldots, x_K &\geq 0 \end{aligned}$$

to produce a nontrivial solution x^*. As before, we shall assume that we can compute a strictly feasible initial solution $\hat{x} = (\hat{x}_0, \hat{x}_k) > 0$ satisfying the block constraints $C_k x_0 + B_k x_k = b_k$. Introducing the scaled variables $x_k = \hat{X}_k(e+\xi_k)$ for $k = 0, 1, \ldots, K$, we can rewrite the given affine constraints **PA** as follows:

$$\begin{aligned} A_0\hat{X}_0\xi_0 + \sum_{k=1}^{K} A_k\hat{X}_k\xi_k &= r_A, \\ C_k\hat{X}_0\xi_0 + \quad B_k\hat{X}_k\xi_k &= 0, \qquad k = 1, \ldots, K, \end{aligned}$$

where

$$r_A = \sum_{k=0}^{K} A_k\hat{x}_k - b_0. \tag{3.1}$$

Next we consider the *affine error-reduction subproblem*

$$\textbf{RA:} \qquad \min\{\, \mathcal{E}_A(\xi_0, \xi_1, \ldots, \xi_K) \mid (\xi_0, \xi_1, \ldots, \xi_K) \in L_A \cap \mathcal{B}_\alpha \,\},$$

where $\mathcal{B}_\alpha = \{\, (\xi_0, \xi_1, \ldots, \xi_K) \mid \|\xi_0, \xi_1, \ldots, \xi_K\| \leq \alpha \,\}$ is the Euclidean ball of radius α and

$$L_A = L_A(\hat{x}) = \left\{\, \xi_0, \xi_1, \ldots, \xi_K \mid C_k\hat{X}_0\xi_0 + B_k\hat{X}_k\xi_k = 0, \quad k = 1, \ldots, K \,\right\}. \tag{3.2}$$

Observe that for $N = 0$ the subspace L_A is the intersection of the K orthogonal "block" subspaces $L_k = \{\xi_k \mid B_k\hat{X}_k\xi_k = 0\}$, $k = 1, \ldots, K$.

The duality relations (2.6) introduced earlier hold for the affine case as well:

$$\begin{aligned}\min_{\xi\in L_A\cap\mathcal{B}_\alpha}\mathcal{E}_A(\xi_0,\xi_1,\ldots,\xi_K) &= \min_{\xi\in L_A\cap\mathcal{B}_\alpha}\left\|r_A+\sum_{k=0}^{K}A_k\hat{X}_k\xi_k\right\| \\ &= \min_{\xi\in L_A\cap\mathcal{B}_\alpha}\max_{y\in U}\left\{y^Tr_A+y^T\sum_{k=0}^{K}A_k\hat{X}_k\xi_k\right\} \\ &= \max_{y\in U}\min_{\xi\in L_A\cap\mathcal{B}_\alpha}\left\{y^Tr_A+y^T\sum_{k=0}^{K}A_k\hat{X}_k\xi_k\right\} = \max_{y\in U}\mathcal{D}_A(y).\end{aligned} \tag{3.3}$$

The latter minimum is attained at the point $\xi_A = -\alpha P_A c(y)/\|P_A c(y)\|$, where $c(y)$ is given by (2.8), and P_A is now the projection matrix for L_A. We thus arrive at the problem $\mathbf{DA_y}$, analogous to $\mathbf{D_y}$ for the homogeneous case, as follows:

$$\mathbf{DA_y}: \qquad \max\left\{\mathcal{D}_A(y)=r_A^Ty-\alpha\sqrt{y^TG_Ay}\;\;\middle|\;\;\|y\|\le 1\right\},$$

where

$$G_A=\sum_{k=1}^{K}G_k+S\Gamma S^T. \tag{3.4}$$

The matrices S, Γ, and G_k above are defined by (2.13), (2.12), and (2.10), respectively. Each G_k can be computed independently for each block $k=1,\ldots,K$.

For a fixed α, the problem $\mathbf{DA_y}$ can be solved by Algorithm A for the homogeneous case of §2.3. And, analogous to Lemma 1, we have the following implication for any feasible problem **PA**:

$$y^TG_Ay=0 \;\Rightarrow\; r_A^Ty=0 \;\text{ for any } y\in\mathbb{R}^M. \tag{3.5}$$

To see this, assume without loss of generality that $K=1$ and $C=0$ and note that the feasibility of **PA** implies the feasibility of the systems $y^TA\hat{X}\xi+y^Tr_A=0$ and $B\hat{X}\xi=0$ for any $y\in\mathbb{R}^M$. Then, for $r_A^Ty\neq 0$, the linear form $c^T(y)\xi=y^TAX\xi$ cannot be constant on the linear space $L_A=\{\xi\in\mathbb{R}^n\mid B\hat{X}\xi=0\}$. Consequently, $y^TG_Ay=\|P_Ac(y)\|^2\neq 0$.

For $G_A=QDQ^T$ and $g=Q^Tr_A$, where Q is orthonormal and $D=\operatorname{diag}\{d_1,\ldots,d_M\}\ge 0$, it is also easy to see from (3.5) that the condition $d_i=0$ implies $g_i=0$ (for otherwise **PA** is infeasible). Hence, for

$$\min\{\mathcal{E}_A(\xi)\mid\xi\in L\cap\mathcal{B}_\alpha\}=\mathcal{D}_A^*>0$$

or, equivalently, for

$$\rho_A=\sqrt{g_+^TD_+g_+}=\max\{r_A^Ty\mid y^TG_Ay\le 1\}>\alpha,$$

problem $\mathbf{DA_y}$ has a unique solution y^*, which can be computed by Procedure $\mathcal{U}^*$ of §2.3. As before, D_+ and g_+ are the positive partitions of D and g, respectively. Substitute y^* in (2.8) to find $c^*=c(y^*)$. Then project c^* onto L_A by the formulae (2.14)–(2.15), omitting (2.16). The resulting projection η_A^* yields the exact optimal solution $\xi_A^*=-\alpha\eta_A^*/\|\eta_A^*\|$ of the affine error-reduction subproblem **RA**. In practice, it

is advantageous to perform a line search to minimize the error function $\mathcal{E}^2(\hat{X}(e+\beta\xi_A^*))$ over $\beta \in [0, \gamma\beta_{\max}]$, where

$$\beta_{\max} = \max\{\beta > 0 \mid e + \beta\xi_A^* \geq 0\} = \min\{-1/(\xi_A^*)_j \mid (\xi_A^*)_j < 0,\ j = 1, \ldots, n\}$$

is the maximum step length toward the boundary of the nonnegative orthant $\mathbb{R}^n_+$. The scalar $\gamma \in (0,1)$ is a fixed "back-off" factor. Since $\mathcal{E}^2(\hat{X}(e+\beta\xi_A^*))$ is a quadratic function in β, the line search can be done exactly. Although theoretically such a "long-step" variant of the affine scaling algorithm may not converge for $\gamma > 2/3$ [10, 27], our experiments encountered no difficulties with a value of $\gamma = 0.95$, which also seems to be the customary choice of this parameter for the standard variant of the affine scaling algorithm for linear programming ($K = M = 1$).

If $\rho_A \leq \alpha$, then by the duality relation (3.3) and Lemma 2, $\min\{\mathcal{E}(\xi) \mid \xi \in L_A \cap \mathcal{B}_\alpha\} = \mathcal{D}_A^* = 0$. This means that **RA** has an exact solution in $L_A \cap \mathcal{B}_\alpha$. Such a solution can be found by applying Procedure $\mathcal{U}^*$ with $\alpha = \rho_A$, which can be shown exactly as in the proof of Lemma 3. Moreover, for $\rho_A \leq 1$, the so-computed solution is guaranteed to be nonnegative. In other words, the algorithm terminates in one step whenever $\rho_A \leq 1$.

Finally, it is easy to see that for any infeasible problem **RA** we still have $\rho_A \to +\infty$ with the number of iterations since the convex error function $\mathcal{E}(x)$ decreases by at least a factor of $(1 - 1/\rho_A)$ at each iteration. Unfortunately, in contrast with the projective Algorithm A of §2.5, we are not able to claim that $\rho_A \leq \sqrt{n(n-1)}$ for any feasible problem **RA**.

The affine scaling algorithm performs the following steps, starting from a strictly block-feasible solution $\hat{x}_k$ for each $k = 1, \ldots, K$. Initially we set $\alpha \in (0,1)$ and, to guarantee convergence, we select some $\gamma \in (0, 2/3)$. The algorithm also assumes that the desired feasibility tolerance $\varepsilon_{\text{feas}}$, a binary search tolerance ε_{bin}, and a large constant R have been specified.

ALGORITHM A′.

while $\|r\| = \left\|\sum_{k=0}^K A_k\hat{x}_k - b_0\right\| > \varepsilon_{\text{feas}}$ **do**
  Use Cholesky factorization to compute $F_k = \left(B_k\hat{X}_k^2B_k^T\right)^{-1}$ for $k = 1, \ldots, K$.
  $G := \sum_{k=1}^K G_k = \sum_{k=1}^K A_k\hat{X}_kP_k\hat{X}_kA_k^T$, where $P_k = I_k - \hat{X}_kB_k^TF_kB_k\hat{X}_k$.
  $\Gamma := \left(I_N + \sum_{k=1}^K \hat{X}_0C_k^TF_kC_k\hat{X}_0\right)^{-1}$.
  $S := A_0\hat{X}_0 - \sum_{k=1}^K A_k\hat{X}_k^2B_k^TF_kC_k\hat{X}_0$.
  $G := G + S\Gamma S^T$.
  Compute Q and $D = \text{diag}(d_1, \ldots, d_M) \geq 0$ such that $G = QDQ^T$ and $QQ^T = I$.
  $g := Q^Tr \in \mathbb{R}^M$. Define $\mathcal{M} = \left\{ i \in M \mid |g_i| > \varepsilon_{\text{feas}}/\sqrt{M} \right\}$
  $\rho' := \sqrt{\sum_{i\in\mathcal{M}} g_i^2/d_i}$
  if $\rho' \geq R$, **then stop**
    elseif $1 < \rho' < R$ **then** $t_{\max} := \alpha^{-1}\sqrt{g^TDg}$. Compute t (binary search
      on $[0, t_{\max}]$) such that $\left|\alpha - \sqrt{\sum_{i\in\mathcal{M}} g_i^2d_i/(t+d_i)^2}\right| \leq \alpha\varepsilon_{\text{bin}}$.
    else $(\rho' \leq 1)$ $t := 0$, $\gamma := 1$.
  endif
  $u_i^* := g_i/(t+d_i)$ if $i \in \mathcal{M}$ and 0 otherwise.
  $y^* := Qu^*/\|Qu^*\|$.
  $c_k^* := \hat{X}_kA_k^Ty^*$, $k = 0, 1, \ldots, K$.
  $\xi_0^* := \Gamma(c_0^* - \sum_{k=1}^K \hat{X}_0C_k^TF_kB_k\hat{X}_kc_k^*)$.

19 $\xi_k^* := P_k c_k^* - \hat{X}_k B_k^T F_k C_k \hat{X}_0 \xi_0^*$.

20 $\xi^* := -(\xi_0^*, \xi_1^*, \ldots, \xi_k^*)$. Normalize ξ^*.

21 $\beta_{\max} := \min\left\{ -1/\xi_j^* \mid \xi_j^* < 0,\ j = 1, \ldots, n \right\},\ \beta_{\text{opt}} = r^T A\hat{X}\xi^* / \|A\hat{X}\xi^*\|^2$.

22 **if** $\beta_{\text{opt}} \leq \gamma\beta_{\max}$ **then** $\beta^* := \beta_{\text{opt}}$ **else** $\beta^* := \beta_{\max}\gamma$ **endif**

23 $\hat{x}_k := \hat{X}_k(e + \beta^* \xi_k^*)$, for $k = 0, 1, \ldots, K$.

24 **endwhile**.

Note that g in line 8 of Algorithm A′ is the error vector r in the orthogonal system Q and u^* in line 15 is the direction for updating the current iterate in the same coordinate system. The composition of the set $\mathcal{M}$ is designed so as to leave out those components of g whose total contribution to the error does not exceed $\varepsilon_{\text{feas}}$. This practical device enhances the stability of computing u^* in line 15. Another possible source of numerical error is the gradual deterioration of the condition of the scaled matrices $B_k X_k$ in line 2. Should the Cholesky factorization of the matrices $B_k \hat{X}_k^2 B_k^T$ suffer from ill-conditioning for some blocks k, appropriate pivoting on $(C_k\hat{X}_0, B_k\hat{X}_k)$ produces one or more nearly zero rows of $B_k\hat{X}_k$. As outlined in Remark 1 of §2, the corresponding rows of the so-transformed $C_k\hat{X}_0$ and the right-hand side for each such ill-conditioned block k can be collected in a system of equations with respect to the coupling variables x_0. This rearrangement either sets the values of x_0 for this iteration or leads to the elimination of some components of x_0 and the creation of an additional $(k+1)$st block of slack variables with $B_{k+1} = I$.

A *simplified* version of the algorithm is obtained by bypassing lines 3–15 and by replacing line 16 with $y^* = r/\|r\|$. This corresponds to the selection of large values of t in the main algorithm, $t >> \max\{ d_1, \ldots, d_M \}$, which is equivalent to minimizing the error function $\mathcal{E}(\xi)$ over $L \cap \mathcal{B}_\alpha$ for a small α. The main advantage of this well-known scaled affine gradient descent method (see, e.g., [8, 5]) is that it avoids the computation of the matrix G. It is still necessary to project $c(y^*)$ onto L, but this can be done reasonably fast. In many instances, the simplified algorithm is very effective initially, but the error reductions it achieves per iteration often degrade rapidly. It seldom reaches the commonly desired feasibility tolerances. Nevertheless, its attractive initial performance offers convenient preprocessing steps for Algorithm A′.

Algorithm A′ can be readily applied to problems (1.2)–(1.3) having coupling *inequalities* $A_0x_0 + \sum_{k=1}^K A_k x_k \leq b_0$ instead of equalities (1.2). It suffices to create an additional block of slack variables x_k, $k = K+1$, with $A_k = I$, $C_k = 0$, and vacuous B_k. Even though the matrix $(B_k^T X_k^2 B_k^T)$ does not exist for this block, the projection matrix P_k in line 3 of the algorithm is well defined and equal to the identity matrix.

Finally, it is easy to verify that each iteration of Algorithm A or A′ can be implemented to run in a number of arithmetic operations bounded by (1.4).

4. Computational results. The objective of our computational experiments was to investigate whether the number of iterations of the methods we have outlined compares favorably with results obtained by using interior point methods to the overall problem without taking advantage of its structure. We implemented the special case of the affine Algorithm A′ of §3 to compute feasible solutions of *block-angular* linear systems of equations in nonnegative variables. This preliminary experimental implementation was coded in FORTRAN-77 and developed on a SUN SS10/41 workstation. All tests were repeated on an IBM RS/6000 Model 580 with 256Mb of RAM running under the AIX 3.2 operating system. Execution times were also measured by calling this system's MCLOCK routine.

Test problems. For these experiments we generated block-angular constraints with $K = 32, 64, 128, 256$ blocks, $M = 63, 127, 257, 511, 1023$ coupling rows, and blocks of fixed dimension $(m_k, n_k) = (100, 299)$. Each (m_k, n_k)-submatrix B_k and each (M, n_k)-submatrix A_k was 5% dense with positive elements not exceeding 1,000 in magnitude, randomly drawn from a gamma distribution. In order to simulate the so-called "dense column problem" which arises in factorizations used in interior point methods (see, e.g., [30]), we created five randomly-placed columns of 50% density for each A_k. The desired overall 5% density for each submatrix was achieved by randomly placing nonzeros so that no row or column remained uncovered. The right-hand side of the problem was created by first postulating a nonnegative $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_K)$, and then by setting $b_k = B_k \tilde{x}_k$ for each k and $b_0 = \sum_{k=1}^{K} A_k \tilde{x}_k$. The postulated $\tilde{x}$ was a 50% dense vector with random nonzeros in the range $(0, 1)$. Since this $\tilde{x}$ contains a substantial proportion of small components, it takes into account the effect of highly overestimated variable bounds used by interior point methods to compute an initial scaling.

A test instance was thus created by discarding $\tilde{x}$ and by asking for a solution $x = (x_1, \ldots, x_K)$ of the generated feasible problem:

$$\begin{aligned} &\sum_{k=1}^{K} A_k x_k = b_0, \\ &B_k x_k = b_k, \quad x_k \geq 0, \qquad k = 1, \ldots, K. \end{aligned} \tag{4.1}$$

Table 1 gives the sizes of these test instances. Although the densities of 5% for each A_k and B_k translate to overall densities of 0.03 to 1.33%, the total number of nonzeros is substantial even for the sparser problems. Note that the $5K$ dense columns in the coupling rows, whose inclusion is designed to cause fill-in in step 3 of our algorithm, are not dense for methods that treat the problem in its entirety.

Initialization. The affine Algorithm A′ requires a strictly feasible block initial solution $\hat{x}$, which we construct by appending an artificial variable ζ_k to each block variable x_k, the zero column to each A_k, and the column $b_k - B_k e$ to each block matrix B_k. We also append an extra coupling row to force the sum of the artificials to zero. That is, the so-augmented problem has the form

$$\begin{aligned} &\sum_{k=1}^{K} \begin{pmatrix} A_1 & 0 \\ 0 & \mu \end{pmatrix} \begin{pmatrix} x_k \\ \zeta_k \end{pmatrix} = \begin{pmatrix} b_0 \\ 0 \end{pmatrix}, \\ &(B_k, b_k - B_k e) \begin{pmatrix} x_k \\ \zeta_k \end{pmatrix} = b_k, \quad x_k, \zeta_k \geq 0, \qquad k = 1, \ldots, K, \end{aligned} \tag{4.2}$$

where $\mu > 0$ is a large penalty parameter (see below). The vector of all ones, $(\hat{x}_1, \zeta_1, \hat{x}_2, \zeta_2, \ldots, \hat{x}_K, \zeta_K) = e$, is a strictly feasible block solution, but it violates the coupling equations of (4.2). In our test problems, the 2-norm of the initial violation of the first M coupling equations of (4.2) was in the range of 10^5 to 10^7.

Iterative scheme. The rest of our implementation follows Algorithm A′, except for a heuristic which alternates between simplified and interior point iterations. Initially, at least $M/6$ simplified iterations are performed and such iterations continue for as long as the improvement in the 2-norm of the error in the coupling constraints from one iteration to the next is more than 10%. Then an interior point iteration is invoked, followed by a number of simplified iterations for as long as they are "worthwhile." This is decided on the basis of the running time and improvement of the error provided by

the last interior point iteration versus the running time and improvement provided by the simplified iteration. For the larger values of M, interior point iterations are substantially more expensive than simplified iterations. Unfortunately, the effectiveness of the simplified iterations deteriorates to the point where an "investment" in another interior point iteration is unavoidable.

This process is repeated until the final interior point iteration moves to a solution. Even though we use sparse linear algebra operations wherever possible, our implementation employs the standard Cholesky and Schur factorization routines of LAPACK [2], with no preconditioning. We made no effort to "tune" our experimental code in order to increase pipelining in the RS/6000 and expect that its running time can be improved by more careful programming. Substantial speedups can be expected if all block operations are performed in parallel and if finer-grain parallel schemes for all linear-algebraic operations, in particular for inner products, are employed.

Experiments. In these experiments we used the following parameters and tolerances: $\alpha = 0.995$, back-off factor used in line search 0.995, binary search tolerance 10^{-8} initially and 10^{-10} for the final interior point iteration, feasibility accuracy for the 2-norm of the coupling constraint error 10^{-10}, penalty for artificial variables $\mu = 10^{12}$, and tolerance for artificials 10^{-10}. Once every artificial variable drops below this tolerance, the algorithm renders the artificial row (the $M + 1$st coupling constraint in (4.2)) ineffective by setting its right-hand side element to the left-hand side. This freezes the artificials at their current values, which is equivalent to deleting them from the problem.

In all cases, the accuracy of the computed solution x^* was as follows: value of artificials $\leq 10^{-10}$; maximum violation in any coupling equation $\leq 10^{-10}$; and maximum violation in any block equation $\leq 10^{-9}$, relative to the right-hand sides.

The results listed in Table 2 indicate that the number of interior point iterations for all problems is small, provided that these iterations are interspersed with simplified iterations as described above. This is necessary for two reasons. First, long sequences of simplified iterations can quickly deteriorate, causing the algorithm to stall at an unacceptably high error norm. Second, interior point iterations, which are very productive during the latter stages of the algorithm, provide only meager improvements early on. Running times increase linearly with the number of blocks and nonlinearly with the number of coupling rows. Further scrutiny of Tables 1 and 2 reveals that, for fixed block sizes, the running time per nonzero matrix element varies narrowly (1 to 2.6 ms/element) for the values of K and M in the test set.

Comparison with existing LP codes. A natural question is whether our interim implementation might be competitive in terms of running time with current simplex LP codes. We considered the performance of two such codes, MINOS [18] and IBM's OSL [21]. The former, essentially a nonlinear programming code, provided an unattractive performance for the few problems we tried. OSL offers several primal simplex strategies for block-angular linear programs. We experimented with the fastest of these options, EKKLPDC, an initialization heuristic which splits b_0 into K parts, solves block problems separately by the simplex method, constructs an initial basis using these (possibly infeasible) block bases, and reverts to the simplex method to solve the overall problem. This was also unsuccessful in solving most of our larger problems within a reasonable time. Solution times for the few problems solved in this fashion are shown in Table 3.

A more meaningful comparison is with an interior point method. For this purpose we used OSL's call EKKBSLV, which implements a primal-dual predictor-corrector interior point method [17]. This code spends almost all of its time forming the initial

overall adjacency matrix of AA^T and the Cholesky factor. Their size grows steeply with the problem dimension and the number of nonzeros (2.30 – 4.24 elements in the adjacency matrix and 4.25 – 7.12 elements in the Cholesky factor are created for each nonzero in the original matrix). The code solves the smaller problems very efficiently, but it requires an enormous amount of storage for the larger problems, three of which could not proceed through the initial adjacency matrix task, while larger storage declarations prevented the executable module from loading. In any case, the

TABLE 1. *Block-angular test problems.*

K	$M = 63$		$M = 127$		$M = 255$		$M = 511$		$M = 1023$	
	m	n	m	n	m	n	m	n	m	n
	nz	den†	nz	den	nz	den	nz	den	nz	den
32	3,263	9,568	3,327	9,568	3,455	9,568	3,711	9,568	4,223	9,568
	75,392	0.24	106,017	0.33	167,329	0.51	290,056	0.82	535,715	1.33
64	6,463	19,136	6,527	19,136	6,655	19,136	6,911	19,136	7,423	19,136
	150,784	0.12	212,036	0.17	334,658	0.26	580,110	0.44	1,071,427	0.75
128	12,863	38,272	12,927	38,272	13,055	38,272	13,311	38,272	13,823	38,722
	301,568	0.06	424,073	0.09	669,315	0.13	1,160,214	0.23	2,142,851	0.40
256	25,663	76,544	25,727	76,544	25,855	76,544	26,111	76,544	26,623	76,544
	603,136	0.03	848,149	0.04	1,338,630	0.07	2,320,419	0.12	4,285,699	0.21

† m, n, nz, and den are the rows, columns, nonzeros, and density (in %) of the overall matrix, respectively. All blocks are of dimension $(m_k, n_k) = (100, 299)$.

TABLE 2. *Performance of the affine Algorithm A′.*

K	$M = 63$ †			$M = 127$			$M = 255$			$M = 511$			$M = 1023$		
	S	I	T‡	S	I	T	S	I	T	S	I	T	S	I	T
32	42	4	46	52	3	55	87	2	89	173	2	175	172	1	173
	0:52	0:15	1:07	1:05	0:26	1:30	1:51	0:51	2:42	3:53	3:18	7:11	4:10	8:00	12:10
64	43	4	47	52	3	55	87	3	90	173	3	176	172	2	174
	1:46	0:30	2:16	2:09	0:49	2:59	3:42	2:19	6:01	7:41	8:14	15:56	8:19	23:46	32:04
128	42	5	47	62	4	66	87	3	90	173	3	176	343	3	346
	3:27	1:14	4:40	5:08	2:09	7:17	7:23	4:23	11:45	15:22	14:47	30:09	36:05	1:05:32	1:34:32
256	46	6	52	71	5	76	95	5	100	218	4	222	343	3	346
	7:31	2:57	10:28	11:45	5:19	17:04	16:06	14:15	30:20	38:39	37:12	1:15:51	1:09:37	1:46:15	2:55:52

†Algorithm A′ solves problem (4.2), which has $M + 1$ coupling rows, $n_k + 1$ columns per block, and about 1% more nonzeros than problem (4.1).

‡ [S I T] ≡ [Simplified, Interior point, Total] *iterations*, and CPU times in *h:mm:ss* of an IBM RS/6000 Model 580.

TABLE 3. *Solving some smaller problems by primal simplex with block start* (EKKLPDC).

K	$M = 63$		$M = 127$		$M = 255$	
	pivots	*time*†	*pivots*	*time*	*pivots*	*time*
32	8,892	13:51	13,400	30:16	19,368	1:11:23
64	18,785	1:09:04	28,686	2:43:22	‡	>3:20:10

† All dimensions and *time* as defined in Table 4.
‡ Abandoned after indicated time.

FIG. 1. EKKBSLV *actual and projected execution times.*

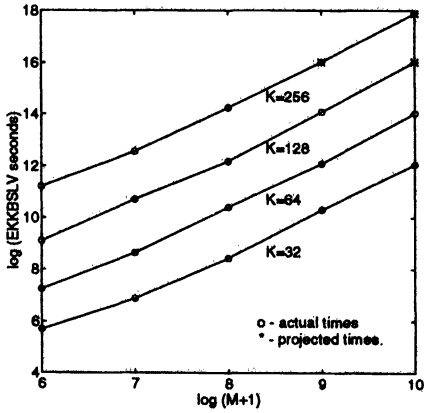


TABLE 4. *Solution of the test problems by* OSL*'s* EKKBSLV*, a primal-dual predictor-corrector interior point implementation.*

K	$M = 63$	$M = 127$	$M = 255$	$M = 511$	$M = 1023$
	nza	*nza*	*nza*	*nza*	*nza*
	nzc †	*nzc*	*nzc*	*nzc*	*nzc*
	iter *time*	*iter* *time*	*iter* *time*	*iter* *time*	*iter* *time*
32	175,184	283,755	516,156	1,034,530	2,272,697
	321,441	523,626	926,964	1,800,489	3,813,815
	9 0:52	10 1:58	9 5:44	9 21:03	10 1:10:28
64	348,084	559,925	1,001,090	1,940,696	4,024,965
	641,287	1,032,703	1,812,605	3,448,479	6,943,301
	9 2:32	10 6:39	9 22:29	9 1:12:38	10 4:39:36
128	693,642	1,111,745	1,970,478	3,757,459	[7.6×10^6]
	1,280,769	2,050,558	3,583,224	6,734,747	[13.3×10^6]
	9 9:10	9 27:48	9 1:16:57	10 4:50:58	* [18.5 hrs]
256	1,385,331	2,216,115	3,908,209	[7.4×10^6]	[14.6×10^6]
	2,560,336	4,086,192	7,124,682	[13.3×10^6]	[25.8×10^6]
	9 39:14	10 1:40:38	9 5:24:57	* [18.5 hrs]	* [67.0 hrs]

† *nza, nzc:* nonzeros in the initial adjacency matrix and Cholesky factor, respectively. *Time:* CPU time in *hh:mm:ss* of an IBM RS/6000 M580 with 256Mb of RAM, running OSL [21] under AIX 3.2.

* Run was aborted by OSL due to insufficient storage in forming the adjacency matrix. Counts and times in square brackets are estimates (see Figure 1).

estimated running times for these three problems are so long that their solution would not have been instructive. For the problems solved to completion, the running time per nonzero matrix element increased rapidly with K and M (from about 1 to 16 ms/element), and the maximum speedup of the affine Algorithm A′ relative to EKKBSLV was 10.7. The corresponding projected times for the three problems we were unable to solve by EKKBSLV are 31, 29, and 57 ms/element, and the speedups are 12, 15, and 23. (See Table 4 and Figure 1.)

The computational results presented here are hardly definitive, but they do seem to support the view that serial implementations of interior point algorithms that take advantage of the block-diagonal structure in *large, weakly-coupled* problems *can* provide speedups of up to 20 or more, versus solving the problem as a whole. This speedup figure discounts the experimental nature of our implementation, the efficiency of a commercial interior point code such as OSL's EKKBSLV, and the ease with which the structured affine algorithm would lend itself to a coarse-grain parallel implementation by distributing all block operations. As we have shown, the general bordered block-diagonal structure can be handled by the same algorithm but with some additional linear-algebraic operations. Further experiments would be needed to extend our observations to this more general structure.

5. Compact inverse-based methods. In the previous sections we examined a structured projective method for solving the general bordered block-diagonal problem **PA**. Here we show that such problems can also be solved in $O(\sqrt{n}L)$-iterations, each of which has the arithmetic complexity (1.4).

The iterative computation of search directions for various existing $O(\sqrt{n}L)$-iteration LP algorithms (see, e.g., [22, 9, 5, 17, 19, 20, 31]) reduces to the solution of a least-squares system of linear equations $HD^2H^Tx = h$, where D is a positive diagonal matrix and

$$H = \begin{pmatrix} A_0 & A \\ C & B \end{pmatrix} \tag{5.1}$$

is the constraint matrix of problem **PA**. In order to simplify notation, we first apply the D-scaling to H to produce the partitioned matrix

$$\mathcal{H} = HD = \begin{pmatrix} \mathcal{A}_0 & \mathcal{A} \\ \mathcal{C} & \mathcal{B} \end{pmatrix}. \tag{5.2}$$

The inverse of $\mathcal{H}\mathcal{H}^T = HD^2H^T$ can then be written as follows:

$$\begin{pmatrix} G^{-1} & 0 \\ -\mathcal{F}(\mathcal{C}|\mathcal{B})(\mathcal{A}_0|\mathcal{A})^T G^{-1} & \mathcal{F} \end{pmatrix} \begin{pmatrix} I & -(\mathcal{A}_0|\mathcal{A})(\mathcal{C}|\mathcal{B})^T\mathcal{F} \\ 0 & I \end{pmatrix}, \tag{5.3}$$

where

$$\mathcal{F} = \left(\mathcal{B}\mathcal{B}^T + \mathcal{C}\mathcal{C}^T\right)^{-1} = F\left(I - \mathcal{C}\Gamma\mathcal{C}^T F\right)$$

is a well-known identity, with

$$F = \left(\mathcal{B}\mathcal{B}^T\right)^{-1}, \quad \Gamma = \left(I + \mathcal{C}^T F\mathcal{C}\right)^{-1}, \tag{5.4}$$

and

$$G = (\mathcal{A}_0|\mathcal{A})\left(I - (\mathcal{C}|\mathcal{B})^T\mathcal{F}\ (\mathcal{C}|\mathcal{B})\right)(\mathcal{A}_0|\mathcal{A})^T. \tag{5.5}$$

The factorization (5.3) can be verified directly. For the block-diagonal case, i.e.,

$$B = \operatorname{diag}(B_1, \ldots, B_K), \quad A = (A_1|\ldots|A_K), \quad \text{and} \quad C = \left(C_1^T, \ldots, C_K^T\right)^T,$$

the matrix F has the form $F = \mathrm{diag}\left(\left(\mathcal{B}_1\mathcal{B}_1^T\right)^{-1}, \ldots, \left(\mathcal{B}_K\mathcal{B}_K^T\right)^{-1}\right)$. By rewriting Γ in (5.4) as

$$\Gamma = \left(I + \sum_{k=1}^{K} \mathcal{C}_k^T \left(\mathcal{B}_k\mathcal{B}_k^T\right)^{-1} \mathcal{C}_k\right)^{-1},$$

we see that this expression coincides with (2.12) for the case $D = \mathrm{diag}(\hat{X})$, $\mathcal{B}_k = B_k\hat{X}_k$, and $\mathcal{C}_k = C_k\hat{X}_k$. Furthermore, the Schur complement (5.5) can be represented in the form

$$G = \sum_{k=1}^{K} \mathcal{A}_k \left(I - \mathcal{B}_k^T(\mathcal{B}_k\mathcal{B}_k^T)^{-1}\mathcal{B}_k\right) \mathcal{A}_k^T + S\Gamma S^T,$$

with

$$S = \mathcal{A}_0 - \sum_{k=1}^{K} \mathcal{A}_k\mathcal{B}_k^T(\mathcal{B}_k\mathcal{B}_k^T)^{-1}\mathcal{C}_k,$$

which is identical to the definition (3.4) with $D = \mathrm{diag}(\hat{X})$. In view of the compact-inverse representation (5.3), the system $HD^2H^Tx = h$ for the bordered block-diagonal matrix (5.1) can be solved in a number of arithmetic operations bounded by (1.4).

REFERENCES

[1] P. Brucker, *An $O(n)$ algorithm for quadratic knapsack problems*, Oper. Res. Lett., 3 (1984), pp. 163–166.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[3] J. R. Birge and L. Qi, *Computing block-angular Karmarkar projections with applications to stochastic programming*, Management Sci., 34 (1988), pp. 1472–1479.

[4] I.C. Choi and D. Goldfarb, *Solving multicommodity network flow problems by an interior point method*, in Large-Scale Numerical Optimization, Papers from the Workshop held at Cornell University, T.F. Coleman and Y. Li, eds., SIAM Proc. in Applied Math. 46, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 58–69.

[5] D. Den Hertog, *Interior Point Methods for Linear Programming*, Master's thesis, Faculty of Math. and Informatics, Technical University, Delft, The Netherlands, 1989.

[6] D. Goldfarb and M.J. Todd, *Chapter* II. *Linear programming*, in Handbook of OR & MS, G. Nemhauser, A. Rinnooy Kan, and M. Todd, eds., Elsevier Science (North–Holland), Amsterdam, 1989.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[8] C. C. Gonzaga and L. A. Carlos, *A Primal Affine-Scaling Algorithm for Linearly Constrained Convex Programs*, Tech. report ES-238/90, Dept. of Systems Engineering and Computer Science, COPPE Federal University of Rio de Janeiro, 1990.

[9] C. C. Gonzaga and M. J. Todd, *An $O(\sqrt{n}L)$-iteration large-step primal-dual affine algorithm for linear programming*, SIAM J. Optim., 2 (1992), pp. 331–342.

[10] L. A. Hall and R. J. Vanderbei, *Two-Thirds Is Sharp for Affine Scaling*, Tech. report 92–44, DIMACS, Rutgers University, New Brunswick, NJ, 1992.

[11] J. A. Kaliski and Y. Ye, *A decomposition variant of the potential reduction algorithm for linear programming*, Management Sci., 39 (1993), pp. 757–776.

[12] S. Kapoor and P.M. Vaidya, *Fast algorithms for convex quadratic programming and multicommodity flows*, in Proc. of the 18th Annual ACM Symp. on Theory of Computing, Berkeley, CA, 1986, pp. 147–159.

[13] ———, *Speeding-up Karmarkar's algorithm for multicommodity flows*, Math. Programming, 73 (1996), pp. 111–127.

[14] N. KARMARKAR, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[15] L. G. KHACHIYAN, *On the Complexity of Linear Programs*, Math. and Cybernetics Series, Znanie Publishers, Moscow, 1987. (In Russian.)

[16] S. MEHROTRA, *Variants of Karmarkar's Algorithm: Theoretical Complexity and Practical Implementation*, Ph.D. thesis, Dept. of Industrial Engineering and Operational Research, Columbia University, New York, NY, 1987.

[17] ———, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[18] B. A. MURTAGH AND M. A. SAUNDERS, *MINOS* 5.4 *User's Guide*, Tech. Report SOL-83-20R, Dept. of Operational Research, Stanford University, Stanford, CA, December 1983 (rev. March 1993).

[19] R. C. MONTEIRO AND I. ADLER, *Interior path-following primal-dual algorithms*, Math. Programming, 44 (1988), pp. 37–41.

[20] YU. NESTEROV AND A. NEMIROVSKY, *Self-concordant functions and polynomial time methods in convex programming*, SIAM Studies in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[21] IBM CORPORATION, *Optimization Subroutine Library (OSL) Guide and Reference, Release* 2, Pub. SC23-0519-02, IBM Corp., Kingston, NY, 1991.

[22] J. RENEGAR, *A polynomial-time method algorithm, based on Newton's method, for linear programming*, Math. Programming, 40 (1988), pp. 59–93.

[23] T.R. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[24] A. SCHRIJVER, *Theory of Linear and Integer Programming*, John Wiley, New York, 1986.

[25] M. J. TODD, *Exploiting special structure in Karmarkar's algorithm for linear programming*, Math. Programming, 41 (1988), pp. 97–113.

[26] M. TORABI, *Decomposed block Cholesky factorization in the Karmarkar algorithm. Solving a class of super large LP problems*, Comput. Math. Appl., 20 (1990), pp. 1–7.

[27] T. TSUCHIYA AND M. MARUMATSU, *Global Convergence of the Long-Step Affine Scaling Algorithm for Degenerate Linear Programming Problems*, Res. Memorandum 423, The Institute of Statistical Mathematics, Tokyo, January 1992 (rev. September 1992).

[28] P. M. VAIDYA, *Speeding-up linear programming using fast matrix multiplication*, in Proc. of the 30th Annual Symp. on Foundations of Computer Science, Los Alamitos, CA, IEEE Computer Soc. Press, 1990, pp. 332–337.

[29] J. P. VIAL, *Decomposition of structured linear programs based on analytic centers*, in Lecture Notes in Economics and Mathematical Systems 353, Springer-Verlag, Berlin, 1991, pp. 190–203.

[30] M. H. WRIGHT, *Interior methods for constrained optimization*, in Acta Numerica 1992, A. Iserles, ed., Cambridge University Press, New York, NY, 1992, pp. 341–407.

[31] Y. YE, *An* $O(n^3L)$ *potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

[32] Y. YE AND E. TSE, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming, 44 (1989), pp. 157–179.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 933–960, November 1996
© 1996 Society for Industrial and Applied Mathematics
003

SOLVING LP PROBLEMS VIA WEIGHTED CENTERS*

AIPING LIAO† AND MICHAEL J. TODD‡

Abstract. The feasibility problem for a system of linear inequalities can be converted into an unconstrained optimization problem by using ideas from the ellipsoid method, which can be viewed as a very simple minimization technique for the resulting nonlinear function. This function is related to the volume of an ellipsoid containing all feasible solutions, which is parametrized by certain weights which we choose to minimize the function. The center of the resulting ellipsoid turns out to be a feasible solution to the inequalities. Using more sophisticated nonlinear minimization algorithms, we develop and investigate more efficient methods, which lead to two kinds of weighted centers for the feasible set. Using these centers, we develop new algorithms for solving linear programming problems.

Key words. weighted center, the ellipsoid method, Newton's method, linear programming

AMS subject classifications. 65K, 90C

1. Introduction and a history of centers.

In this paper we will consider a linear programming problem of the form

$$\text{(LP)} \qquad \begin{array}{l} \min\ c^T x \\ l \le A^T x \le u \end{array}$$

and the associated feasibility problem

$$\text{(FP)} \qquad l \le A^T x \le u,$$

where A is an $n \times m$ matrix with full rank. We assume that $m \ge n \ge 2$ and $l < u$. As is well known, any linear programming problem can be reduced to so-called inequality form:

$$\text{(ILP)} \qquad \begin{array}{l} \min\ c^T x \\ \tilde{A}^T x \le b, \\ x \ge 0. \end{array}$$

If the input data are all integers, (ILP) is equivalent to

$$\text{(ILPB)} \qquad \begin{array}{l} \min\ c^T x \\ \tilde{A}^T x \le b, \\ (l_x =)0 \le x \le u_x, \end{array}$$

where $u_x = 2^L e$ with L the input length of (ILP) and e the vector of ones, which can be easily adapted to the form (LP) (see, e.g., Burrell and Todd [3]). So, theoretically speaking, our form (LP) is not restrictive. On the other hand, since "many linear

* Received by the editors August 23, 1993; accepted for publication (in revised form) August 10, 1995.

† Advanced Computing Research Institute and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853. The research of this author was partially supported by the Cornell Theory Center, which receives major funding from the National Science Foundation and IBM Corporation, with additional support from the State of New York and its Corporate Research Institutes; and by the NSF, the AFOSR, and the ONR through grant DMS-8920550.

‡ School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853. The research of this author was partially supported by the NSF, the AFOSR, and the ONR through grant DMS-8920550.

programming problems involve explicit upper bounds on individual variables [in the standard form]" (Chvátal [4])—in particular, the upper bounds may correspond to limits on resources—we thus often, in practice also, are confronted with linear programming problems in the form (ILPB), which as above can be converted to the form (LP). Thus (LP) is not very restrictive in practice either.

In §§2 and 3 we will discuss methods for solving the linear inequality system (FP) and introduce two weighted centers. Then §4 develops algorithms for solving (LP) via these two kinds of center. We also report preliminary encouraging numerical results. Section 5 presents a summary and conclusion.

One recent development in linear programming can be regarded as a study of various concepts of center for a polytope $P \subset R^n$. Such a center must be in the (relative) interior of P. Since 1984 when Karmarkar proposed his famous projective-scaling algorithm [9], interior-point methods have become the mainstream of research in linear programming. Basically speaking, in this field, there are two aspects under study. One is the appropriate definition of center, which provides the interior point; the other is the method of moving the center and the merit function for measuring the progress made by this movement.

The first kind of center of a polytope that was used in optimization is the center of gravity or centroid used by Levin [12] in his algorithm, the method of central sections, for minimizing a convex function f over a convex polytope P (see also Newman [18]). In that paper the centroid is used as the test point. If the current centroid, say x^k, satisfies the convergence criterion, stop with a satisfactory approximate solution; otherwise, cut the current polytope P^k into two pieces by a hyperplane through x^k whose normal is a subgradient of f at x^k, and replace P^k by that part containing all optimal solutions. The volume of the polytope under consideration is thus reduced, and this procedure can be repeated until a satisfactory centroid is obtained. This method is very concise and $1 - \exp(-1)$ is a guaranteed reduction of the volume of successive polytopes. The disadvantage is the difficulty of calculating the centroid.

Yudin and Nemirovskii [30] discuss the computational difficulties of Levin's method and propose a modified method of centered cross-sections, using ellipsoids instead of polyhedra. This modified method is computationally implementable. They also point out that this ellipsoid method is a special case of Shor's algorithm [21] with space dilation in the direction of the subgradient. Shor [22] independently developed the ellipsoid method. Later, in 1979, Khachiyan [10] showed that the ellipsoid method is a polynomial-time algorithm for solving linear programming problems.

We note that John's result [8] shows that for every convex polytope $P \subset R^n$, the minimum volume ellipsoid containing P exists and is unique. It is easy to show that the center of this minimum volume ellipsoid is in P; otherwise, by one step of the ellipsoid algorithm, a smaller ellipsoid containing P can be obtained. Thus, the center of this ellipsoid can be used as a center of P. However, the smallest ellipsoid containing P is hard to find in general, and so is its center.

Tarasov, Khachiyan, and Erlich [24] study the method of inscribed ellipsoids, using the center of the maximal inscribed ellipsoid for a polytope as a center of the polytope. In [11], Khachiyan and Todd discuss the problem of approximating the maximal inscribed ellipsoid and related problems. They also propose algorithms for finding these ellipsoids.

Karmarkar's projective-scaling algorithm [9] uses centers in a different way. At each iteration, the current interior point is mapped by a projective transformation which is chosen so that its image becomes a certain center (the *analytic* center, see

Sonnevend [23]) of the transformed feasible region. Then it is easy to move this transformed point in order to make a sufficient decrease in a certain potential function.

Renegar [19] also uses the analytical center to develop his algorithm for linear programming problems, which is polynomial-time bounded. The analytical center is easier to approximate compared with the centroid. Actually, in his algorithm, Renegar showed that an ε-analytical center is enough. Unlike the centroid, the analytical center is not analytically independent; it depends on the way in which the polytope P is represented. Renegar [19] makes use of this property to improve the convergence of his algorithm by adding some extra constraints.

In the paper [28] Vaidya introduces the *volumetric center* which is the center of the ellipsoid with largest volume among a certain set of ellipsoids that are contained in P and proposes an algorithm with a better global convergence rate and time complexity than the ellipsoid method.

The centers of P we propose here are the centers of the ellipsoids with the smallest volumes among certain sets of ellipsoids that contain P. These ellipsoids are of simpler structure and are relatively easier to construct.

2. Model I and the associated center. We consider the feasibility problem:

$$\text{find } \bar{x} \text{ such that } \bar{x} \in P := \{x : l \le A^T x \le u\}. \tag{1}$$

For convenience we denote $r := \frac{l+u}{2}$ and $s := \frac{u-l}{2}$. We also denote by e the all-one vector and by e_j the jth column of the identity matrix. In [3] Burrell and Todd proposed a parallel-cut ellipsoid algorithm based on the results of Todd [26].

Note that P can be alternately written as

$$P = \{x \in R^n : (a_i^T x - l_i)(a_i^T x - u_i) \le 0, i = 1, \ldots, m\}, \tag{2}$$

where a_i is the ith column of A and l_i, u_i are the corresponding components of l, u, respectively. Now choose a nonnegative diagonal matrix $D := \operatorname{diag}(d) := \operatorname{diag}(d_1, \ldots, d_m)$, and combine the inequalities above with weights d_i. We thus obtain a set

$$E := E(d) := \{x \in R^n : (A^T x - l)^T D(A^T x - u) \le 0\}. \tag{3}$$

It is obvious that $P \subset E$. We suppose that ADA^T is nonsingular. Then E is actually an ellipsoid. Further calculation shows that

$$E = \{x \in R^n : (x - x_c)^T ADA^T (x - x_c) \le x_c^T (ADA^T) x_c - l^T Du\}, \tag{4}$$

where $x_c := x_c(d) := (ADA^T)^{-1} ADr$ is the center of E.

If the current center violates some constraint, say, $l_i \le a_i^T x \le u_i$, by the results of Todd [26] we can construct a new ellipsoid that contains that part of the previous one between the parallel hyperplanes $a_i^T x = l_i$ and $a_i^T x = u_i$, and the volume of the ellipsoid decreases by a factor which is, at worst, $\exp(-\frac{1}{2(n+1)})$. Only one component of d changes in this process.

The idea of this paper is to consider the volume of $E(d)$, or some surrogate of this volume, as a nonlinear function of d; then by applying an efficient algorithm (usually a variant of Newton's method) to minimize this function, we obtain a d^* for which we can show that $x_c(d^*)$ lies in P.

There is another way in which ellipsoids of a form very similar to $E(d)$ arise. Suppose that we model (FP) as the linear programming problem

$$\text{(FPP)} \qquad \begin{array}{lrcrcl} \max & & & \lambda & & \\ & A^T x & + & e\lambda & \leq & u, \\ & -A^T x & + & e\lambda & \leq & -l. \end{array}$$

The dual of this problem is

$$\text{(FPD)} \qquad \begin{array}{lrcrcl} \min & u^T y_u & - & l^T y_l & & \\ & A y_u & - & A y_l & = & 0, \\ & e^T y_u & + & e^T y_l & = & 1, \\ & y_u, & & y_l & \geq & 0, \end{array}$$

a problem in Karmarkar's canonical form. Todd [27] and Ye [29] show that at each iteration of Karmarkar's algorithm applied to (FPD), an ellipsoid is naturally generated that contains all optimal solutions of (FPP). This ellipsoid has the form

$$E'(d, w, \nu) := \{x : x^T A D A^T x - 2w^T A^T x + \nu \leq 0\}$$

for $D := \operatorname{diag}(d) := Y_u^2 + Y_l^2$, where (y_u, y_l) is the current feasible iterate for (FPD) and $Y_l := \operatorname{diag}(y_l)$, $Y_u := \operatorname{diag}(y_u)$, which is similar to $E(d)$ above. Moreover, Todd and Ye show that the logarithm of the volume of this ellipsoid is closely related to Karmarkar's potential function and hence argue that Karmarkar's algorithm is typically much more efficient than the ellipsoid method, although the iterations are more expensive. Our motivation is similar, but our methods are much closer in spirit to the ellipsoid method. We hope that by using sophisticated methods to minimize the nonlinear function of d, we will need far fewer iterations than the ellipsoid method, even though our iterations are again more expensive. There are also considerable differences between our methods and the interpretation of Karmarkar's method above: w and ν are not generally equal to Dr and $l^T Du$, respectively, so $E(d)$ and $E'(d, w, \nu)$ typically differ. Further, in Karmarkar's method, y_u and y_l constitute a feasible solution to (FPD), while our d only has to lie in a set whose closure is the nonnegative orthant. Lastly, our method for updating d is very different from Karmarkar's for updating (y_u, y_l).

Consider the problem

$$(\mathrm{P}_{fh}) \qquad \begin{array}{l} \min\ v(d) := f(d) \cdot h(d) \\ \text{s.t.}\ \ d \in \mathcal{D}, \end{array}$$

where

$$\begin{aligned} f(d) &:= r^T D A^T (A D A^T)^{-1} A D r - l^T D u = x_c^T A D A^T x_c - l^T D u, \\ h(d) &:= (\det(A D A^T))^{-\frac{1}{n}}, \\ \mathcal{D} &:= \{d : d \geq 0, A D A^T \text{ is nonsingular}\}. \end{aligned}$$

Note that the volume of $E(d)$ is $\kappa_n \cdot (v(d))^{\frac{n}{2}}$, where κ_n is the volume of the unit ball in R^n. Liao and Todd [14] show that the Burrell–Todd algorithm is basically the coordinate descent method applied to (P_{fh}), together with rules for updating the bounds l and u. They also propose a simpler way to perform the updating.

Newton's method is typically fast in practice as well as in theory, while the coordinate descent algorithm is usually considered not fast enough for practical use. But

here the function $v(d)$ is a homogeneous function of degree 0, and hence it can easily be shown that

$$\nabla v(d) + \nabla^2 v(d)^T d = 0,$$
$$d^T \nabla v(d) = 0,$$

so d is the Newton direction, as well as a direction of constancy for the function, and therefore useless as a search direction. We could solve a constrained problem to alleviate this difficulty, but we still face a nonconvex minimization.

The first new model is as follows:

$$(\mathrm{P}_{f+h}) \qquad \begin{array}{l} \min\ F(d) := f(d) + h(d) \\ \text{s.t. } d \in \mathcal{D}. \end{array}$$

We note that if $f(d) > 0$, by the arithmetic–geometric mean inequality, $F(d) \geq 2\sqrt{v(d)}$. Moreover, it is easy to see that f is homogeneous of degree 1 and h of degree -1; thus, for any given $d \in \mathcal{D}$, $F(d)$ attains its minimum over the half-line $\{\lambda d : \lambda \in R, \lambda > 0\}$ at $d^* := \sqrt{h(d)/f(d)} \cdot d$ which gives the value $F(d^*) = 2\sqrt{v(d^*)}$.

2.1. Properties of (P_{f+h}). In the following we give some properties of (P_{f+h}).

PROPOSITION 2.1. *$\mathcal{D}$ defined above is a convex set.*

Proof. The straightforward derivation is omitted. □

Now we show that both $f(d)$ and $h(d)$ are convex functions over $\mathcal{D}$. We first prove some lemmas.

LEMMA 2.2.

$$(5) \qquad \begin{aligned} \frac{\partial (ADA^T)^{-1}}{\partial d_i} &= -(ADA^T)^{-1} a_i a_i^T (ADA^T)^{-1}, \\ \frac{\partial (\det(ADA^T))}{\partial d_i} &= \det(ADA^T) a_i^T (ADA^T)^{-1} a_i. \end{aligned}$$

Proof. The proof is easy using the rank-1 update formulae. □

PROPOSITION 2.3. *For $i = 1, \ldots, m$, the ith components of $\nabla f(d)$ and $\nabla h(d)$ are*

$$\frac{\partial f(d)}{\partial d_i} = -(a_i^T x_c - l_i)(a_i^T x_c - u_i),$$
$$\frac{\partial h(d)}{\partial d_i} = -\frac{1}{n} \det(ADA^T)^{-\frac{1}{n}} a_i^T (ADA^T)^{-1} a_i.$$

Proof. This result can be obtained by direct calculation using Lemma 2.2. □

If $B = (b_{ij})_{m \times n}$, $C = (c_{ij})_{m \times n}$ are two m-by-n matrices, the Schur product of B and C is defined by $B \circ C := (b_{ij} c_{ij})_{m \times n}$. The following result can be found, for example, in [7].

LEMMA 2.4. *If B and C are m-by-m positive semidefinite matrices, so is $B \circ C$.*

PROPOSITION 2.5. *f and h are convex functions over $\mathcal{D}$, and hence so is $F = f + h$.*

Proof. By using Lemma 2.2, direct calculations give the Hessian of f as follows:

$$(6) \qquad H_f = 2 \cdot \mathrm{diag}(A^T x_c - r) A^T (ADA^T)^{-1} A \cdot \mathrm{diag}(A^T x_c - r),$$

which is obviously positive semidefinite for all $d \in \mathcal{D}$. As for h, calculations using Lemma 2.2 give us the following expression for the (i,j)-element of the Hessian of h:

$$H_h(i,j) = \frac{1}{n^2} \det(ADA^T)^{-\frac{1}{n}} [a_i^T (ADA^T)^{-1} a_i \cdot a_j^T (ADA^T)^{-1} a_j + n(a_i^T (ADA^T)^{-1} a_j)^2]. \tag{7}$$

Let $\nu_i := (ADA^T)^{-\frac{1}{2}} a_i$. Then

$$H_h(i,j) = \frac{1}{n^2} \det(ADA^T)^{-\frac{1}{n}} [(\nu_i^T \nu_i) \cdot (\nu_j^T \nu_j) + n(\nu_i^T \nu_j)^2], \tag{8}$$

which shows that H_h is positive semidefinite, since the first term in the square brackets is a matrix of a product of a vector by its transpose, which is thus positive semidefinite, while the second term is positive semidefinite from Lemma 2.4. □

We now make the following assumption:

$$\text{(A1)} \qquad \text{int}(P) \neq \emptyset \text{ and } \|a_i\| \neq 0 \quad \text{for all } i = 1, \ldots, m.$$

Here and throughout we denote by $\|\cdot\|$ the l_2-norm.

Theoretically speaking, if P is not empty then by a perturbation we can get an equivalent system which satisfies $\text{int}(P) \neq \emptyset$. Thus (A1) is not that restrictive.

In the following we show that, under (A1), the solution set of (P_{f+h}) is nonempty and $x_c(d^*)$ is an interior point of P and unique for any solution d^* of (P_{f+h}). The feasibility problem for the system (FP) is thus converted into an unconstrained optimization problem.

First we prove a simple lemma.

LEMMA 2.6. *If* $\text{int}(P) \neq \emptyset$, *then* $f(d) > 0$ *for all* $d \in \mathcal{D}$.

Proof. For $d \in \mathcal{D}$, $E(d)$ is an ellipsoid that contains P. If $\text{int}(P)$ is not empty, $E(d)$ is a full-dimensional ellipsoid; thus the volume of $E(d)$ is positive, i.e., $v(d) > 0$. Therefore $f(d)h(d) > 0$, and so $f(d) > 0$, since $h(d) > 0$. □

THEOREM 2.7. *Suppose* (A1) *holds. Then* (P_{f+h}) *is a convex programming problem, and the solution set, denoted* $\mathcal{S}_{f+h}$, *is not empty; moreover,* $x_c(d^*)$ *is an interior point of* P *for any optimal solution* d^* *of* (P_{f+h}).

Proof. From the previous propositions, (P_{f+h}) is a convex program. We now prove its solution set is not empty. Define

$$\tilde{F}(d) := \begin{cases} f(d) + h(d) & \text{if } d \in \mathcal{D}, \\ +\infty & \text{otherwise.} \end{cases}$$

Since $\tilde{F}$ is not identically $+\infty$ and $\tilde{F}(d) > -\infty$ for every d, $\tilde{F}$ is a proper convex function.

On the other hand, since $\tilde{F}(d)$ is continuous on $\text{dom}(\tilde{F}) = \mathcal{D}$, if d_k lies in the relative interior of $\mathcal{D}$ and $d_k \to d$, then, by Lemma 2.6,

$$\tilde{F}(d_k) \longrightarrow \left\{ \begin{array}{ll} F(d) & \text{if } d \in \mathcal{D}, \\ +\infty & \text{if } d \notin \mathcal{D}, \end{array} \right\} = \tilde{F}(d).$$

So $\tilde{F}$ is closed (since $\text{cl}\tilde{F} = \tilde{F}$).

In the following we prove that $\tilde{F}$ has no (nonzero) direction of recession in R^m. Suppose we are given a direction, say y. We show that y cannot be a recession direction of $\tilde{F}$. Since $\tilde{F}(d) = \infty$ for $d \notin R_+^m$ where $R_+^m := \{d \in R^m : d \geq 0\}$, no $y \notin R_+^m$ can be a recession direction for $\tilde{F}$. Therefore we need only consider $y \in R_+^m$.

Suppose we are given $y \in R_+^m$ and $y \neq 0$. Without loss of generality we assume $y_1 \neq 0$. Since A is of full rank and $a_1 \neq 0$, there is a set of n columns of A including a_1 which is linearly independent; further, by Cauchy's formula,

$$h(e + \lambda y) \leq \alpha^{-\frac{2}{n}}(1 + \lambda y_1)^{-\frac{1}{n}},$$

where α is the determinant of the matrix formed by the n independent columns. Thus $h(e + \lambda y) \longrightarrow 0$. On the other hand, since $\text{int}(P)$ is not empty and $P \subset E$, there is a $\delta > 0$ such that $f(e + \lambda y)h(e + \lambda y) > \delta > 0$. Thus

$$f(e + \lambda y) \longrightarrow \infty \text{ as } \lambda \longrightarrow \infty,$$

which shows f, thus $\tilde{F}$, has no recession direction in R_+^m.

We have thus proved that $\tilde{F}$ has no recession direction in R^m. Therefore, by Theorem 27.3 of Rockafellar [20], $\tilde{F}$ attains its minimum in R^m. Equivalently, F attains its minimum over $\mathcal{D}$, i.e., $\mathcal{S}_{f+h}$ is not empty.

On the other hand, by the Karush–Kuhn–Tucker (KKT) conditions, d^* is optimal if and only if there are $\lambda^T = (\lambda_1, \ldots, \lambda_m) \geq 0$ such that $\nabla F(d^*) = \lambda$, i.e.,

$$-(a_i^T x_c(d^*) - l_i)(a_i^T x_c(d^*) - u_i) = \frac{1}{n} \det(ADA^T)^{-\frac{1}{n}} a_i^T (ADA^T)^{-1} a_i + \lambda_i \tag{9}$$

for all $i = 1, \ldots, m$. The right-hand side of (9) is positive; hence $x_c(d^*)$ is an interior point. □

The following corollaries show that, under assumption (A1), (P_{f+h}) and (P_{fh}) are essentially the same.

COROLLARY 2.8. *Under* (A1), *the solution set of*

$$(\text{P}_{fh}) \qquad \begin{array}{ll} \min & v(d) := f(d) \cdot h(d) \\ \text{s.t.} & d \in \mathcal{D} \end{array}$$

is not empty.

Proof. Suppose d^* is an optimal solution to (P_{f+h}). We show that d^* is an optimal solution to (P_{fh}). Otherwise, there exists $\bar{d}$ such that $v(d^*) > v(\bar{d})$. Since $f(d) > 0$ for $d \in \mathcal{D}$, we can let $\lambda = \sqrt{h(\bar{d})/f(\bar{d})}$ and have

$$2\sqrt{v(\bar{d})} = 2\sqrt{v(\lambda\bar{d})} = f(\lambda\bar{d}) + h(\lambda\bar{d}),$$

which leads to

$$f(d^*) + h(d^*) \geq 2\sqrt{v(d^*)} > 2\sqrt{v(\bar{d})} = 2\sqrt{v(\lambda\bar{d})} = f(\lambda\bar{d}) + h(\lambda\bar{d}),$$

a contradiction. □

As a matter of fact, if we let $\mathcal{S}_{fh}$ and $\mathcal{S}_{f+h}$ be the solution sets of (P_{fh}) and (P_{f+h}), respectively, we have the following result.

COROLLARY 2.9. *Under assumption* (A1), $\mathcal{S}_{f+h} \subset \mathcal{S}_{fh}$ *and* $\mathcal{S}_{fh} = \text{cone}(\mathcal{S}_{f+h}) \setminus \{0\}$.

The following example shows that the Hessian of F might be only positive semidefinite, and the optimal solution set $\mathcal{S}_{f+h}$ can be a segment.

Example. Let

$$A = \begin{pmatrix} 1 & 0 & 1 & -2 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

and

$$l = \left(-\sqrt{\frac{1}{2}}, -\sqrt{\frac{1}{2}}, -1, -\sqrt{\frac{5}{2}}\right)^T, \quad u = \left(\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}, 1, \sqrt{\frac{5}{2}}\right)^T.$$

Then, if we take $d^0 = (\frac{1}{4}, \frac{5}{8}, \frac{1}{4}, \frac{1}{8})^T$, $\nabla^2 F(d^0)$ is singular. Indeed, $r = 0$ so $x_c(d) = 0$ for all d, whence $H_f(d) = 0$. Also, it is easy to see that $(6, 3, -2, -1)^T$ lies in the null space of H_h (see (7)). Moreover, let

$$d = \left(0, \frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)^T, \quad \bar{d} = (1, 1, 0, 0)^T, \quad \text{and} \quad d_\lambda = \lambda d + (1-\lambda)\bar{d}, \quad \lambda \in [0, 1].$$

Then

$$\begin{aligned} F(d_\lambda) &= f(d_\lambda) + h(d_\lambda) \\ &= r^T D_\lambda A^T (A D_\lambda A^T)^{-1} A D_\lambda r - l^T D_\lambda u + \det(A D_\lambda A^T)^{-\frac{1}{n}} \\ &\equiv 2 \text{ (note that } r = 0,\ -l^T D_\lambda u = 1,\ A D_\lambda A^T = I), \end{aligned}$$

and $\nabla F(d_\lambda) = 0$ (see Proposition 2.3). Therefore, the solution set of (P_{f+h}) is (at least) $\{d_\lambda = \lambda d + (1-\lambda)\bar{d} : \lambda \in [0, 1]\}$.

We now show that $x_c(d^*)$ is unique even though the optimal solution set $\mathcal{S}_{f+h}$ may not be a singleton. We can thus define $x_c(d^*)$ as a (weighted) center. We first require several lemmas.

LEMMA 2.10. *For all* $d, \bar{d} \in \mathcal{S}_{f+h}$,

$$f(d) = f(\bar{d}) = h(d) = h(\bar{d}). \tag{10}$$

Proof. Since $\mathcal{S}_{f+h}$ is a convex set, $F(d) = F(d')\ \forall d, d' \in \mathcal{S}_{f+h}$; on the other hand, because f and h are nonnegative functions which are homogeneous with degrees 1 and -1, respectively, for $d \in \mathcal{S}_{f+h}$, $f(d) = h(d) = \frac{1}{2}F(d)$. The lemma follows immediately. □

LEMMA 2.11. *If* $\det(I + \varepsilon\Lambda) \equiv 1$ *for all* $\varepsilon \in (0, \eta)$, *where* η *is some positive number, and* $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, *then* $\Lambda = 0$.

Proof. We have $\det(I + \varepsilon\Lambda) = \prod_{i=1}^n (1 + \varepsilon\lambda_i) \equiv 1\ \forall \varepsilon \in (0, \eta)$. Thus the coefficient of ε^n should be zero, i.e., $\prod_{i=1}^n \lambda_i = 0$; so, at least one of the λ_i's is zero, say $\lambda_n = 0$. Thus we have

$$\det(I + \varepsilon\Lambda) = \prod_{i=1}^{n-1} (1 + \varepsilon\lambda_i).$$

The same argument implies that at least one of $\{\lambda_1, \ldots, \lambda_{n-1}\}$ is zero; we keep using this argument until eventually we have $\Lambda = 0$. □

PROPOSITION 2.12. *For all* $d, \bar{d} \in \mathcal{S}_{f+h}$, $A D A^T = A \bar{D} A^T$.

Proof. By Lemma 2.10,

$$\det(A(D + \varepsilon(\bar{D} - D))A^T) = \det(ADA^T) \quad \forall \varepsilon \in [0, 1]. \tag{11}$$

On the other hand,

$$\begin{aligned} \det(A(D + \varepsilon(\bar{D} - D))A^T) &= \det(ADA^T + \varepsilon A(\bar{D} - D)A^T) \\ &= \det(ADA^T)^{\frac{1}{2}} \det(I + \varepsilon(ADA^T)^{-\frac{1}{2}} A(\bar{D} - D)A^T (ADA^T)^{-\frac{1}{2}}) \det(ADA^T)^{\frac{1}{2}}. \end{aligned}$$

Thus, (11) implies $\det(I+\varepsilon(ADA^T)^{-\frac{1}{2}}A(\bar{D}-D)A^T(ADA^T)^{-\frac{1}{2}}) = 1$. By Lemma 2.11, the eigenvalues of $(ADA^T)^{-\frac{1}{2}}A(\bar{D}-D)A^T(ADA^T)^{-\frac{1}{2}}$ are all zeros, so

$$(ADA^T)^{-\frac{1}{2}}A(\bar{D}-D)A^T(ADA^T)^{-\frac{1}{2}} = 0,$$

i.e., $A(\bar{D}-D)A^T = 0$. Hence $A\bar{D}A^T = ADA^T$. □

Thus the minimum volume ellipsoids containing P all have the same shape but possibly different locations. We now show that the location is also unique.

PROPOSITION 2.13. *For all* $d, \bar{d} \in \mathcal{S}_{f+h}$, $ADr = A\bar{D}r$.

Proof. From Lemma 2.10 and Proposition 2.12,

$$\begin{aligned} &r^T(D+\varepsilon(\bar{D}-D))A^T(ADA^T)^{-1}A(D+\varepsilon(\bar{D}-D))r - l^T(D+\varepsilon(\bar{D}-D))u \\ &\quad = r^TDA^T(ADA^T)^{-1}ADr - l^TDU \ \forall \varepsilon \in [0,1]. \end{aligned}$$

Thus by letting the coefficient of ε^2 be zero,

$$r^T(\bar{D}-D)A^T(ADA^T)^{-1}A(\bar{D}-D)r = 0.$$

Since $(ADA^T)^{-1}$ is positive definite, $A(\bar{D}-D)r = 0$, i.e., $A\bar{D}r = ADr$. □

THEOREM 2.14 (uniqueness). *If* d^* *solves* (P_{f+h}), $x_c(d^*)$ *and* $E(d^*)$ *are unique.*

Proof. By Propositions 2.12 and 2.13, for any $d \in \mathcal{S}_{f+h}$, $x_c^* = (ADA^T)^{-1} \cdot ADr$ is a constant vector. □

Now we show that f can be used to determine whether $P = \emptyset$. This result provides us a tool that is useful later in the analysis of the output of our model. Recall from Lemma 2.6 that $f(d) \leq 0$ implies that the interior of P is empty. To complement this, we have the following result; the hard part of the proof can be skimmed or skipped at a first reading.

THEOREM 2.15. *If* $l < u$ *and* A *is of full rank, then*

$$P = \emptyset \Longleftrightarrow f(d) < 0 \text{ for some } d \in \mathcal{D}.$$

Proof. "$\Longleftarrow$". If there is a d such that $f(d) < 0$ and $P \neq \emptyset$, then, by taking $\bar{l}_i = l_i - \delta, \bar{u}_i = u_i + \delta$ and noting that f is a continuous function of l and u for any given d, we can choose small δ so that for the corresponding system we have $\mathrm{int}(P) \neq \emptyset$ and $f(d) < 0$, which contradicts Lemma 2.6.

"$\Longrightarrow$". First of all, we recall Helly's theorem which can be found, e.g., in Rockafellar [20]:

> Let $\{C_i : i \in \mathcal{I}\}$ be a finite collection of convex sets in R^n. If every subcollection consisting of $n+1$ or fewer sets has a nonempty intersection, then the entire collection has a nonempty intersection.

If $P = \emptyset$, by Helly's theorem there are $n+1$ constraints which are inconsistent. If the rank of the set of the corresponding a_i's is $k < n$, then, since Helly's theorem is invariant under affine transformation, we can find $k+1$ constraints which are inconsistent too. Moreover, since we assume that A is of full rank, we can expand these $k+1$ ones to $n+1$ constraints which are inconsistent and the corresponding a_i's are of rank n.

Therefore, if $P = \emptyset$, there are $n+1$ constraints, say the first $n+1$ ones, which are inconsistent and the corresponding a_i's are of rank n. Without loss of generality (by making a nonsingular affine transformation of the space), we assume that

$$\bar{A} = [I, a], \quad l_i = -1, u_i = 1 \ \forall i = 1, \ldots, n, \quad \text{and } \|a\|_1 = 1,$$

and the corresponding system

$$\begin{pmatrix} -e \\ l_{n+1} \end{pmatrix} \leq \begin{pmatrix} I \\ a^T \end{pmatrix} x \leq \begin{pmatrix} e \\ u_{n+1} \end{pmatrix} \tag{12}$$

is inconsistent. We claim that there exists a $d \in \{d \in \mathcal{D} : d_i = 0 \ \forall i = n+2, \ldots, m\}$ for which $f(d) < 0$. Equivalently, we show that the claim is true for system (12).

We denote $s := \frac{u-l}{2}$ for this reduced system as before. Since system (12) has no solution, if we let $\mathcal{X} := \{x \in R^n : -1 \leq x_j \leq 1, j = 1, \ldots, n\}$, then either $a^T x < l_{n+1} \ \forall x \in \mathcal{X}$, or $a^T x > u_{n+1} \ \forall x \in \mathcal{X}$. We assume the first case. Note that $a^T x$ is maximized over $\mathcal{X}$ by some vertex of $\mathcal{X}$, which gives its optimal value as $\sum_{j=1}^n |a_{[j]}| = \|a\|_1 = 1$, where $a = (a_{[1]}, \ldots, a_{[n]})^T$. Therefore, $l_{n+1} > 1$.

We now show how to define d so that $f(d) < 0$. Confirming this inequality requires some straightforward but messy manipulations; for details see [13, 15].

(i) If $a_{[j]} \neq 0 \ \forall j = 1, \ldots, n$, take $d_j = |a_{[j]}|$ for $j = 1, \ldots, n$ and

$$d_{n+1} = \frac{l_{n+1}u_{n+1} - \sum_{j=1}^n d_j \sum_{j=1}^n a_{[j]}^2/d_j}{2s_{n+1}^2 \sum_{j=1}^n a_{[j]}^2/d_j}.$$

(ii) If some $a_{[j]}$'s are zeros, then we take

$$d_j = \begin{cases} |a_{[j]}| - \varepsilon & \text{if } a_{[j]} \neq 0, \\ \frac{n'}{n-n'}\varepsilon & \text{if } a_{[j]} = 0, \end{cases}$$

where n' is the number of $a_{[j]}$'s with $a_{[j]} \neq 0$ and d_{n+1} as above. By letting ε be sufficiently small, we get d so that $f(d) < 0$. □

2.2. An algorithm for (P_{f+h}). We now consider how to use model (P_{f+h}) to solve the original feasibility problem and how to obtain the center $x_c^* := x_c(d^*)$ (by the uniqueness theorem, this is well defined).

Here is a coordinate descent algorithm.

ALGORITHM 2.1.

- Initialization. Choose $d^0 > 0$, and scale it so that $f(d^0) = h(d^0)$; set $k = 0$.
- For $k = 0, 1, \ldots$, do
  If $x_c^k := x_c(d^k) \in P$, stop with the feasible solution x_c^k; otherwise, choose j with the jth constraint violated by x_c^k. Let

$$\beta := \frac{s_j^2}{f(d^k) a_j^T (ADA^T)^{-1} a_j}, \quad \gamma := a_j^T (ADA^T)^{-1} a_j;$$

and

$$(l_0, u_0) = \begin{cases} (l_j, l_j + (f(d^k)\gamma)^{\frac{1}{2}}) & \text{if } a_j^T x_c^k < l_j, \\ (u_j - (f(d^k)\gamma)^{\frac{1}{2}}, u_j) & \text{if } a_j^T x_c^k > u_j. \end{cases}$$

If $l_0 > u_j$ or $u_0 < l_j$, stop with the conclusion that the original system is not consistent.

(i) (canonical case) If $\beta \le \frac{1}{4}$, we take $\lambda^k = \frac{1}{n\gamma}$, and $d^{k+1} = d^k + \lambda^k e_j$.

(ii) If $\beta > \frac{1}{4}$, we take $\lambda^k = \frac{1}{n\gamma}$, and $d^{k+1} = d^k + \lambda^k e_j$, AND update the bounds as follows:

$$l_j \leftarrow \frac{d_j^k l_j + \tilde{d}_0^{k+1} l_0}{d_j^k + \tilde{d}_0^{k+1}} \quad \text{if } a_j^T x_c^k > u_j, \text{ or}$$

$$u_j \leftarrow \frac{d_j^k u_j + \tilde{d}_0^{k+1} u_0}{d_j^k + \tilde{d}_0^{k+1}} \quad \text{if } a_j^T x_c^k < l_j,$$

where $\tilde{d}_0^{k+1} = \frac{1}{n\gamma}$.

Then, scale d^{k+1} so that $f(d^{k+1}) = h(d^{k+1})$, where f is computed with the updated jth bounds; set $k \leftarrow k+1$, and repeat.

Let us explain the noncanonical case briefly. Without loss of generality, $a_j^T x_c^k < l_j$, and $\beta > \frac{1}{4}$ implies that $u_j > l_j + (f(d^k)\gamma)^{\frac{1}{2}} =: u_0 > a_j^T x_c^k + (f(d^k)\gamma)^{\frac{1}{2}}$ so that the hyperplanes $\{x : a_j^T x = u_j\}$ and $\{x : a_j^T x = u_0\}$ don't intersect the current ellipsoid. It follows that the upper bounds u_j and u_0 on $a_j^T x$ are implied by the remaining bounds and that the polytope P is unchanged if u_j is replaced by u_0 or by any convex combination of these. Case (ii) above corresponds to the following procedure: we add a new constraint $l_0 \le a_0^T x \le u_0$, where $a_0 := a_j$, with current weight $d_0^k = 0$, which doesn't change the current ellipsoid. The current center violates the lower bound of this new pair of constraints, for which the value of β is exactly $\frac{1}{4}$. Thus we perform the canonical update with this constraint so that d_0 becomes $\frac{1}{n\gamma}$. It turns out that the resulting ellipsoid can also be described with just the old constraints if we update the weights *and* u_j as described above; the two terms of the defining inequality corresponding to indices 0 and j are thus combined into one. The difference from the Liao and Todd algorithm [14] is that here the objective function is $F(d) := f(d) + h(d)$ instead of $v(d) := f(d) \cdot h(d)$ as used there.

THEOREM 2.16. *Suppose that* $\{d^k\}$ *is the sequence generated by the algorithm. Then* $\frac{F(d^{k+1})}{F(d^k)} \le 1 - \frac{1}{16n^2}$, *as long as* $x_c(d^k) \notin P$.

Proof. Let $\tilde{d}^{k+1}$ be $d^k + \lambda^k e_j$ (the next iterate before scaling). Also let $\theta = \lambda\gamma$ and $\theta_0 := \frac{(a_j^T x_c - l_j)(a_j^T x_c - u_j)}{s_j^2} \ge 0$. Then the same analysis as in Liao and Todd [14] leads to

$$\begin{aligned}
\frac{F(\tilde{d}^{k+1})}{F(d^k)} &= 1 + \frac{1}{F(d^k)} \left[f(d^k)\beta \frac{\theta}{1+\theta}(\theta - \theta_0) - (1 - (1+\theta)^{-\frac{1}{n}}) h(d^k) \right] \\
&= 1 - \frac{f(d^k)}{f(d^k) + h(d^k)} \left[1 - (1+\theta)^{-\frac{1}{n}} - \beta \frac{\theta}{1+\theta}(\theta - \theta_0) \right] \\
&= 1 - \frac{1}{2} \left[1 - (1+\theta)^{-\frac{1}{n}} - \beta \frac{\theta}{1+\theta}(\theta - \theta_0) \right].
\end{aligned}$$

Let $R(\theta) := 1 - (1+\theta)^{-\frac{1}{n}} - \beta \frac{\theta}{1+\theta}(\theta - \theta_0)$. As in [14], it is enough to deal with the canonical case, i.e., $\beta \le \frac{1}{4}$. Since $\theta_0 > 0$,

$$\begin{aligned}
R(\theta) &\ge 1 - (1+\theta)^{-\frac{1}{n}} - \beta \frac{\theta^2}{1+\theta} \\
&\ge 1 - (1+\theta)^{-\frac{1}{n}} - \frac{1}{4} \cdot \frac{\theta^2}{1+\theta} =: r(\theta).
\end{aligned}$$

The following inequality can be obtained from Taylor's theorem:

$$\left|(1+\theta)^{\frac{1}{n}} - \left(1+\frac{\theta}{n}\right)\right| \leq \frac{n-1}{2n^2}\theta^2 \text{ for } 0 < \theta \leq 1.$$

Hence, noting that $\theta > 0$,

$$\left|(1+\theta)^{-\frac{1}{n}} - \left(1+\frac{\theta}{n}\right)^{-1}\right| \leq \frac{n-1}{2n^2}\theta^2.$$

Therefore, for $0 < \theta \leq 1$,

$$\begin{aligned} r(\theta) &\geq 1 - \left(1+\frac{\theta}{n}\right)^{-1} - \frac{1}{4}\cdot\frac{\theta^2}{1+\theta} - \frac{n-1}{2n^2}\theta^2 \\ &= \frac{\theta}{n+\theta} - \frac{1}{4}\cdot\frac{\theta^2}{1+\theta} - \frac{n-1}{2n^2}\theta^2 \\ &= \theta\left[\frac{4-n\theta+4\theta-\theta^2}{4(n+\theta)(1+\theta)} - \frac{n-1}{2n^2}\theta\right]. \end{aligned}$$

If we take $\theta = \frac{1}{n}$, we have

$$\begin{aligned} r(\theta) &\geq \frac{1}{n}\left[\frac{3}{4(1+\theta)(n+1)} - \frac{n-1}{2n^3}\right] \geq \frac{1}{n}\left[\frac{3}{8(n+1)} - \frac{n-1}{2n^3}\right] \\ &\geq \frac{1}{n}\left[\frac{6n^3-8n^2+8}{16n^3(n+1)}\right] \geq \frac{1}{n}\cdot\frac{1}{8n} = \frac{1}{8n^2}. \end{aligned}$$

So

$$\frac{F(\tilde{d}^{k+1})}{F(d^k)} = 1 - \frac{1}{2}R(\theta) \leq 1 - \frac{1}{2}r(\theta) \leq 1 - \frac{1}{16n^2}. \qquad \square$$

In terms of the shrinkage of the volumes of the corresponding ellipsoids, we have

$$\frac{\text{volume}(E^+)}{\text{volume}(E)} \leq \left(1 - \frac{1}{16n^2}\right)^n \leq \exp\left(-\frac{1}{16n}\right).$$

Although, from the above results, the coordinate descent algorithm is polynomial, it converges slowly. In the next section we will propose another model where Newton's method can be employed completely. We also note that a partial Newton step algorithm is proposed in [13].

When we use Model I to solve the feasibility problem (1), there are three possible outcomes according to the values of $F_{\text{inf}} := \inf\{F(d) : d \in \mathcal{D}\}$.

THEOREM 2.17.

(i) *If $F_{\text{inf}} > 0$, then there is a $d^* \in \mathcal{D}$ such that $F(d^*) = F_{\text{inf}}$ and $x_c(d^*)$ is feasible.*

(ii) *If $F_{\text{inf}} < 0$, then $P = \emptyset$.*

(iii) *If $F_{\text{inf}} = 0$, then $\text{int}(P) = \emptyset$ but $P \neq \emptyset$.*

Proof. It is easy to see that if we replace (A1) by $F_{\text{inf}} > 0$ then f has no direction of recession and so the conclusions of Theorem 2.7 still hold; (i) thus follows. If $F_{\text{inf}} < 0$, then there is a $d \in \mathcal{D}$ such that $F(d) < 0$ and therefore $f(d) < 0$. Hence, by

Theorem 2.15, $P = \emptyset$, which shows (ii). Outcome (iii) follows from Theorem 2.7 and Theorem 2.15. □

In case (iii), if there is a $d^* \in \mathcal{D}$ at which $f(d^*) = 0$, then $x_c(d^*)$ is feasible; however, in general, $x_c(d^k)$ does not approach the feasible set even if d^k is such that $F(d^k) \longrightarrow 0$ as shown in the following example.

Example. Let

$$\begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} x \leq \begin{pmatrix} 2 \\ 2 \\ 4 \\ 30 \end{pmatrix}.$$

By calculation, we have

$$x_c = \left(\frac{d_1 + 3d_3}{d_1 + d_3}, \frac{d_2 + 15d_4}{d_2 + d_4} \right),$$

$$f(d) = \frac{(d_1 + 3d_3)^2}{d_1 + d_3} + \frac{(d_2 + 15d_4)^2}{d_2 + d_4} - 8d_3,$$

$$h(d) = \frac{1}{\sqrt{(d_1 + d_3)(d_2 + d_4)}}.$$

If we take $d_1^k = d_3^k = \frac{1}{2}$ and $d_2^k = d_4^k = \frac{1}{k}$, then $v(d^k) \longrightarrow 0$ as $k \longrightarrow \infty$, but $x_c(d^k) \longrightarrow (2, 8)^T$ which is not feasible. By scaling, this sequence of d^k can be converted to $\bar{d}^k$ such that $F(\bar{d}^k) \longrightarrow 0$ as $k \longrightarrow \infty$ and $x_c(\bar{d}^k) = x_c(d^k) \longrightarrow (2, 8)^T$.

3. Model II and the associated center. Model I provides us with a kind of center for P, but this center is not so easy to obtain practically. In this section we propose a simpler model whose associated center can be obtained more efficiently.

By examining the previous model, we find that it is based on the homogeneity properties of f and h. Since f provides a tool for determining the feasibility of our problem we want to keep it, but the role of function h is really like a barrier: it balances the weights $d_i, i = 1, \ldots, m$, and forces the corresponding $x(d)$ to approach the center x_c^*. We will use a much simpler function to replace h. The new model is more efficient in practice.

As a matter of fact, any convex homogeneous function B of degree -1 defined on $\mathcal{D}$ such that

(i) $f + B$ has an optimal solution in $\mathcal{D}$,

(ii) $\nabla B(d) \leq 0$ for any $d \in \mathcal{D}$

can play the same role as h does. (The self-concordant barrier function $-\sum \ln d_i$ familiar in interior-point methodology fails to be homogeneous; the related multiplicative barrier $(\prod d_i)^{-1/m}$ is a possibility, but seems to be more complicated than the simple function we propose below and does not allow the convergence analysis below.)

One such function, which we choose for our new model, is as follows:

$$B(d) := e^T d^{-1} := \sum_{i=1}^{m} \frac{1}{d_i}.$$

We let $B(d) = e^T d^{-1} = +\infty$ if any $d_i = 0$.

The corresponding model, defined as Model II, is

$$(\mathrm{P}_{f+B}) \qquad \begin{array}{l} \min\ G(d) := f(d) + B(d) \\ \text{s.t.}\ \ d \in \mathcal{D}. \end{array}$$

We note that (P_{f+B}) is equivalent to minimizing $G(d)$ over $R^m_{++} := \{d \in R^m : d > 0\}$.

3.1. Properties of (P_{f+B}). Obviously, $B(d)$ is a strictly convex function, so $G(d)$ is strictly convex too. Accordingly, we have the following theorem.

THEOREM 3.1. *If* (A1) *holds, then* $G(d)$ *is a closed proper strictly convex function and* $G(d)$ *has no direction of recession. Thus there is an unique solution to* (P_{f+B}); *moreover, the level set*

$$\mathcal{D}_e := \{d \in \mathcal{D} : G(d) \le G(e)\}$$

is a closed bounded convex set.

Proof. The proof of Theorem 3.1 is similar to that of Theorem 2.7. We just note that, under assumption (A1), $f(d)$ has no recession direction in R^m_+; hence, $G(d)$ has no recession direction either. □

It is easy to prove that if d^* is the unique solution to (P_{f+B}), the associated $x_c(d^*)$ is an interior point of P. We thus define $x_c(d^*)$ as the *center* associated with Model II. In the following we show that (P_{f+B}) is equivalent to finding a minimum volume ellipsoid among those with a particular shape, and $x_c(d^*)$ is the center of this ellipsoid.

First we note that (P_{f+B}) is equivalent to

$$(P_{fB}) \qquad \begin{array}{l} \min\ v^{\mathrm{II}}(d) := f(d) \cdot B(d) \\ \text{s.t.}\ \ d \in \mathcal{D} \end{array}$$

in the following sense: if d^* is the solution to (P_{f+B}), it is also a solution to (P_{fB}); and if d^* is a solution to (P_{fB}), then $\sqrt{B(d^*)/f(d^*)}d^*$ is the unique solution to (P_{f+B}).

LEMMA 3.2. *Suppose that* $d_i \ge 0, i = 1, \ldots, m$. *Then*

$$\sum_{i=1}^{m} d_i w_i^2 \le 1 \tag{13}$$

implies

$$\sum_{i=1}^{m} w_i^2 \le e^T d^{-1}. \tag{14}$$

Proof. If there is i_0 such that $d_{i_0} = 0$, then (14) is always true, since we have set $e^T d^{-1}$ to ∞ in this case; otherwise we suppose that $d_{i_0} = \min\{d_i : i = 1, \ldots, m\}$. Then $d_{i_0} \sum_{i=1}^m w_i^2 \le \sum_{i=1}^m d_i w_i^2 \le 1$, which leads to $\sum_{i=1}^m w_i^2 \le \frac{1}{d_{i_0}} \le e^T d^{-1}$. □

We define

$$E^{\mathrm{II}}(d) := \left\{ x \in R^n : \frac{1}{e^T d^{-1}} (x - x_c)^T A A^T (x - x_c) \le f(d) \right\} \tag{15}$$

where $x_c = x_c(d) = (ADA^T)^{-1} ADr$ and $f(d) = x_c^T ADA^T x_c - l^T Du$. Then (P_{fB}) is equivalent to minimizing the volume of the ellipsoid $E^{\mathrm{II}}(d)$.

The relationship between this ellipsoid and the one defined by (4) is described in the next proposition.

PROPOSITION 3.3. *If* $E(d)$ *is the ellipsoid defined by* (4), *then* $E(d) \subset E^{\mathrm{II}}(d)$.

Proof.

$$E(d) = \{x \in R^n : (x - x_c)^T ADA^T (x - x_c) \le f(d)\}$$
$$= \{x \in R^n : (A^T(x - x_c))^T D(A^T(x - x_c)) \le f(d)\}.$$

Let $w := A^T(x - x_c)$. Then, by Lemma 3.2,

$$E(d) = \{x \in R^n : w^T Dw \le f(d)\} = \left\{x \in R^n : \sum_{i=1}^m d_i w_i^2 \le f(d)\right\}$$
$$\subseteq \left\{x \in R^n : \sum_{i=1}^m w_i^2 \le f(d) e^T d^{-1}\right\}$$
$$= \left\{x \in R^n : \frac{1}{e^T d^{-1}} (x - x_c)^T AA^T (x - x_c) \le f(d)\right\} = E^{\mathrm{II}}(d). \qquad \square$$

Since $P \subset E(d)$, the above result also shows that $P \subset E^{\mathrm{II}}(d)$.

We note that since ADA^T can be nonsingular if one component of d is zero, the volume of $E(d)$ may be finite, but for such a d the volume of $E^{\mathrm{II}}(d)$ is infinite; hence E^{II} and E are not equivalent, i.e., there is no constant $\lambda > 0$ such that

$$E(d) \subset E^{\mathrm{II}}(d) \subset \lambda E(d) \text{ for all } d \in \mathcal{D}.$$

3.2. Algorithms for $(\mathbf{P}_{f+B})$. We describe first a coordinate descent algorithm to solve the feasibility problem (P) by using Model II.

ALGORITHM 3.1.

- Initialization. Normalize the constraints so that $\|a_i\| = 1 \ \forall i = 1, \ldots, m$. Choose $d^0 > 0$ and scale it so that $f(d^0) = B(d^0)$; set $k = 0$.
- For $k = 0, 1, \ldots$, do
  If $x_c^k := x_c(d^k) \in P$, stop with the feasible solution x_c^k; otherwise, choose j with the jth constraint violated by x_c^k. Then take $\lambda^k = \arg\min\{G(d^k + \lambda e_j) : \lambda \ge 0\}$, and $d^{k+1} = d^k + \lambda^k e_j$; scale d^{k+1} so that $f(d^{k+1}) = B(d^{k+1})$; set $k \leftarrow k + 1$, and repeat.

To analyze this algorithm we first need the following result.

LEMMA 3.4. *Suppose that the feasible region P of (1) is "fat" so that there is a ball, say $\mathcal{B}$, of radius δ_0 contained in P, i.e., there is some $\hat{x}$ such that $\mathcal{B} := \mathcal{B}(\hat{x}; \delta_0) \subset P$, and suppose that $\|a_i\| = 1$ for all $i = 1, \ldots, m$. Then, for all $i = 1, \ldots, m$, and $d \in \mathcal{D}_+$ with $f(d) = B(d)$ and $G(d) \le G(e)$,*

$$\kappa_i(d) := \frac{f(d)}{a_i^T (ADA^T)^{-1} a_i} \le \frac{(G_e)^2}{4\delta_0^2},$$
$$\beta_i(d) := \frac{s_i^2}{f(d) a_i^T (ADA^T)^{-1} a_i} \le \frac{s_{\max}^2}{\delta_0^2},$$

where

$$G_e := G(e) = r^T A^T (AA^T)^{-1} Ar - l^T u + m,$$
$$s_{\max} := \max\{s_i : i = 1, \ldots, m\}.$$

Proof. Since $2\sqrt{f(d)a_i^T(ADA^T)^{-1}a_i}$ is the width of the ellipsoid $E(d)$ in the direction a_i, we have $\sqrt{f(d)a_i^T(ADA^T)^{-1}a_i} \geq \delta_0$. Thus

$$\kappa_i(d) = \frac{f(d)}{a_i^T(ADA^T)^{-1}a_i} = \frac{f(d)^2}{f(d)a_i^T(ADA^T)^{-1}a_i}$$
$$= \frac{G(d)^2}{4f(d)a_i^T(ADA^T)^{-1}a_i} \leq \frac{(G(e))^2}{4f(d)a_i^T(ADA^T)^{-1}a_i} \leq \frac{(G_e)^2}{4\delta_0^2}.$$

On the other hand,

$$\beta_i(d) = \frac{s_i^2}{f(d)a_i^T(ADA^T)^{-1}a_i} \leq \frac{s_{\max}^2}{\delta_0^2}. \qquad \square$$

THEOREM 3.5. *Assume the hypotheses of Lemma* 3.4 *and let* $\omega := \max\{\frac{(G_e)^2}{4\delta_0^2}, \frac{s_{\max}^2}{\delta_0^2}, 1\}$. *Then as long as* $x_c(d^k) \notin P$, *we have*

$$\frac{G(d^{k+1})}{G(d^k)} \leq 1 - \frac{1}{16n^6\omega^5}.$$

Proof. For simplicity we write d for d^k, λ for λ^k, and d^+ for $d + \lambda^k e_j$. Then

$$B(d^+) = B(d) - \frac{\lambda}{d_j(d_j + \lambda)},$$

and, from [14],

$$f(d^+) = f(d) + f(d)\frac{\beta_j\theta}{1+\theta}(\theta - \theta_0) \leq f(d)\left(1 + \frac{\beta_j\theta^2}{1+\theta}\right),$$

where $\beta_j = \beta_j(d)$, $\theta = \lambda\gamma_j$, $\gamma_j = a_j^T(ADA^T)^{-1}a_j$, and $\theta_0 := \frac{(a_j^Tx_c - l_j)(a_j^Tx_c - u_j)}{s_j^2} \geq 0$. Also,

$$\sum d_i a_i^T(ADA^T)^{-1}a_i = \operatorname{tr}(DA^T(ADA^T)^{-1}A)$$
$$= \operatorname{tr}(ADA^T(ADA^T)^{-1}) = n,$$

so $d_j\gamma_j \leq n$. Thus, noting that $f(d) = B(d)$,

$$\frac{G(d^+)}{G(d)} \leq 1 - \frac{1}{2}\left[\frac{\lambda}{f(d)d_j(d_j+\lambda)} - \beta_j\frac{\theta^2}{1+\theta}\right]$$
$$\leq 1 - \frac{1}{2}\left[\frac{\theta}{nf(d)(d_j+\lambda)} - \beta_j\frac{\theta^2}{1+\theta}\right]$$
$$\leq 1 - \frac{1}{2}\left[\frac{\theta}{n\kappa_j(d)(n+\theta)} - \beta_j\frac{\theta^2}{1+\theta}\right]$$
$$\leq 1 - \frac{1}{2}\left[\frac{\theta}{n\omega(n+\theta)} - \omega\frac{\theta^2}{1+\theta}\right].$$

If we take $\theta = \frac{1}{\omega^2 n^2}$, then

$$\frac{G(d^+)}{G(d)} \leq 1 - \frac{1}{16n^6\omega^5}. \qquad \square$$

In terms of the shrinkage of the volumes of the corresponding ellipsoids we have

$$\frac{\text{volume}(E^{\text{II}}(d^+))}{\text{volume}(E^{\text{II}}(d))} \leq \exp\left(-\frac{1}{16n^5\omega^5}\right).$$

This shrinkage rate depends on ω and hence on the bound on $\kappa_i(d)$. We note that if the initial point is chosen as $d^0 = \lambda e$ such that $G(d^0) = 2f(d^0) = 2B(d^0)$, then we can obtain a simpler bound. We denote

$$\text{Lev}_0^{\text{II}} := \{d : G(d) \leq G(d^0)\}.$$

Then, for $d \in \text{Lev}_0^{\text{II}}$ with $f(d) = B(d)$,

$$\begin{aligned}\kappa_i(d) &= \frac{f(d)^2}{f(d)a_i^T(ADA^T)^{-1}a_i} = \frac{(\frac{G(d)}{2})^2}{f(d)a_i^T(ADA^T)^{-1}a_i} \\ &\leq \frac{(\frac{G(d^0)}{2})^2}{f(d)a_i^T(ADA^T)^{-1}a_i} = \frac{f(d^0)^2}{f(d)a_i^T(ADA^T)^{-1}a_i} \\ &\leq \frac{f(d^0)B(d^0)}{\delta_0^2} \leq \zeta^{\frac{2}{n}} \cdot \det(AA^T)^{\frac{1}{n}},\end{aligned}$$

where $\zeta = \frac{\text{volume}(E_0^{\text{II}})}{\text{volume}(\mathcal{B})}$ which is the ratio of the volume of the initial ellipsoid over the volume of the ball contained in the feasible region P.

By scaling and changing variables we can assume, without loss of generality, $AA^T = I$. Then Algorithm 3.1 is a kind of ball method [2] for which Todd [25] and Goffin [5] show that an exponential number of iterations may be required. But here, unlike the ball method, we can use Newton's method to solve (P_{f+B}), thus obtaining a quadratic convergence rate; this follows from Theorem 3.1, since the Hessian of G is bounded in the compact level set.

We now state a Newton scaling algorithm for solving (P_{f+B}).

ALGORITHM 3.2.

- Initialization. Take $\tilde{d}^0 = e$ and scale it to d^0, i.e., $d^0 = \mu^0\tilde{d}^0$ with $\mu^0 = \sqrt{B(\tilde{d}^0)/f(\tilde{d}^0)}$; set $k = 0$.
- For $k = 0, 1, \ldots,$ do
  If a convergence condition holds, stop. Otherwise, perform a line search along the Newton direction, that is: take $\tilde{d}^{k+1} = d^k + \lambda^k d_{nt}$ where $d_{nt} := -(\nabla^2 G(d^k))^{-1} \cdot \nabla G(d^k)$, and $\lambda^k = \arg\min\{G(d^k + \lambda d_{nt}) : \lambda \geq 0\}$. Let $d^{k+1} = \mu^{k+1}\tilde{d}^{k+1}$ with $\mu^{k+1} = \sqrt{B(\tilde{d}^{k+1})/f(\tilde{d}^{k+1})}$. Set $k = k + 1$ and repeat.

The following theorem describes the convergence properties of this algorithm.

THEOREM 3.6. *Suppose* (A1) *holds, that is,* $\text{int}(P) \neq \emptyset$, *and* $\{d^k\}$ *is the sequence of points generated by Algorithm* 3.2. *Then* d^k *converges to* d^* *q-quadratically; further, Algorithm* 3.2 *solves the feasibility problem* (1) *in finitely many iterations.*

Proof. Since Algorithm 3.2 is a descent method, by Theorem 3.1 we can restrict our attention to the compact set $\mathcal{D}_e$. Let

$$\begin{aligned}\lambda_u &:= \max\{\text{largest eigenvalue of } H(d) : d \in \mathcal{D}_e\}, \\ \lambda_l &:= \min\{\text{smallest eigenvalue of } H(d) : d \in \mathcal{D}_e\},\end{aligned}$$

where $H(d)$ is the Hessian of G. Let H denote $H(d^k)$ and ∇G denote $\nabla G(d^k)$. Then for any $t > 0$ there is $d(\xi) := d^k + \xi d_{nt}$ with $0 < \xi < t$ such that

$$\begin{aligned} G(d^{k+1}) &\le G(\tilde{d}^{k+1}) \le G(d^k + td_{nt}) \\ &= G(d^k) + t\nabla G(d^k)^T d_{nt} + \frac{t^2}{2} d_{nt}^T H(d(\xi)) d_{nt} \\ &\le G(d^k) + t\nabla G(d^k)^T d_{nt} + \frac{t^2}{2}\lambda_u \|d_{nt}\|^2 \\ &\le G(d^k) - t\nabla G(d^k)^T H^{-1}\nabla G(d^k)^T + \frac{t^2}{2}\lambda_u \|H^{-1}\nabla G\|^2 \\ &\le G(d^k) - t\lambda_u^{-1}\|\nabla G\|^2 + \frac{t^2}{2}\lambda_u\lambda_l^{-2}\|\nabla G\|^2 \\ &= G(d^k) - \sigma\|\nabla G\|^2 \quad (\text{for } t = \lambda_u^{-2}\lambda_l^2), \end{aligned} \tag{16}$$

where $\sigma := \frac{1}{2}\lambda_u^{-3}\lambda_l^2$. Thus d^k must converge to d^* otherwise $G(d^k) \longrightarrow -\infty$, contradicting (A1). On the other hand, from Theorem 3.1, there is $\bar{d} > 0$ such that

$$d_i \le \bar{d} \text{ for all } d \in \mathcal{D}_e \text{ and all } i.$$

If $\|\nabla G(d)\| \le \bar{d}^{-2}$, then, for $i = 1, 2, \ldots, m$,

$$-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) - \frac{1}{d_i^2} \ge -\frac{1}{\bar{d}^2}.$$

Thus,

$$-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) \ge \frac{1}{d_i^2} - \frac{1}{\bar{d}^2} \ge 0,$$

which in turn shows that $l_i^k \le a_i^T x_c(d) \le u_i^k \ \forall i = 1, 2, \ldots, m$, and thus $x_c(d)$ is feasible for (FP). Therefore, by (16), $x_c(d^k)$ must be feasible to (FP) for $k \ge \frac{G(d^0)\bar{d}^4}{\sigma} - 1$. Hence, the algorithm solves the feasibility problem (1) in finitely many iterations.

Since for any k, $\tilde{d}^k$ is the result of one step of Newton's method, it is easy to see that there is a constant, say $\gamma > 0$, such that

$$\|\tilde{d}^{k+1} - d^*\| \le \gamma \|d^k - d^*\|^2. \tag{17}$$

On the other hand, it can be shown by Taylor's theorem that, for all $d \in \mathcal{D}_e$,

$$\frac{1}{2}\lambda_l \|d - d^*\|^2 \le G(d) - G(d^*) \le \frac{1}{2}\lambda_u \|d - d^*\|^2.$$

Thus,

$$\frac{\lambda_l}{2}\|d^{k+1} - d^*\|^2 \le G(d^{k+1}) - G(d^*) \le G(\tilde{d}^{k+1}) - G(d^*) \le \frac{\lambda_u}{2}\|\tilde{d}^{k+1} - d^*\|^2,$$

which implies

$$\|d^{k+1} - d^*\|^2 \le \frac{\lambda_u}{\lambda_l}\|\tilde{d}^{k+1} - d^*\|^2.$$

The theorem thus follows by (17). □

As in Model I, when Model II is used to solve the feasibility problem (1), there are three possible outcomes according to the values of $G_{\inf} := \inf\{G(d) : d \in R^m_{++}\}$.

THEOREM 3.7.

(i) *If* $G_{\inf} > 0$, *then there is a* $d^* \in R^m_{++}$ *such that* $G(d^*) = G_{\inf}$ *and* $x_c(d^*)$ *is feasible.*

(ii) *If* $G_{\inf} < 0$, *then* $P = \emptyset$.

(iii) *If* $G_{\inf} = 0$, *then* $\text{int}(P) = \emptyset$ *but* $P \neq \emptyset$.

Proof. The proof of this theorem is similar to that of Theorem 2.17. □

The example in the last part of the previous section can be adapted here to show that $x_c(d^k)$ may not approach the feasible region even if d^k is such that $G(d^k) \longrightarrow 0$. We should mention that this phenomenon did not occur in our computational experiments; in particular, our computational results always gave a feasible solution for this example.

3.3. Computational results. Here we briefly describe some preliminary computational experiments we made with Algorithm 3.2. We generated 81 random problems, of sizes ranging from 40×45 to 200×250 for the matrix A, with various ways to choose l and u that guaranteed a feasible solution to (FP). The instances chosen used a Graeco-Latin square design so that we could make statistical tests from a relatively small number of observations. Solving these problems using MATLAB [16] on a Sun Sparcstation 2 required between 0 and 10 Newton iterations and between 0.2 and 390.6 seconds (much of the latter, for an 80×125 problem, arose from the line search, since evaluation of G requires the solution of a linear system). Details may be found in [13, 15].

Our statistical analyses (see [13]) show that (i) the number of iterations for solving the feasibility problems is very stable, almost independent of the sizes of problems; and (ii) the running time needed for solving the feasibility problems depends only on n and m, i.e., the size of the problem; it does not depend on the lower and upper bounds so much.

4. Solving linear programming problems via weighted centers. In the previous sections we introduced two kinds of centers. We discuss in this section how to use them to design algorithms for solving linear programming problems.

After defining a center, there are several possible ways to use it to solve linear programming problems. For example, we can cut the current polytope by a hyperplane through the current center and throw away the part that does not contain the optimal solution and repeat this procedure; we thus get a sequence of shrinking polytopes which contain the optimal solution, as do Levin [12] and Newman [18]. Alternatively, we can push the center toward the optimum by adjusting the appropriate bound as done by Renegar [19]. In §§4.1 and 4.2 we will use the same strategies to develop our algorithms via Models I and II, respectively. Since the center is usually a solution to some linear system, one natural way to solve the corresponding linear programming problem is to solve a linear system that combines the primal and dual in R^{m+n} and an inequality linking their objective functions. By duality theory, the solution of this system provides solutions of both the primal and the dual problems. However, from a practical viewpoint there are several disadvantages to this approach. First, the combined system is a higher-dimensional problem and thus needs more storage and more computational effort; next, since all solutions are within the hyperplane defined by letting the objectives of the primal and the dual be equal, the feasible region has zero volume, and thus some perturbation is necessary to ensure that the interior is not empty and this makes the computation rather difficult. In §4.3 we will propose a pulling technique so that, empirically, solving linear programming problems can be reduced to solving a linear system without the above disadvantages.

4.1. Sliding objective method via Model I. The sliding objective function method was first proposed by Yudin and Nemirovskii [30] and Shor [22]. The idea is to reduce the linear programming problem to a sequence of feasibility problems formed by letting the objective be an extra constraint and decreasing the bound corresponding to the objective function as long as it is possible.

Suppose we want to solve (LP). The sliding objective algorithm based on model I is as follows.

ALGORITHM 4.1.

- Initialization. Use Model I to find an approximate center x_c of the system:

$$l \le A^T x \le u, \tag{18}$$

and let d be the corresponding weights. The corresponding ellipsoid is E^0. Let $u_0^0 := c^T x_c$ and $l_0^0 := c^T x_c - (f(d)(c^T(ADA^T)^{-1}c)^{\frac{1}{2}}$, so that l_0^0 is tight for the current ellipsoid E^0. Set $A \leftarrow (c, A)$, i.e., let c be the 0th column, and $l^0 := (l_0^0, l^T)^T$, $u^0 := (u_0^0, u^T)^T$. Set $k = 0$.

- For $k = 0, 1, \ldots,$ do
  Use Model I to solve system

$$l^k \le A^T x \le u^k \tag{FP$_k$}$$

and get the (approximate) center x_c^k, weights d^k, and the current ellipsoid E^k. Let $u_0^{k+1} := c^T x_c^k$, $l_0^{k+1} := c^T x_c^k - (f(d^k)(c^T(AD^kA^T)^{-1}c)^{\frac{1}{2}}$, and $u^{k+1} := (u_0^{k+1}, u^T)^T$; $l^{k+1} := (l_0^{k+1}, l^T)^T$. Set $k \leftarrow k+1$, and repeat.

The convergence property of the above algorithm is similar to that of the standard sliding objective approach of the ellipsoid method. We refer to [2] and [6] for details.

We now discuss a practical stopping criterion. The dual of (LP) is

$$\begin{aligned} &\max\ l^T y_1 - u^T y_2 \\ &Ay_1 - Ay_2 = c, \\ &y_1,\ y_2 \ge 0. \end{aligned} \tag{DP}$$

By the results of Burrell and Todd [3], (DP) is equivalent to

$$\begin{aligned} &\max\ \psi(y) := l^T y_+ - u^T y_- \\ &Ay = c, \end{aligned} \tag{DP$_1$}$$

where $y_- := (\max\{0, -y_i\})$, $y_+ := (\max\{0, y_i\})$, so $y = y_+ - y_-$. By taking

$$y = (c^T(ADA^T)^{-1}c)^{\frac{1}{2}} D(A^T z - r),$$

where $z = x_c - (c^T(ADA^T)^{-1}c)^{-\frac{1}{2}}(ADA^T)^{-1}c$ and d is such that $f(d) = 1$, Burrell and Todd [3] show that

$$\psi(y) \ge c^T z = c^T x_c - (c^T(ADA^T)^{-1}c)^{\frac{1}{2}}.$$

We can use $\psi(y)$ as a lower bound on $c^T x$. Thus the duality gap associated with x_c^k and y is

$$gap(x_c^k, y) = c^T x_c^k - \psi(y) \le (c^T(ADA^T)^{-1}c)^{\frac{1}{2}}.$$

We note that $(c^T(ADA^T)^{-1}c)^{\frac{1}{2}}$ or, equivalently, $(f(d)c^T(ADA^T)^{-1}c)^{\frac{1}{2}}$ is half the width of the current ellipsoid along direction c. The above theorems ensure that this quantity goes to zero as the ellipsoids shrink. We thus can use it, or equivalently $f(d)c^T(ADA^T)^{-1}c$, as the stopping criterion.

4.2. Obtaining an ε-optimal solution via Model II. In this section we describe how to get an ε-optimal solution via Model II. Here we define an ε-optimal solution x^ε as a feasible solution with $c^T x^\varepsilon - z^* \leq \varepsilon$. For convenience, we suppose that $\|a_i\| = 1$ for all $i = 1, 2, \ldots, m$ and $\|c\| = 1$. The idea is that we use the sliding objective function method via Model II and show that if the feasible region of the expanded system (with the objective as its 0th constraint) contains a ball $\mathcal{B}$ with radius ε, then the sliding method converges linearly. Thus within finitely many iterations an ε-optimal solution can be obtained. In the following we first state the algorithm, then prove some convergence results, and finally describe some relaxed versions. For convenience, we use H for the Hessian of G.

ALGORITHM 4.2.

- Initialization. Use Model II to find the (approximate) center x_c of the system:

$$l \leq A^T x \leq u, \tag{19}$$

and let d be the corresponding weights. The corresponding ellipsoid is E^0. Let $u_0^0 := c^T x_c$ and

$$l_0^{-1} := c^T x_c - (f(d) c^T (ADA^T)^{-1} c)^{\frac{1}{2}},$$
$$l_0^0 := l_0^{-1} - q,$$

where $q \geq \frac{3n^2(u_0^0 - l_0^{-1})}{4}$. Set $A \leftarrow (c, A)$, i.e., let c be the 0th column, and

$$l^0 := (l_0^0, l^T)^T, \; u^0 := (u_0^0, u^T)^T.$$

Set $k = 0$.

- For $k = 0, 1, \ldots,$ do
  (1) Use Algorithm 3.2 to solve system

$$l^k \leq A^T x \leq u^k, \tag{FP$_k$}$$

  i.e., find $d^k = \arg\min\{G^k(d) : d \in R_+^{m+1}\}$, where the quantities with superscripts are those corresponding to (FP$_k$). So $G^k(d)$ denotes $G(d)$ using l^k and u^k.
  (2) Let x_c^k be the center of the current ellipsoid E^k $(= E^{\mathrm{II}}(d^k)$ using l^k and $u^k)$ and set $u_0^{k+1} := c^T x_c^k$ and $u^{k+1} := (u_0^{k+1}, u^T)^T$; and $l^{k+1} := (l_0^0, l^T)^T$. Set $k \leftarrow k + 1$, and repeat.

Note that after setting the lower bound for the 0th constraint at the initialization step the lower bound for system (FP$_k$) will remain the same and the distance between the optimal value and the 0th lower bound is at least q.

In the following we discuss the convergence properties of this algorithm. We assume that there is a ball, say $\mathcal{B}$, with radius δ_0 contained in P^k, the feasible region of system (FP$_k$). We first prove some lemmas about the subiterations in Step 1 in the algorithm.

LEMMA 4.1. *Suppose there is a ball $\mathcal{B}$ with radius δ_0 contained in P^k, and let*

$$\mathrm{Lev}_k := \{d \in R_+^{m+1} : G^k(d) \leq G^k(e)\}$$

be the level set. Then, for any $d \in \mathrm{Lev}_k$,

$$\underline{d}^k := \frac{1}{G_0^k} \leq d_i \leq \bar{d}^k := \frac{nG_0^k}{\delta_0^2} \;\; \forall i = 0, 1, \ldots, m, \tag{20}$$

where $G_0^k := G^k(e) = (r^k)^T A^T (AA^T)^{-1} A r^k - (l^k)^T u^k + (m+1)$.

Proof. Since $B(d) \le G^k(d) \le G_0^k$ for all $d \in \text{Lev}_k$, $\frac{1}{d_i} \le G_0^k$ for all i, which leads to the first inequality of (20). As for the second inequality we note that $2\sqrt{f^k(d) a_i^T (ADA^T)^{-1} a_i}$ is the width of ellipsoid $E(d)$ (not $E^{\text{II}}(d)$) along a_i, so

$$\sqrt{f^k(d) a_i^T (ADA^T)^{-1} a_i} \ge \delta_0.$$

Thus,

$$f^k(d) a_i^T (ADA^T)^{-1} a_i \ge \delta_0^2. \tag{21}$$

From $\sum_{i=0}^m d_i a_i^T (ADA^T)^{-1} a_i = n$ and (21), we have, for any i,

$$d_i \le \frac{n}{a_i^T (ADA^T)^{-1} a_i} \le \frac{n f^k(d)}{\delta_0^2} \le \frac{n G_0^k}{\delta_0^2}.$$

Thus the proof is complete. □

Without the assumption of (FP_k) being consistent, the level set might be unbounded. This lemma provides us bounds for the level set in terms of the inscribing ball $\mathcal{B}$. The bounds $\underline{d}^k$ and $\bar{d}^k$ will play important roles in the convergence analysis below.

LEMMA 4.2. *Under the same assumptions as in Lemma* 4.1, *if there is* $d \in \text{Lev}_k$ *with*

$$\|\nabla G^k(d)\| \le \frac{1}{(\bar{d}^k)^2}, \tag{22}$$

then $x_c(d)$ *is feasible for* (FP_k).

Proof. Note that $\nabla G_i^k(d) = -(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) - \frac{1}{d_i^2}$ for $i = 0, 1, \ldots, m$. Now (22) implies $|\nabla G_i^k(d)| \le (\bar{d}^k)^{-2}$. Therefore,

$$-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) - \frac{1}{d_i^2} \ge -\frac{1}{(\bar{d}^k)^2},$$

whence $-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) \ge d_i^{-2} - (\bar{d}^k)^{-2} \ge 0$, which in turn shows that $l_i^k \le a_i^T x_c(d) \le u_i^k$, $i = 0, 1, \ldots, m$, and thus $x_c(d)$ is feasible for (FP_k). □

Now we show that $G^k(d^k)$ converges to zero linearly if there is a ball $\mathcal{B}$ with radius $\delta_\infty > 0$ contained in P^k for all k. Let

$$\underline{d}^\infty := \frac{1}{G_0^\infty}, \quad \bar{d}^\infty := \frac{n G_0^\infty}{\delta_\infty^2},$$

where $G_0^\infty := \max\{G(e) : u_0 \in [z_0, u_0^0]\}$ with $z_0 := \max\{c^T x : x \in \mathcal{B}\}$; note that the function G depends on the bounds l and u—here we take the maximum as u_0 varies. It is obvious that

$$\underline{d}^\infty \le \underline{d}^k, \quad \bar{d}^\infty \ge \bar{d}^k \quad \text{for all } k.$$

Thus, from Lemma 4.1, for all $d \in \text{Lev}_\infty := \cup \text{Lev}_k$,

$$\underline{d}^\infty \le d_i \le \bar{d}^\infty \ \forall i = 0, 1, \ldots, m, \tag{23}$$

and Lemma 4.2 still holds for these bounds. We define

$$\lambda_u := \max\{\text{largest eigenvalue of } H : u_0 \in [z_0,\ u_0^0], d \in \mathrm{Lev}_\infty\},$$
$$\lambda_l := \min\{\text{smallest eigenvalue of } H : u_0 \in [z_0,\ u_0^0], d \in \mathrm{Lev}_\infty\}.$$

Since $\underline{d}^\infty > 0$, $\bar{d}^\infty < \infty$, and $\underline{d}^\infty \le d_i \le \bar{d}^\infty\ \forall i$, both λ_u and λ_l are finite and positive.

We now suppose that the output of Step 1 of the algorithm d^k is such that

$$\|\nabla G^k(d^k)\| \le \varepsilon \quad \text{with } 0 < \varepsilon \le \frac{1}{2(\bar{d}^\infty)^2} < \frac{1}{(\bar{d}^\infty)^2}, \tag{24}$$

$f^k(d^k) = B(d^k)$, and $G^k(d^k) \le G^k(d^{k-1})$ (this holds if we take the initial point as $d^0 = e$ if $G^k(e) \le G^k(d^{k-1})$ and $d^0 = d^{k-1}$ otherwise).

Then from Lemma 4.2, $x_c(d^k)$ is feasible to P^k and for each i

$$(a_i^T x_c - l_i)(a_i^T x_c - u_i) \in \left(-\frac{1}{2(\bar{d}^\infty)^2} - \frac{1}{d_i^2},\ \frac{1}{2(\bar{d}^\infty)^2} - \frac{1}{d_i^2}\right). \tag{25}$$

THEOREM 4.3. *There is a number $\delta = \delta(m, n, \underline{d}^\infty, \bar{d}^\infty)$ with $0 < \delta < 1$ such that*

$$\frac{f^{k+1}(d^{k+1}) \cdot B^{k+1}(d^{k+1})}{f^k(d^k) \cdot B^k(d^k)} \le \delta.$$

Proof. By (23), we have, for all $d \in \mathrm{Lev}_\infty$,

$$d_0 \cdot B(d) = d_0 \cdot e^T d^{-1} = 1 + \frac{d_0}{d_1} + \cdots + \frac{d_0}{d_m} \le 1 + \frac{m\bar{d}^\infty}{\underline{d}^\infty} =: \bar{\delta}.$$

On the other hand, if we let d be d^k,

$$\begin{aligned}
\frac{1}{2}(u_0^k - c^T x_c^k)^2 d_0^3 c^T (ADA^T)^{-1} c &\le \frac{1}{2}\frac{u_0^k - c^T x_c^k}{c^T x_c^k - l_0^k}\left(\frac{1}{2(\bar{d}^\infty)^2} + \frac{1}{d_0^2}\right) d_0^3 c^T (ADA^T)^{-1} c \\
&\quad \text{(using (25))} \\
&\le \frac{n}{2}\frac{u_0^k - c^T x_c^k}{c^T x_c^k - l_0^k}\left(\frac{d_0^2}{2(\bar{d}^\infty)^2} + 1\right) \\
&\quad \text{(using } d_0 c^T (ADA^T)^{-1} c \le n) \\
&\le \frac{3n}{4}\frac{u_0^k - c^T x_c^k}{c^T x_c^k - l_0^k} \\
&\le \frac{3n}{4}\frac{u_0^0 - l_0^{-1}}{q} \le \frac{1}{n}.
\end{aligned}$$

Thus, letting $\Delta u := u_0^k - c^T x_c^k$ and f^{k+1} be the function f with $u = u^{k+1} = u^k - \Delta u \cdot e_0$,

$$\begin{aligned}
f^{k+1}(d) &= (r^k)^T D A^T (ADA^T)^{-1} A D r^k - (l^k)^T D u^k - \Delta u d_0 c^T x_c^k \\
&\quad + \Delta u d_0 l_0^k + \frac{1}{4}(\Delta u)^2 d_0^2 c^T (ADA^T)^{-1} c \\
&= f^k(d) + d_0 \Delta u (l_0^k - c^T x_c^k) + \frac{1}{4}(\Delta u)^2 d_0^2 c^T (ADA^T)^{-1} c \\
&= f^k(d) + d_0 (u_0^k - c^T x_c^k)(l_0^k - c^T x_c^k) + \frac{1}{4}(\Delta u)^2 d_0^2 c^T (ADA^T)^{-1} c
\end{aligned}$$

$$\leq f^k(d) - \frac{1}{d_0}\left(1 - \frac{d_0^2}{2(\bar{d}^\infty)^2}\right) + \frac{1}{4}(u_0^k - c^T x_c^k)^2 d_0^2 c^T (ADA^T)^{-1} c \text{ (using (25))}$$
$$\leq f^k(d) - \frac{1}{2d_0}\left(1 - \frac{1}{2}(u_0^k - c^T x_c^k)^2 d_0^3 c^T (ADA^T)^{-1} c\right)$$
$$\leq f^k(d) - \frac{1}{2d_0}\left(1 - \frac{1}{n}\right).$$

Suppose the $(k+1)$st output of Step 1 is d^{k+1}. Then

$$2\sqrt{(f^{k+1}(d^{k+1})B(d^{k+1})} = f^{k+1}(d^{k+1}) + B(d^{k+1}) \leq f^{k+1}(d) + B(d)$$
$$\leq f^k(d) + B(d) - \frac{1}{2d_0}\left(1 - \frac{1}{n}\right) = 2\sqrt{f^k(d)B(d)} - \frac{1}{2d_0}\left(1 - \frac{1}{n}\right).$$

Thus,

$$\frac{f^{k+1}(d^{k+1})B(d^{k+1})}{f^k(d)B(d)} \leq \left(1 - \frac{n-1}{4n}\frac{1}{d_0 B(d)}\right)^2 \leq \left(1 - \frac{n-1}{4n\bar{\delta}}\right)^2.$$

Letting $\delta := (1 - \frac{n-1}{4n\bar{\delta}})^2$, the theorem thus follows. □

In the following we consider how to use Model II to get a d^k satisfying (24) in Step 1. For this purpose we let k' denote the iteration index within Step 1.

LEMMA 4.4. *Suppose d^* is the optimal solution of Model II for G^k. Then for all $d \in \text{Lev}_\infty$,*

$$\frac{\lambda_l}{2}\|d - d^*\|^2 \leq G^k(d) - G^k(d^*) \leq \frac{\lambda_u}{2}\|d - d^*\|^2.$$

Proof. It is easy to prove this lemma by using Taylor's theorem. □

From (16), we have, where G denotes G^k,

$$G(d^{k'+1}) \leq G(d^{k'}) - \sigma\|\nabla G(d^{k'})\|^2,$$

where $\sigma := \frac{1}{2}\lambda_u^{-3}\lambda_l^2$. From the above result, after at most $\frac{G(d^0)}{\sigma\varepsilon} - 1$ iterations, we have $\|\nabla G(d^{k'})\| \leq \varepsilon$.

Therefore, to get a point satisfying (24) in Step 1, we only need to take

$$\varepsilon = \frac{1}{2(\bar{d}^\infty)^2} \tag{26}$$

and then use Algorithm 3.2 to get a d with $\|\nabla G(d)\| \leq \varepsilon$ (this can be done in finitely many iterations as shown above). The above results also give us a relaxed version of Algorithm 4.2 (we suppose that both G_0^∞ and δ_∞ are known): in Step 1 we need only find an approximate solution d^k to (FP_k) such that $\|\nabla G^k(d^k)\| \leq \varepsilon$ with ε defined by (26) and then scale d^k so that $f(d^k) = B(d^k)$.

THEOREM 4.5. *Under assumption* (A1), *Algorithm* 4.2 *(or its relaxed version) provides an ε-optimal solution of* (LP) *in finitely many iterations.*

Proof. Suppose $\{x_c^k\}$ is a sequence of centers generated by Algorithm 4.2 (or its relaxed version) and suppose that for all $k = 0, 1, \ldots,$

$$|c^T x_c^k - c^T x^*| > \varepsilon.$$

Then, there is a ball $\mathcal{B}$ with some positive number, say δ_∞, as its radius contained in P^k for all k. From Theorem 4.3, $\frac{v^{k+1}}{v^k} \leq \delta$, i.e.,

$$\frac{\text{volume}(E^{k+1})}{\text{volume}(E^k)} \leq \delta^{\frac{n}{2}},$$

for some $\delta < 1$. Since $\mathcal{B} \subset P^k \subset E^k$ for all k, we thus have

$$0 < \kappa_n \cdot \delta_\infty^n = \text{volume}(\mathcal{B}) \leq \text{volume}(E^k) \longrightarrow 0$$

as $k \longrightarrow \infty$, which is a contradiction. Thus, noting that the number of subiterations within Step 1 is finite, Algorithm 4.2 (or its relaxed version) provides an ε-optimal solution of (LP) in finitely many iterations. □

The above results can be summarized as follows: if the algorithm does not converge, then a linear convergence rate for the volume of the corresponding ellipsoid can be proved; thus, the algorithm does converge and the rate of shrinking of the volume of the ellipsoid is reduced as x_c^k approaches x^*. It can be shown, by Lemma 4.4, that if $G^k \longrightarrow 0$ linearly, $x_c^k \longrightarrow x^*$ linearly. The linear convergence rate is most likely the best. An extreme example is when $m + 1 = n$ and, after putting in the objective vector, the system becomes an $n \times n$ square system; thus, $x_c^k = A^{-1}r^k$ and $x_c^k \longrightarrow x^*$ at most linearly.

4.3. Computational techniques and numerical results. In this section, we report preliminary numerical results with Algorithm 4.2 when applied to problem (LP).

For given $d \in R_+^{m+1}$, consider $x_c(d)$ as a function of l_0, the lower bound imposed on the objective function $c^T x$. Then we have

$$c^T x_c(d) \longrightarrow -\infty \text{ as } l_0 \longrightarrow -\infty.$$

This suggests that if l_0 is chosen to be a very large negative number, finding an approximate solution of (LP) might need only one iteration, i.e., be reduced to a feasibility problem (FP). We call this technique the pulling technique, since it pulls the current center to the optimal solution. Note that most bound-update methods, e.g., Levin [12], Newman [18], and Renegar [19], decrease an appropriate bound so as to push the current center to approach the optimal solution. Our Algorithm 4.2 is a combination of these techniques, since we choose l_0^0 to be a very large negative number (pulling) but then update u_0^k (pushing). Our limited test results are reported in Table 1. The test problems we use are those in Avis and Chvatal [1] but with upper bounds on each component of x, namely,

$$\begin{aligned} &\max\ e^T x \\ &Nx \leq 10^4 e, \\ &0 \leq x \leq 10e. \end{aligned} \tag{27}$$

N is an $n \times n$ matrix with integer elements chosen randomly in the range $1, \ldots, 1000$. We reduce (27) to our form (LP):

$$\begin{aligned} &\max\ e^T x \\ &l \leq \tilde{N}x \leq u \end{aligned} \tag{28}$$

with

$$l = 0,\ \tilde{N} = \begin{pmatrix} N \\ I \end{pmatrix},\ u = (10^4 e^T, 10e^T)^T.$$

TABLE 1
Computational results of Algorithm 4.2.

Dimension	Simplex	Algorithm 4.2
10	iter=12	iter=3 (9+40+6), $\Delta = 2.9 \times 10^{-4}$
20	iter=30	iter=3 (9+58+5), $\Delta = 9.1 \times 10^{-4}$
30	iter=37	iter=4 (11+71+7+11), $\Delta = 6.6 \times 10^{-6}$
40	iter=85	iter=3 (16+152+12), $\Delta = 5.4 \times 10^{-4}$
50	iter=128	iter=4 (9+39+16+7), $\Delta = 6.9 \times 10^{-4}$

TABLE 2
The effect of q on the number of iterations.

	Dimension=30	Dimension=40	Dimension=50
$q = 10$	> 35	> 35	> 35
$q = 10^2$	> 35	> 35	> 35
$q = 10^3$	$> 35(> 114)$	$> 35(> 121)$	> 35
$q = 10^4$	$27(11 + 8 + 4 \cdot 5 + 21 \cdot 4 = 87)$	$18(16 + 13 + 2 \cdot 5 + 7 \cdot 4 + 7 \cdot 3 = 88)$	$> 35(> 134)$
$q = 10^5$	$9(11 + 16 + 8 + 17 + 5 \cdot 6 = 82)$	$6(16 + 17 + 13 + 12 + 6 + 6 = 70)$	$11(9 + 16 + 7 + 9 + 2 \cdot 6 + 5 \cdot 5 = 102)$
$q = 10^6$	$5(11 + 27 + 12 + 7 + 9 = 66)$	$4(16 + 50 + 8 + 7 = 81)$	$6(9 + 20 + 8 + 7 + 8 + 8 = 60)$
$q = 10^7$	$4(11 + 43 + 8 + 12 = 74)$	$3(16 + 76 + 12 = 104)$	$4(9 + 38 + 15 + 8 = 70)$
$q = 10^8$	$4(11 + 71 + 7 + 11 = 100)$	$3(16 + 152 + 12 = 180)$	$4(9 + 71 + 14 + 11 = 105)$

We choose $q = 10^8$. The data in the column corresponding to the simplex method are from Nazareth [17]; these are average numbers of iterations for a number of random problems. $\Delta := f(d)c^T(ADA^T)^{-1}c$ is the width of the ellipsoid in the direction c after the final iteration and it serves here as the stopping criterion (we terminate if $\Delta \leq 10^{-3}$). The stopping criterion for Model II is $\|\nabla G(d)\| \leq 10^{-5}$. The numbers in the parentheses are the numbers of iterations of Algorithm 3.2 for obtaining each approximate center. The solution times for the last three problems of dimensions 30, 40, and 50 were 43, 146, and 124 seconds, respectively. Table 2 shows the effects of q on the number of iterations. It also shows that the effect of pulling is much more significant than that of pushing. Thus, choosing an appropriate q will save computational effort. Again, the runs were performed using MATLAB on a Sun Sparcstation 2.

5. Summary and conclusions. In this paper we have analyzed and tested some generalized models for linear inequality systems and then proposed some algorithms for solving linear programming problems via these models. Foremost among these is the idea of generalizing Burrell and Todd's approach [3] which is closely related to the coordinate descent method so that Newton's method can be employed. We develop our models in §§2 and 3. The key result is that the function $v(d) = f(d)h(d)$, where v is such that $\kappa_n v^{\frac{n}{2}}$ is the volume of the ellipsoid, can be, equivalently for our purposes, replaced by $F(d) = f(d) + h(d)$ because $f(d)$ and $h(d)$ are homogeneous functions of degrees 1 and -1, respectively. The first model, Model I, is thus established by using $F(d) = f(d) + h(d)$ as the objective function. The advantage of this model is that it is a convex program and it preserves the advantages of Burrell and Todd's approach [3]. We prove the existence and uniqueness of the smallest ellipsoid of the form (3) which contains the feasible region P. The center of this smallest ellipsoid is proved to be an interior point of P; it is our first center. For solving Model I we propose a coordinate descent method obtained by modifying the Liao and Todd algorithm [14]. It is proved to have a polynomial-time bound. Since the objective function of Model I might not be strictly convex, it thus prevents the use of a full version of Newton's method, but a partial Newton step algorithm is described in [13]. To overcome this disadvantage we replace $h(d)$ by a strictly convex function $B(d)$

which forms our second model, Model II. Similarly, we propose two algorithms for solving this model: a coordinate descent algorithm and a Newton algorithm. The first one is actually a ball-like method [2] which has been shown not to be a polynomial-time algorithm. The second algorithm is a combination of Newton's algorithm and a scaling technique, which works well in practice. The statistical analysis shows that this algorithm is numerically robust. We also provide a rather detailed analysis of the outputs of these models according to their minimal values over their effective domains.

We note that while line search techniques are used in our algorithms, trust region methods can also be employed. Since our models are convex, these two techniques can be expected to perform similarly.

In §4 we develop methods for solving linear programming problems via these models. They are basically sliding objective function methods. The first one is based on Model I and enjoys geometrical convergence rates as measured by the volume of the corresponding ellipsoid and by the objective value of the linear program, respectively. The second algorithm, based on Model II, provides an ε-optimal solution of (LP) in finitely many iterations. One property of this approach is that if $c^T x_c$, the current objective value, is ε far away from z^*, the optimal value, a constant decrease (depending on ε) can be obtained. In other words, if x_c does not approach the optimal set, x_c approaches the optimal set linearly. In addition to the pushing technique similar to that used in the sliding objective function method [2] and Renegar [19], we also propose a pulling technique: we let the lower bound corresponding to the objective be a very large negative number, i.e., $l_0 \sim -\infty$ (or $u_0 \sim \infty$ for a maximization linear program). With the help of this pulling technique, solving (LP) is almost equivalent to solving a feasibility problem as shown by the numerical tests.

Finally, we comment on the major unfinished tasks that we believe are evident from this paper. The biggest one is, as mentioned above, to analyze our approach to see if Newton's algorithm is polynomial for linear programming problems within our framework. We list below some suggestions for further work in this direction:

(1) We showed that if $f(d) < 0$ for some $d \geq 0$, then the system (FP) is inconsistent. Further research might investigate the number of iterations needed to find the infeasibility if the system is not consistent.

(2) We gave artificial examples showing that, in the case that $\text{int}(P) = \emptyset$ but $P \neq \emptyset$, for both Model I and Model II, there exist sequences $\{x_c(d^k)\}$ which fail to approach the feasible region although d^k is such that $F(d^k)$ (or $G(d^k)$) approaches zero. But Algorithm 3.2 always gave feasible solutions in our computational tests. Investigation is thus needed for this kind of behavior of the algorithm.

(3) From the numerical tests, we find that the pulling technique is an important factor with regard to the number of major iterations, i.e., number of applications of Model II. The bigger q is, the fewer major iterations are needed. We thus hope to prove that solving (LP) is equivalent to finding the center of (FP_0) with a large q.

Acknowledgments. We would like to thank Yinyu Ye and two anonymous referees for very helpful comments.

REFERENCES

[1] D. Avis and V. Chvatal, *Notes on Bland's pivoting rule*, Math. Programming Study, 8 (1978), pp. 24–34.

[2] R. G. BLAND, D. GOLDFARB, AND M. J. TODD, *The ellipsoid method: A survey*, Oper. Res., 29 (1981), pp. 1039–1091.
[3] B. P. BURRELL AND M. J. TODD, *The ellipsoid method generates dual variables*, Math. Oper. Res., 10 (1985), pp. 688–700.
[4] V. CHVÁTAL, *Linear Programming*, W. H. Freeman and Company, San Francisco, CA, 1980.
[5] J. L. GOFFIN, *On the non-polynomiality of the relaxation method for systems of linear inequalities*, Math. Programming, 22 (1982), pp. 93–103.
[6] D. GOLDFARB AND M. J. TODD, *Modifications and implementation of the ellipsoid algorithm for linear programming*, Math. Programming, 23 (1982), pp. 1–19.
[7] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990.
[8] F. JOHN, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays (Courant Anniversary Volume), Interscience, New York, 1948.
[9] N. K. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[10] L. G. KHACHIYAN, *A polynomial algorithm for linear programming*, Dokl. Akad. Nauk USSR, 244 (1979), pp. 1093–1096.
[11] L. G. KHACHIYAN AND M. J. TODD, *On the complexity of approximating the maximal inscribed ellipsoid for a polytope*, Math. Programming, 61 (1993), pp. 137–159.
[12] A. Y. LEVIN, *On an algorithm for the minimization of convex functions*, Soviet Math. Dokl., 6 (1965), pp. 286–290.
[13] A. LIAO, *Algorithms for Linear Programming via Weighted Centers*, Ph.D. thesis, Cornell University, Ithaca, NY, 1992.
[14] A. LIAO AND M. J. TODD, *The ellipsoid algorithm using parallel cuts*, Comput. Optim. Appl., 2 (1993), pp. 299–316.
[15] ———, *Solving LP Problems via Weighted Centers*, CCOP Report 93-10, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1993.
[16] C. B. MOLER, J. LITTLE, S. BANGERT, AND S. KLEIMAN, *Pro-Matlab User's Guide*, The MathWorks, Inc., 1987.
[17] J. L. NAZARETH, *Pricing criteria in linear programming*, in Progress in Mathematical Programming, N. Meggiddo, ed., Springer-Verlag, New York, 1989, pp. 105–130.
[18] D. J. NEWMAN, *Location of the maximum on unimodal surfaces*, J. Assoc. Comput. Mach., 12 (1965), pp. 395–398.
[19] J. RENEGAR, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–93.
[20] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[21] N. Z. SHOR, *Utilization of the operation of space dilatation in the minimization of convex functions*, Cybernetics, 6 (1970), pp. 7–15.
[22] ———, *Cut-off method with space extension in convex programming problems*, Cybernetics, 13 (1977), pp. 94–96.
[23] G. SONNEVEND, *An analytical center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in Lecture Notes in Control and Information Sciences, No. 84, Springer-Verlag, Berlin, 1986, pp. 866–875.
[24] S. P. TARASOV, L. G. KHACHIYAN, AND I. I. ERLICH, *The method of inscribed ellipsoids*, Sov. Math. Dokl., 37 (1988), pp. 226–230.
[25] M. J. TODD, *Some Remarks on the Relaxation Method for Linear Inequalities*, Technical Report 419, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1979.
[26] ———, *On minimum volume ellipsoids containing part of a given ellipsoid*, Math. Oper. Res., 7 (1980), pp. 253–261.
[27] ———, *Improved bounds and containing ellipsoids in Karmarkar's linear programming algorithm*, Math. Oper. Res., 13 (1988), pp. 650–659.
[28] P. M. VAIDYA, *A new algorithm for minimizing convex functions over convex sets*, Math. Programming, 73 (1996), pp. 291–341.
[29] Y. YE, *Karmarkar's algorithm and the ellipsoid method*, Oper. Res. Lett., 6 (1987), pp. 177–182.
[30] D. B. YUDIN AND A. S. NEMIROVSKII, *Informational complexity and efficient methods for the solution of convex extremal problems*, Matekon, 13 (1976), pp. 3–25.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 961–977, November 1996
© 1996 Society for Industrial and Applied Mathematics
004

PARALLEL DECOMPOSITION: RESULTS FOR STAIRCASE LINEAR PROGRAMS*

ROBERT ENTRIKEN†

Abstract. This paper documents the results of an experimental computer code that implements a parallel decomposition algorithm for staircase linear programs. The tests were conducted on an IBM 3090/600E computer, which has 6 processors. This type of linear program represents a worst-case example for measuring the performance of parallel decomposition. Indeed, one may find it interesting that there is any opportunity for speedup under these conditions, but experiments on 13 small- to medium-sized "real world" test problems show that in addition to speedups of from 2 to 10 provided by decomposition alone, there are speedups of up to 3.3 provided by using 4 processors. On average, the speedup provided by 4 processors over the faster of the 2 single processor algorithms is about 2. These results can be extrapolated to situations with larger problems and/or more processors with a fair degree of confidence.

Key words. parallel algorithms and results, large-scale linear programming, parallel computers

AMS subject classifications. Primary, 65K05; Secondary, 90C06

1. Introduction.

We present the experimental results of using a Fortran77 program to solve staircase linear programs [17] by using parallel benders decomposition. This code has been run on two different shared-memory multiprocessing computers: a Sequent Balance 8000 and an IBM 3090/600E. Preliminary results on the Sequent computer were reported in [8]. More extensive results on the IBM 3090 will be reported here.

This research is built upon that in [1, 2, 4, 6, 10, 12, 16, 26]. Recent work in the area of parallel optimization can be found in [13, 21, 24, 25].

Naturally, the decomposition code must solve subproblems that are linear programs. This is accomplished by calling MINOS 5.1 [23] as a subroutine [7]. As a result of using MINOS in this fashion, the best comparison of the decomposition method is to solve the same test problems using MINOS as a stand-alone system. This approach allows many implementation differences to be eliminated and permits the merits of decomposition and parallel decomposition alone to be separated out. The tests of the decomposition code were designed to

- produce results from which to judge the merits of parallel decomposition,
- investigate the algorithm's performance under different parameter settings,
- provide performance extrapolations outside the set of test problems.

Emphasis is placed on demonstrating that decomposition and added processors provide faster solutions, with acceptable accuracy.

The following section describes the problem, the algorithm, and its proof of convergence. The third section gives details of the experimental apparatus and presents results that support the appropriateness of parallel decomposition on staircase problems. Finally, the conclusions section argues the case for parallel decomposition in general and traces directions for future applications of this work.

Our presentation is based on the suggested standards of [3] and [20]. In addition, several similar presentations were considered, including [15, 18].

* Received by the editors November 3, 1994; accepted for publication (in revised form) May 9, 1995.

† Department of Operations Research, Stanford University, Stanford, CA 94305–4022 (entriken@or.stanford.edu).

2. Theory. The following is the standard mathematical formulation of a staircase linear program with N steps:

$$\begin{array}{lccccccc}
\min & c_1^T x_1 & + & c_2^T x_2 & +\cdots+ & c_N^T x_N = z & & \\
\text{s.t.} & A_1 x_1 & & & & \geq b_1, & & \\
& B_1 x_1 & + & A_2 x_2 & & \geq b_2, & & \\
& & \ddots & & \ddots & \vdots & & \\
& & & B_{N-1} x_{N-1} & + & A_N x_N \geq b_N, & & \\
& & & & & x_i \geq 0, & i \in \{1, \ldots, N\}. &
\end{array}$$

This form of problem is quite common in planning and scheduling models because of the need to make time-dependent decisions x_i. Discrete-time optimal control problems also take this form when the state equation is represented explicitly for each time period.

2.1. Forming subproblems. One staircase decomposition subproblem can be formed from each step in the staircase, but it is often more practical to combine adjacent steps to make larger subproblems, especially when the number of processors is limited. The formulated subproblems are solved on an "as-available" basis using p processors. Subproblems are considered to be *available* when they have just received new information from a different subproblem (always an adjacent step on the staircase (see Figure 1). Since in Benders Decomposition, the subproblems are defined by a partitioning of the variables (or columns of the constraint matrix), we prefer to speak of passing information to the right and left subproblems (or subset of adjacent columns).

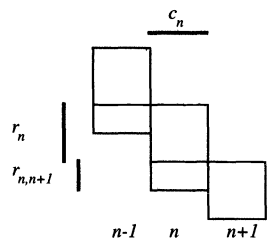


FIG. 1. *Dimensions of a step.*

Figure 1 helps define some important dimensions of the staircase. Namely, c_n is the number of columns in the nth step, r_n is the number of rows in the nth step, and $r_{n,n+1}$ is the number of coupling rows in the nth step.

The information that is communicated consists of primal and dual solutions of the dimension of the coupling rows $r_{n,n+1}$, and it is always passed to subproblems constructed from adjacent column partitions. Primal subproblem solutions are passed rightward, and dual subproblem solutions are passed leftward.

A serial implementation of nested benders decomposition was documented in [7], which describes the following subproblem formulations in detail. There are three types of formulation: leftmost, rightmost, and middle. The leftmost formulation

corresponds to the variables x_1:

$$\begin{array}{lllllll} \min & c_1^T x_1 & & & + & \theta_1 = z_1 \\ \text{s.t.} & A_1 x_1 & & & & \geq b_1, \\ & B_1 x_1 & - & I y_1 & & = 0, \\ & & & \tilde{\Pi}_2^k y_1 & + & \tilde{\delta}_2^k \theta_1 \geq \tilde{p}_2^k, \\ & x_1 & & & & \geq 0. \end{array}$$

The constraints $\tilde{\Pi}_2^k y_1 + \tilde{\delta}_2^k \theta_1 \geq \tilde{p}_2^k$ are a collection of k individual constraints, of the form $\tilde{\pi}_2 y_1 + \tilde{\delta}_2 \theta_1 \geq \tilde{p}_2$, collected from a sequence of k solutions $(\tilde{\pi}_2, \tilde{\delta}_2, \tilde{p}_2)$ from subproblem number two. The symbol $\tilde{\delta}_2$ equals one if the solution of the second subproblem is optimal or zero if the solution is infeasible. If the solution is unbounded, no constraint is added.

The rightmost formulation incorporates the variables x_N:

$$\begin{array}{lllll} \min & & & c_N^T x_N = z_N \\ \text{s.t.} & \tilde{\delta}_{N-1} w_N & & = 0, \\ & \tilde{y}_{N-1} w_N & + & A_N x_N \geq b_N, \\ & w_N, & & x_N \geq 0. \end{array}$$

The parameters $\tilde{\delta}_{N-1}$ and $\tilde{y}_{N-1}$ are replaced by the latest solution from the previous subproblem. The value of $\tilde{\delta}_{N-1}$ is one if the solution of the previous subproblem is optimal, zero if it is unbounded, and undefined if it is infeasible. In the latter case, no modification would be made.

There are $N-2$ middle subproblems for the middle steps of the staircase. They correspond to the variables x_n, for $n = 2, \ldots, N-1$:

$$\begin{array}{lllllllll} \min & & & c_n^T x_n & & & + & \theta_n = z_n \\ \text{s.t.} & \tilde{\delta}_{n-1} w_n & & & & & & = 0, \\ & \tilde{y}_{n-1} w_n & + & A_n x_n & & & & \geq b_n, \\ & & & B_n x_n & - & I y_n & & = 0, \\ & & & & & \tilde{\Pi}_{n+1} y_n & + & \tilde{\delta}_{n+1} \theta_n \geq \tilde{p}_{n+1}, \\ & w_n, & & x_n & & & & \geq 0. \end{array}$$

The parameters $\tilde{\delta}_{n-1}, \tilde{y}_{n-1}, \tilde{\Pi}_{n+1}, \tilde{\delta}_{n+1}$, and p_{n+1} come from the solutions of adjacent subproblems and have characteristics the same as those for the leftmost and rightmost subproblems.

Assuming that no row or column permutations are necessary to obtain the staircase pattern, a simple heuristic is used to provide subproblem dimensions. The purpose of the heuristic is to partition the set of ordered columns, $\mathcal{C}$, of the constraint matrix into a user-specified number of sets N, one for each subproblem. The columns in each set are contiguous, and they are chosen so as to minimize the number of coupling rows between the columns of adjacent sets. First a profile function $(f(\cdot) : \mathcal{C} \rightarrow \mathcal{R})$, mapping the set of columns into the set of rows $\mathcal{R}$, is calculated:

$$f(j) = \min_{j < j' \leq |\mathcal{C}|} \{i \in \mathcal{R} : A_{ij'} \neq 0\} - \max_{1 \leq j' \leq j} \{i \in \mathcal{R} : A_{ij'} \neq 0\}.$$

Then, $N-1$ local minima are found so that the distances between them are nearly the same. It is also desirable to get the local minima as small as possible. If the staircase does not have enough steps to supply N subproblems, a warning is printed, and the number of steps found is used.

2.2. The parallel decomposition algorithm. When a subproblem is solved to optimality, both its primal and dual solutions are communicated. (This is the basis of parallelism for the staircase example; one subproblem spawns work for two others.) When a subproblem is infeasible, only the dual solution is sent, and when it is unbounded, only the primal solution is sent. As a result, the Phase I algorithm (for obtaining a primal feasible solution) is a serial one, and computational results exhibit this property.

The parallel decomposition algorithm is asynchronous, which makes it difficult to describe in a traditional stepwise fashion. In fact, every time it is run on the same problem, a different sequence of steps is likely to occur. As a result, our description centers around the concept of having p processors to service the available subproblems. Available subproblems are serviced immediately by one of the processors, or they wait in line if the processors are all busy.

Steps. We initialize the parallel decomposition algorithm by making all of the subproblems available. Each is processed according to the following steps:

Receive message(s), Make modification(s), Solve, Communicate solution, Wait message.

In the last step before waiting, a subproblem communicates its solution only if the solution has changed since the last time the subproblem was solved. When no new messages are generated by a subproblem, there is less parallelism. When no more subproblems are available, the algorithm has converged.

Messages and modifications. Since this is an implementation of benders decomposition, a primal solution can be received from the left subproblem and a dual solution can be received from the right subproblem. The subproblem formulations make clear how this information is introduced.

Communication. Different information is communicated under the differing exit conditions of the solve step, but regardless of the exit conditions, if a new solution is identical to the previous one, no communication takes place.

When subproblem n is optimal, the primal extreme point is always passed to subproblem $n+1$ if $n < N$, but the dual extreme point is passed to subproblem $n-1$ only if the sender has already had a constraint added. This practice guarantees dual feasibility of the information passed leftward, which is needed for convergence.

When subproblem n is unbounded, in addition to the primal extreme ray, a primal extreme point is communicated to subproblem $n+1$ to an adjacent subproblem. Since MINOS is an implementation of the simplex method, the extreme point is available and used.

When subproblem n is infeasible, the dual extreme ray is passed to subproblem $n-1$; no primal solution is communicated.

Some test problems spend much of their time with infeasible subproblems. An extreme example is SC205, which has only a single, nonzero objective coefficient in its leftmost subproblem. This makes all but the leftmost subproblem feasibility problems; we need only find a feasible point because the objective is vacuous. The Phase I decomposition algorithm can be made parallel by passing an infeasible primal solution to the right, but this information must be discarded if the algorithm converges prematurely. At the time of this writing, we have yet to implement this feature.

2.3. Convergence and the termination criterion. The proof of convergence focuses on there being a finite number of dual solutions that can be passed leftward and is inductive on the rightmost subproblems.

PROPOSITION 2.1. *The parallel decomposition algorithm converges.*

Proof. Assume that the subproblem solution method always finds a dual extreme point. If the same point should be passed twice, there is a condition of optimality between the two communicating subproblems, and the algorithm stops communicating until a new dual point is passed or all of the subproblems have stopped communicating.

With the two rightmost subproblems, the right subproblem has a finite number of dual extreme points that can be passed leftward. In the worst case, all of the dual extreme points must be passed, but they are finite in number and the communication stops, leading to convergence between these two subproblems.

With the k rightmost subproblems, we assume that the $k-1$ rightmost will converge, and thus the dual feasible region of subproblem $k-1$ no longer changes and has a finite number of extreme points. The situation is then the same as for the two rightmost subproblems and they must also converge. □

This proof is identical to that for the serial algorithm, so what effect has parallelism on convergence? In fact, a single processor can simulate the parallel algorithm by solving and passing information in the same sequence as the parallel algorithm. The possible advantage of parallelism is that by utilizing multiple processors and searching for a solution on multiple fronts, there is a greater chance that a shorter path to the solution will be found. In the worst case, the algorithm will solve the problem in a recursive fashion from right to left, as in the above proof, but there may still be an opportunity for speedup.

Termination occurs when all subproblems have stopped communicating, which is when there is a pairwise balance across all adjacent subproblems. To test whether a pairwise balance exists between subproblems n and $n+1$, the objective value of subproblem $n+1$ is sent along with the dual extreme point and compared with the dual objective θ_n for the subproblems following n. Since θ_n is a lower bound on the value of z_{n+1}, if

$$z_{n+1} - \theta_n < tol,$$

then the constraint will be nonbinding, and the solution will not change. The value of *tol* is the default feasibility tolerance used by MINOS.

2.4. Discarding constraints. Typically, a large number of constraints will be added to a given subproblem. However, not all of them are necessary to obtain an optimal solution. At most $r_{n,n+1}$ (see Figure 1) can be binding in subproblem n at the final solution. We actually keep $r_{n,n+1}+2$ for good measure. The decomposition code overwrites the added constraints that are no longer binding by replacing the constraint that has been slack for the greatest number of solves.

2.5. Communication. Messages contain a quantity of information that is a function of the number of coupling constraints $r_{n,n+1}$ between the communicating subproblems. The maximum message length is 1424 bytes for all the test problems, which indicates a very light communication burden.

Since this is a shared-memory computer, we have implemented intersubproblem communication with *mailboxes* as follows. Sending a message involves loading it into a buffer and copying the buffer into the proper mailbox. Receiving a message involves copying it from the proper mailbox into a buffer. Subproblems have one mailbox for each neighboring subproblem. Each mailbox is capable of holding only one message. If a new message arrives before the old one is read, the old one is discarded. Discarding messages in this fashion does not affect finite convergence (but, according to Ho et al. [19], it is possible for such retained information to be used to speed convergence).

2.6. Basis factorization. MINOS maintains a basis factorization that is updated by the decomposition code as appropriate after each modification to a subproblem. The routines for this purpose are in the software package called LUSOL and are documented in [14]. As a result of making both row and column updates, the factorization needs to be recalculated only when it becomes inaccurate or too large. The default settings from MINOS are used to govern refactorization.

3. Testing. The following experiments were designed to test the performance of the parallel decomposition algorithm. A test suite of 13 staircase linear programs was solved with different partitions and different numbers of processors. The conclusions are that the algorithm is consistently well behaved in its use of additional processors and that it outperforms the serial algorithm (the simplex method) in most cases when using only 4 processors. When the problems become larger and when the number of processors increases, the algorithm continues to perform well with no signs of degradation.

3.1. The test environment. We report the test environment in Table 1 so that the interested reader may reproduce similar conditions on a variety of parallel machines.

TABLE 1
The test environment.

Feature	Setting
Language	Fortran77 with IBM parallel Fortran extensions
Compiler	IBM parallel VS Fortran with VS Fortran V2, rel. 1.1
Compiler options	No vector, No parallel, optimize level 3, dynamic shared common
Computer	IBM 3090/600E, 2 Gbytes shared virtual memory and 128 Mbytes real extended memory
Operating system	MVS/XA V2.2.0
Code + local common	0.62 Mbytes
Shared system common	0.27 Mbytes
Shared data common	1.60 Mbytes
Total shared common	1.87 Mbytes
Total memory	2.49 Mbytes
Tolerances	MINOS defaults
Message passing	w/o locks
Job flow control	with locks

Software. The computer code presented here is based on MINOS 5.1. It uses the linear program (LP) portion of MINOS with a few extra routines spliced in. MINOS consists of three basic modules: *input, solve,* and *output.* The parallel decomposition algorithm has two additional modules. The first, *form subs,* is inserted after the MINOS input module and the second, *process subs,* governs parallel MINOS solves and decomposition message handling. In addition, a small amount of extra work is involved in collating the many subproblem solutions into one overall solution before they are output. Thus, the input/output work is slightly greater for decomposition.

Input files. These files are input using the MINOS input module. The standard Mathematical Programming System (MPS) file is input to determine the problem data. It is assumed that this MPS file describes an LP that has a block-diagonal or staircase structure. Normal MINOS input files are sufficient to solve the LP as a single, large problem. However, to decompose a block-diagonal structure into N subproblems, additional information must be provided in the Decomposition Specifi-

TABLE 2
Test problem dimensions.

Prob. #	Prob name	Rows	Cols	Elems
1	SC205	206	203	552
2	SCTAP1	301	480	2052
3	SCFXM1	331	457	2612
4	GROW7	141	301	2633
5	SCSD1	78	760	3148
6	SCFXM2	661	914	5229
7	GROW15	301	645	5665
8	SCSD6	148	1350	5666
9	SCFXM3	991	1371	7846
10	SCTAP2	1091	1880	8124
11	GROW22	441	946	8318
12	SCTAP3	1481	2480	10734
13	SCSD8	398	2750	11334

cations (DSPECS) file, which contains the *additional* information needed to complete the decomposition of a staircase linear program—essentially the number of subproblems and the number of processors.

Output files. Each processor has a standard output file, and the MINOS-type iteration log of each subproblem solved by each processor is appended to the corresponding file. Solving subproblems on an as-available basis means that these output files contain a somewhat random sequence of solutions to the various subproblems. A processor working on the leftmost subproblem assembles the overall LP solution in its standard output file at the conclusion of the computation. In addition, one short summary file is written by the root process that contains the summary statistics.

3.2. The test suite. The test problems were chosen from a collection of 53 linear programs used by Lustig [22] in a performance evaluation of the simplex method. Included in his report are pictures of the patterns of nonzeros. Table 2 lists the dimensions for the problems in the test suite. They are arranged by row and ordered by the number of nonzeros in the constraints. All are available from *netlib* [5, 11].

Table 3 contains the staircase dimensions for each of the test problems. The default number of subproblems created is listed along with the number of steps in the staircase. The minimum and maximum dimensions of each step are given, along with the number of coupling rows between adjacent steps, as described earlier in Figure 1.

3.3. Test designs and results. The physical properties *power, work,* and *time* are excellent terms to describe the performance of a parallel algorithm. In the computing environment, the unit of power is a CPU, the unit of work is a CPU second, and the unit of time is a second, as measured with a wall clock. One can view work, or CPU time, as related to the rent paid for use of the computer. The absolute performance measure, however, is usually the elapsed time needed to obtain a solution. In the following, we refer to the five cases for running the software as

- MINOS—solving the problem with MINOS with one processor,
- DECOMP/p—solving DECOMP with p processors, where p is a value from one to four.

Power. A good parallel algorithm has two properties. First, it makes use of the CPU power; when a new CPU is made available, it is actually used. Most algorithms do not achieve this. Eighty percent is often considered very good. Figure 2 displays the average CPU power applied to solve each test problem when decomposed into

TABLE 3
Test problem step dimensions.

Problem number	Prob name/ # subs	# of steps	Min rows	Max rows	Min cols	Max cols	Min couple	Max couple
1	SC205/7	18	11	15	11	14	5	5
2	SCTAP1/10	10	30	30	48	48	18	18
3	SCFXM1/4	4	66	92	99	126	5	9
4	GROW7/7	7	20	20	43	43	20	20
5	SCSD1/3	3	20	37	190	380	10	10
6	SCFXM2/8	8	66	92	99	126	5	9
7	GROW15/15	15	20	20	43	43	20	20
8	SCSD6/6	7	20	20	190	210	10	10
9	SCFXM3/12	12	66	92	99	126	5	99
10	SCTAP2/10	10	109	109	188	188	62	62
11	GROW22/22	22	20	20	43	43	20	20
12	SCTAP3/10	10	148	148	248	248	86	86
13	SCSD8/7	39	10	17	70	90	10	10

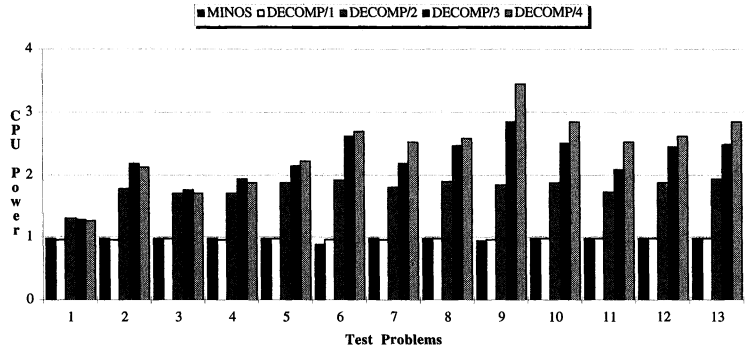


FIG. 2. *CPU power used for each test problem.*

the default number of subproblems given in Table 3. It shows that the algorithm has little trouble utilizing more CPU power, especially on the larger problems. Of course there is a limit. It should be evident that at most $N-1$ processors can be kept busy by the algorithm, where N is the number of nodes/subproblems. These experiments were run with at most four CPUs because although the 3090/600E has six, it cannot effectively offer more than four CPUs in a multiuser environment.

Notice also that the used CPU power for Problem 1 (SC205) is at or near one regardless of p, because in this case the computer spends most of its time obtaining a primal feasible solution.

Useful work and idle time. Figure 3 illustrates how the data for Figure 2 and all subsequent figures were collected. The solid black lines in the figure represent useful serial work doing input and output. These times are ignored. The only times reported are those for the parallel phase, which represents a majority of the work done, especially for large problems.

After data are read from disk, the work fans out to p independent strings of work with one barrier between the form subs and process subs steps. The parallel

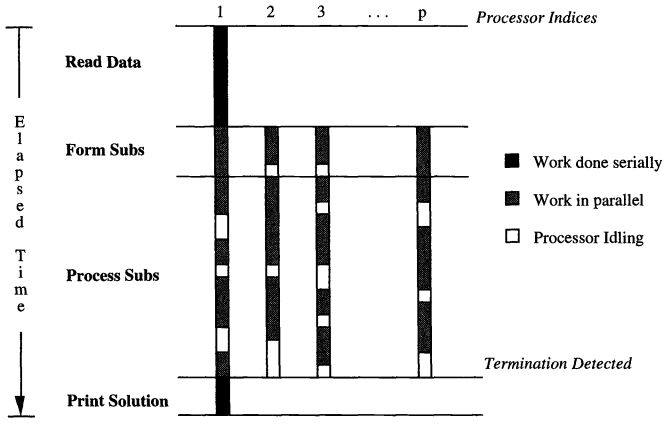


FIG. 3. *Strings of work.*

vertical lines in Figure 3 are shaded gray with intermittent white sections. This is to represent times of useful and idle work. Useful work is spent forming and solving subproblems, whereas idle work is spent counting. In the multiuser environment on the IBM 3090/600E, it is important to "waste time" counting because we must measure how much idle time is really being incurred. A production code would not do this because idle time would be better filled with useful work from other users. Counting idle time degrades performance at the expense of simulating a "generic" computing environment.

The CPU power in Figure 2 is the ratio of useful work (total length of all the gray lines) to the total work (total length of the gray and white lines). It is a measure of the effective CPU power applied to solving the problem.

The workings of both the computer and the algorithm present some difficulties in obtaining a deterministic measure of total work. Thus, the same program configuration was run three successive times yielding different results and the reported times are the average of the three successive runs.

Work. The second property of a good parallel algorithm is that the total work does not increase as the number of processors increases. Figure 4 gives the total parallel work done on each test problem using both MINOS and DECOMP ($p = 1, 2, 3, 4$). Notice that the work actually decreases from $p = 3$ to $p = 4$ for Problem 7. This is possible because there is no control over the path taken to the solution, and different paths can be taken for different numbers of processors. The conclusion to be drawn from these results is that the total work does not substantially increase as the number of processors increases.

Time. In addition to the effective use of CPU power, we obtain respectable reductions in elapsed times as shown in Figure 5. A more effective presentation is made by normalizing the scale for each test problem. In Figure 6, the times of each individual test problem have been normalized by the time used by MINOS. The result is called the speedup over MINOS. In this case, a value of 2 would mean that the decomposition algorithm found the solution twice as fast as MINOS. The figure shows that parallel decomposition is consistently better than the simplex method on the larger problems.

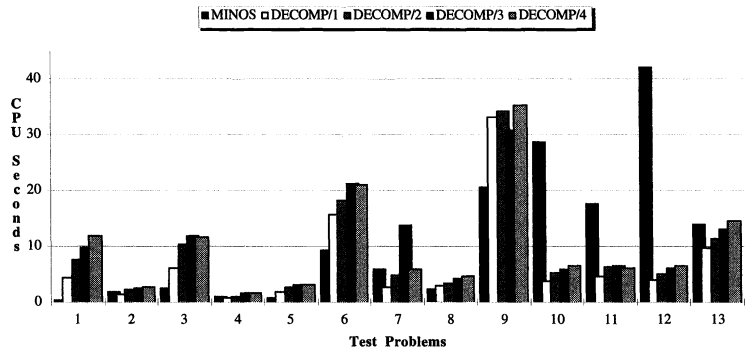


FIG. 4. *Work required to solve each test problem.*

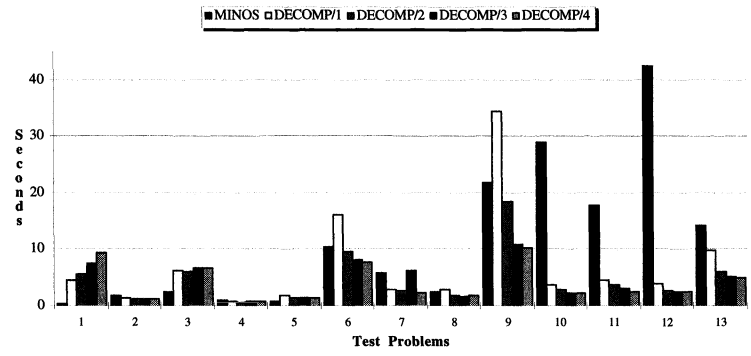


FIG. 5. *Time required to solve each test problem.*

Speedups. There are two benefits derived from parallel decomposition that can generate speedups. The first is that for the larger problems, decomposition alone ($p = 1$) has offered a speedup. For instance, Problem 12 (SCTAP3) is solved 10.5 times faster just because of a change of algorithm.

The second benefit, naturally, is derived from using more CPU power. Figure 7 is a display of elapsed times that were normalized by the time used by DECOMP/1 for each test problem. With this perspective, we can effectively judge the benefits of adding processors. Notice that for Problem 12, the computation was sped up by an *additional factor* of 1.6 over DECOMP/1 because of the addition of 3 CPUs for a total for 4 CPUs. The overall benefit provided by parallel decomposition with 4 processors was a factor of 16.8 speedup as seen in Figure 6. Note that this seemingly superlinear speedup over the simplex method is due to the multiplicative effect of combining decomposition with parallelism.

The number of subproblems. This experiment demonstrates the increase in over-

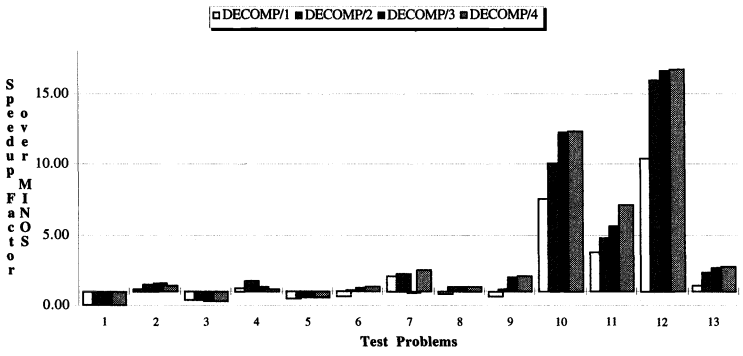


FIG. 6. *Speedup over MINOS for each test problem.*

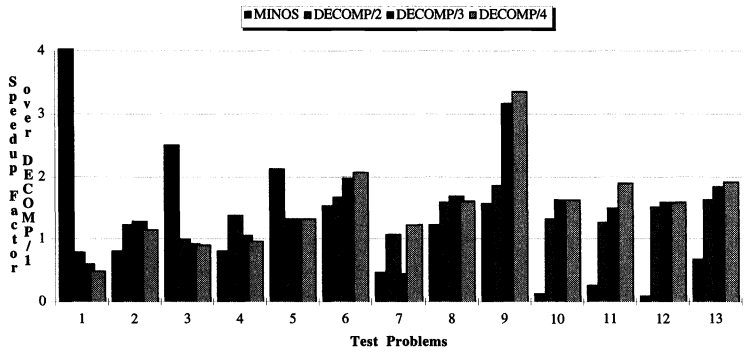


FIG. 7. *Speedup over DECOMP/1 for each test problem.*

head of decomposition as the number of subproblems increases. The largest test problem, SCSD8, has 39 steps. If subproblems are limited to a discrete number of steps, then the number of subproblems is limited to the set $\{2, 3, \ldots, 39\}$. Note also that there are 38 ways to partition the 2-subproblem case. The number of steps per subproblem was chosen to be nearly the same in each case.

Even though our communication times are negligible, there is increasing overhead involved in making the proper response to all received messages as the number of subproblems increases. Also, there is a limit to the benefit from solving a staircase with decomposition. These two effects combine in this experiment. Figure 8 shows a typical bowl shape for the total work used to solve SCSD8 when the number of subproblems N varies between 2 and 39 and the number of processors p varies from 1 to 4.

The run requiring the least amount of work is for $N = 9$ and $p = 1$ with 8.5 CPU seconds. After the average size of a subproblem begins to fall below 3 steps, $N \approx 13$,

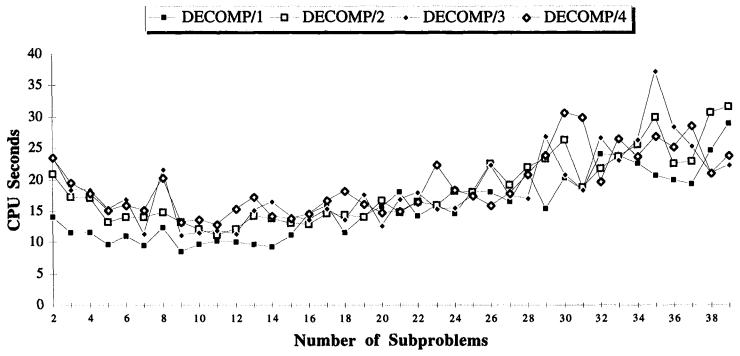


FIG. 8. *Work to solve SCSD8 using increasingly fine partitions.*

the total work increases steadily. This is consistent with a general observation that a staircase problem with less than 2000 nonzeros is not worth decomposing with this software [9], as the decomposition overhead begins to outweigh its benefits. However, with more processors the results are different.

The run solving the fastest overall, as seen in Figure 9, is $N = 11$, $p = 4$ with 3.1 elapsed seconds. For $p = 3$, the minimum is 3.6 elapsed seconds occurring at $N = 9$; for $p = 2$, the minimum is 5.3 elapsed seconds with $N = 11$; and for $p = 1$, the minimum is 7.8 elapsed seconds with $N = 9$.

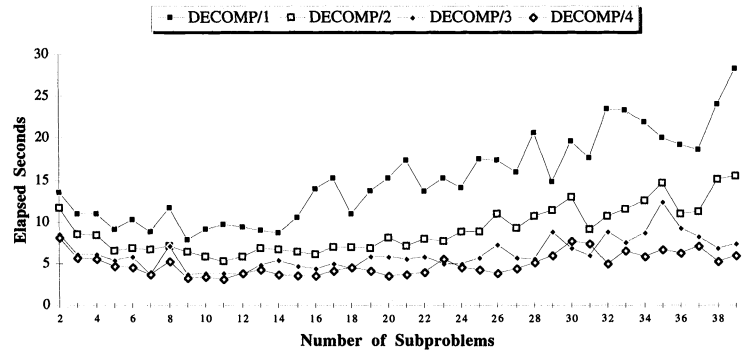


FIG. 9. *Time versus the number of subproblems for SCSD8.*

The best speedup for a fixed number of subproblems is at $N = 21$, $p = 4$, with a factor of 4.8; see Figure 10. The best serial time versus the best parallel time is $7.8/3.1 = 2.5$, but this was obtained only after an exhaustive search.

The trend, though, is consistent: more processors make decomposition faster. The effect of the number of processors on solution time is likely to be a function

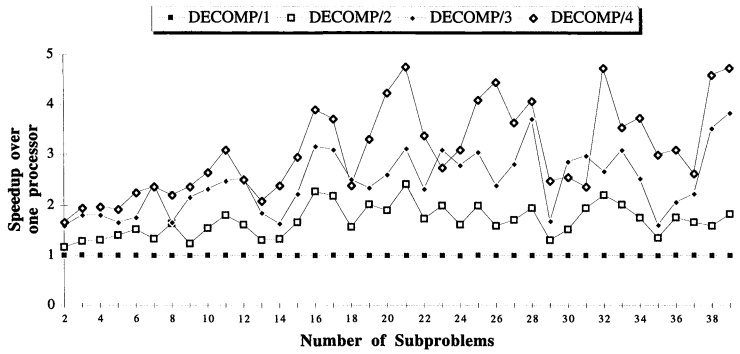


FIG. 10. *Speedup versus the number of subproblems for SCSD8.*

of the solver more than anything else. Small LP subproblems are best solved with a vectorized tableau method. Specialized subproblems like network flows are best solved by combinatorial algorithms. MINOS, the solver used here, performs best on medium-sized problems (relative to decomposition). Solvers should be chosen according to the size and nature of the subproblems, and decomposition offers the opportunity to choose.

The number of processors. This is a study on the effective use of processors. At a time when the computer was lightly loaded, SCSD8 was solved using from 1 to 7 virtual Fortran processors while decomposed into 39 subproblems. In IBM Parallel Fortran, a Fortran processor is a series of MVS operating system tasks, so more than 6 may be requested for a 6-processor machine. Seven is the limit based on memory restrictions.

Figure 11 verifies the effective use of more processors for this problem. Here, speedup is calculated relative to the solution time for decomposition with one processor. Naturally, the point (1,1) is represented. The diagonal line shows the ideal: linear speedup. The superlinear speedup for 6 processors is due to the algorithm using much less work to solve the problem (see Figure 12). CPU power, as usual, is the nonidle CPU time divided by the elapsed time. It measures the effective CPU power applied to solving the problem. The figure shows that the decomposition algorithm readily applies 7 processors to 39 subproblems.

Next, Figure 12 graphs the dichotomy of work versus time for varying numbers of processors. Sharp dips in the amount of work, as for the six-processor case, can only be attributed to good fortune. The elapsed time was 3.7 seconds, close to the record 3.1 seconds for $N = 11$ and $p = 4$. Experiments on a dedicated machine with many more than four processors could settle many uncertainties as to the true source of benefits of parallel decomposition. As of this writing, we can say only that they exist.

3.4. Performance extrapolations. In the next two experiments, we re-examine the results for a fixed number of subproblems by grouping a few of the test problems by family. We consider how, as the problems in a family get larger, parallel decomposition should perform on even larger problems.

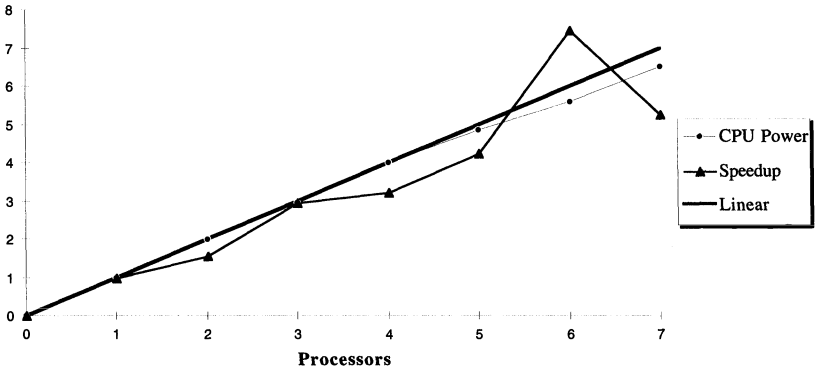


FIG. 11. *Power versus processors for SCSD8.*

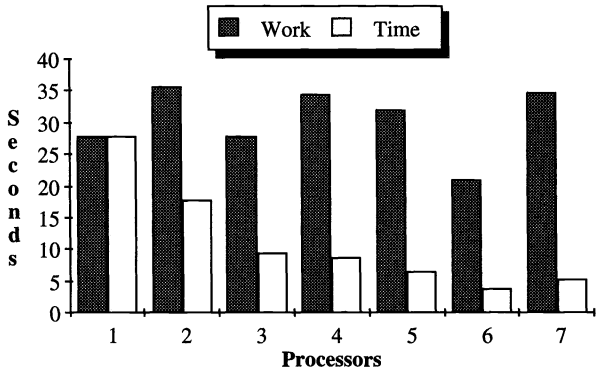


FIG. 12. *Work and time versus processors for SCSD8.*

Extending the staircase. This is the first of two discussions regarding extrapolation of the results beyond the test suite. One way to make a staircase problem larger is to add more steps. This means that either the planning horizon is lengthened or it is represented in finer detail. There are two such series in our test suite: GROW and SCSD. The speedup results from Figure 7 are reproduced in Figure 13, and we can see that as the length of the staircase extends, the parallel algorithm's performance is not degraded, but is enhanced.

Model complexity. Another method of increasing the size of staircase problems is to add more complexity to the model, i.e., to disaggregate. For instance, "dairy products" becomes milk, cheese, yogurt, and ice cream. Adding complexity allows a model to give a more detailed solution and the modeler to address interactions more

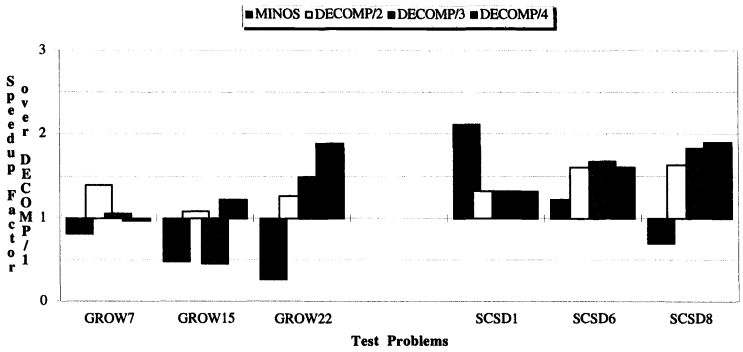


FIG. 13. *Speedup over DECOMP/1 for extending staircases.*

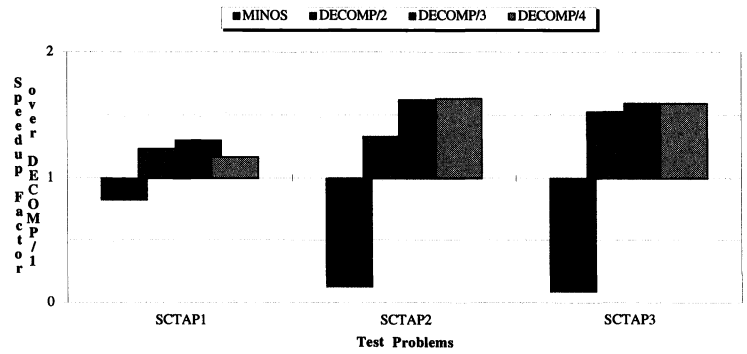


FIG. 14. *Speedup over DECOMP/1 for more complex staircases.*

specifically.

The SCTAP series of problems keeps the same number of steps, but increases the number of rows, columns, and nonzeros per step. Figure 14 is a reproduction of the speedups for this series. It shows that the simplex method has increasing difficulty with this problem, while the performance of the parallel decomposition algorithm does not degrade.

4. Conclusions. The main conclusion to draw from the computational results is that there is a significant degree of parallelism in LP decomposition, even for this worst-case situation of staircase problems. Second, if serial decomposition does well on a given problem then parallel decomposition does also. This is not surprising, but what we have also seen is that even when serial decomposition is slower than the simplex method, adding more processors can make it faster. In general, adding more processors will help, but there is a limit, mostly determined by the number of steps in the staircase.

The subproblems need not be linear programs. In general, convex functions and regions will be approximated by decomposition as piecewise linear functions and extreme-point representations. More pieces and more extreme points are added to obtain a desired accuracy.

Studies have been done [19] that demonstrate the efficacy of parallel computation when decomposing a two-stage problem with many second-stage subproblems. This paper introduces a new source of parallelism in a case that is not as obvious. It shows that even when the following stages do not fan out to multiple subproblems, there is indeed a great amount of parallelism to be exploited. In most of the staircase test problems, the parallel algorithm solves faster with multiple processors than with only one, regardless of whether the comparison is made to MINOS or to DECOMP itself. Studies also show that the algorithm continues to perform well for extrapolations to larger problems and more processors.

Acknowledgments. Professor George Dantzig, Patrick McAllister, Professor Michael Saunders, and John Stone supplied advice and support on the art of large-scale optimization. The parallel computer time used in the development and testing of the decomposition software were provided by the Oak Ridge National Laboratory with help from Tom Dunnigan, Al Geist, and George Ostrochov and the IBM T. J. Watson Research Center with help from John Forrest, Alan Hoffman, David Jensen, and Ellis Johnson. The author is also indebted to an anonymous referee, who contributed valuable comments for the improvement of this paper.

REFERENCES

[1] P. G. ABRAHAMSON, *A Nested Decomposition Approach for Solving Staircase Linear Programs*, Ph.D. Dissertation, Department of Operations Research, Stanford University, Stanford, CA, 1983.

[2] J. F. BENDERS, *Partitioning procedures for solving mixed-variable programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[3] H. CROWDER, R. S. DEMBO, AND J. M. MULVEY, *On reporting computational experiments with mathematical software*, ACM Trans. Math. Software, 5 (1979), pp. 193–203.

[4] G. B. DANTZIG, *Solving Staircase Linear Programs by a Nested Block-Angular Method*, Report SOL 73-1, Department of Operations Research, Stanford University, Stanford, CA, 1973.

[5] J. J. DONGARRA AND E. GROSSE, *Distribution of mathematical software via electronic mail*, Comm. ACM, 30 (1987), pp. 403–407.

[6] G. B. DANTZIG AND P. WOLFE, *The decomposition algorithm for linear programming*, Econometrica, 29 (1961).

[7] R. ENTRIKEN, *Using MINOS as a Subroutine for Decomposition*, Report SOL 87-7, Department of Operations Research, Stanford University, Stanford, CA, 1987.

[8] ———, *A Parallel Decomposition Algorithm for Staircase Linear Programs*, Report SOL 88-21, Department of Operations Research, Stanford University, Stanford, CA, 1988.

[9] ———, *The Parallel Decomposition of Linear Programs*, Report SOL 89-17, Department of Operations Research, Stanford University, Stanford, CA, 1989.

[10] R. W. FOURER, *Solving staircase linear programs by the simplex method, 1: Inversion*, Math. Programming, 23 (1982), pp. 274–313.

[11] D. M. GAY, *Electronic mail distribution of linear programming test problems*, Math. Programming Soc. COAL Newsletter, December 1985.

[12] C. R. GLASSEY, *Nested decomposition and multi-stage linear programs*, Management Sci., 20 (1973), pp. 282–292.

[13] E. G. GOL'SHTEĬN, *A general approach to decomposition of optimization systems*, Soviet J. Comput. Systems Sci., 25 (1987), pp. 105–114 (translated from *Tekhnicheskaya Kibernetika*, 1 (1987), pp. 59–69).

[14] P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *Maintaining LU factors of a general sparse matrix*, Linear Algebra Appl., 88/89 (1987), pp. 239–270.

[15] K. L. HIEBERT, *An evaluation of mathematical software that solves systems of nonlinear equations*, ACM Trans. Math. Software, 8 (1982), pp. 5–20.

[16] J. K. HO, *Nested Decomposition of Large Scale Linear Programs with the Staircase Structure*, Ph.D. Dissertation, Department of Operations Research, Stanford University, Stanford, CA, 1974.

[17] J. K. HO AND E. LOUTE, *A set of staircase linear programming test problems*, Math. Programming, 20 (1981), pp. 245–250.

[18] ———, *An advanced implementation of the Dantzig-Wolfe decomposition algorithm for linear programming*, Math. Programming, 20 (1981), pp. 303–326.

[19] J. K. HO, T. C. LEE, AND R. P. SUNDARRAJ, *Decomposition of linear programs using parallel computation*, Math. Programming, 242 (1988), pp. 391–405.

[20] R. H. F. JACKSON, P. T. BOGGS, S. G. NASH, AND S. POWELL, *Report of the ad hoc committee to revise the guidelines for reporting computational experiments in mathematical programming*, Math. Programming Soc., COAL Newsletter, 1989.

[21] F. A. LOOTSMA AND K. M. RAGSDELL, *State-of-the-art in parallel nonlinear optimization*, Parallel Comput., 6 (1988), pp. 131–155.

[22] I. J. LUSTIG, *An Analysis of an Available Set of Linear Programming Test Problems*, Report SOL 87-11, Department of Operations Research, Stanford University, Stanford, CA, 1987.

[23] B. A. MURTAGH AND M. A. SAUNDERS, *MINOS* 5.1 *User's Guide (revised)*, Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA, 1987.

[24] C. B. STUNKEL, *Linear optimization via message-based parallel processing*, in Proc. of the 1988 Internat. Conference on Parallel Processing, Volume III, The Pennsylvania State University Press, University Park, PA, 1988, pp. 244–271.

[25] J. A. TOMLIN, *Mathematical Programming and Supercomputers*, Ketron Management Science, Inc., Mountain View, CA, 1987.

[26] R. J. WITTROCK, *Advances in a Nested Decomposition Algorithm for Solving Staircase Linear Programs*, Ph.D. Dissertation, Department of Operations Research, Stanford University, Stanford, CA, 1983.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 978–993, November 1996
© 1996 Society for Industrial and Applied Mathematics
005

ON THE SUPERLINEAR CONVERGENCE OF AN $O(n^3L)$ INTERIOR-POINT ALGORITHM FOR THE MONOTONE LCP*

KEVIN A. McSHANE†

Abstract. Karmarkar's partial updating scheme is applied to a polynomial superlinearly convergent algorithm for the monotone linear complementarity problem (LCP). This modification reduces the bound on the number of arithmetic operations required to achieve ϵ-optimality by a factor of $\sqrt{n}$. The complementarity gap sequence generated by the resulting algorithm converges q-quadratically to zero if the LCP is nondegenerate and has a strictly complementary optimal solution. In general, the convergence rate is q-superlinear if the iterates are assumed to converge to a strictly complementary optimal solution.

Key words. interior-point methods, linear complementarity problem, superlinear convergence, partial updating

AMS subject classifications. 90C05, 90C20, 90C33

1. Introduction.

We examine a partial updating variant of a polynomial, superlinearly convergent algorithm for the monotone linear complementarity problem (LCP): find $(x, s) \in \mathbf{R}^n \times \mathbf{R}^n$ which solves

$$\begin{array}{ll} \text{minimize} & x's \\ \text{subject to} & Mx + q = s, \\ & x, s \geq 0, \end{array}$$

where $M \in \mathbf{R}^{n\times n}$ is a positive semidefinite (not necessarily symmetric) matrix and $q \in \mathbf{R}^n$. We assume that the set

$$F^{++} \doteq \{(x, s) \in \mathbf{R}^n \,|Mx + q = s,\ x > 0, s > 0\}$$

is nonempty. The optimal objective function value for the LCP is zero; hence the optimal values for x and s are *complementary*, i.e., orthogonal and nonnegative. We call the LCP *nondegenerate* if it has a unique solution and *degenerate* otherwise. The LCP is a general class of problems which includes as special cases the linear programming (LP) and convex quadratic programming problems.

The modified LCP algorithm described below has the following properties.

- The bound on the number of arithmetic operations required to achieve ϵ-optimality is a factor of $\sqrt{n}$ smaller than that of the original algorithm. If the data are integer, the computational complexity of the variant is $O(n^3L)$, where L is the encoding length of the problem.
- The complementarity gap values (the objective function values) converge to zero at least q-linearly. If the LCP is nondegenerate with a strictly complementary optimal solution (i.e., (x^*, s^*) is feasible for the LCP, $x^{*\prime}s^* = 0$, and $x^* + s^* > 0$), the complementarity gap sequence converges to zero q-quadratically.
- If the problem is degenerate, the convergence rate is q-superlinear under the assumption that the variables converge to a strictly complementary optimal solution.

* Received by the editors November 16, 1994; accepted for publication (in revised form) July 10, 1995.

† 2537 Villanova Drive, Vienna, VA 22180 (mcshane@dgs.dgsys.com).

The use of the partial updating technique to improve the computational complexity of interior-point methods originated in Karmarkar's landmark paper [10]. Karmarkar showed that by updating the current factorization via rank-one updates (based upon approximate values for the variables) rather than performing a complete factorization in each iteration, the computational complexity of his algorithm could be reduced by a factor of $\sqrt{n}$. The technique has since been applied to a variety of other interior-point algorithms for LP and the LCP to reduce their complexity bounds (for example, [1]–[3], [5]–[8], [12]–[15], [17], [19]).

Our algorithm is based on what Gonzaga [9] refers to as the "longest step path-following" (LSPF) algorithm. The LSPF algorithm was first examined in Mizuno, Yoshise, and Kikuchi [16], where it was shown to require at most $O(\sqrt{n}L)$ iterations to solve the LCP with integer data. Each iteration requires $O(n^3)$ arithmetic operations, so the computational complexity of the algorithm is $O(n^{3.5}L)$. Mizuno and Todd [15] later proposed a partial updating variant of the LSPF algorithm which reduces the computational complexity to $O(n^3L)$. The algorithm we propose is different from the Mizuno–Todd algorithm because it uses a different partial updating scheme, does not use the auxiliary sequence of parameters μ_k (related to the complementarity gap), and has superlinear convergence properties. To the best of our knowledge, this is the first superlinear convergence result for an $O(n^3L)$ algorithm for LP or the LCP.

In [11], McShane showed that the complementarity gap sequence generated by the LSPF algorithm converges to zero q-quadratically if the LCP is nondegenerate and has a strictly complementary optimal solution. In the degenerate case, the convergence rate was shown to be q-superlinear, assuming the variables converge to a strictly complementary optimal solution. Bonnans and Gonzaga [4] followed by showing that the iterates do indeed converge in the degenerate case, establishing q-superlinear convergence assuming only the existence of a strictly complementary optimal solution. Recently in [9], Gonzaga has shown that the convergence rate is q-quadratic if a computationally trivial safeguard is applied.

Now, we provide a few notational conventions. The norm $\|\cdot\|$ refers to the l_2-norm unless denoted otherwise by a subscript, and $\log(\cdot)$ refers to the natural logarithm. Given a vector v, a subscript of i or j denotes a particular component, while a superscript (usually k) generally refers to an iteration number. A capital letter (say, X or V) version of a vector (x or v) refers to the diagonal matrix with the ith diagonal entry the ith component of the vector. We may also refer to X as $\operatorname{diag}(x)$. The symbol " $\doteq$ " assigns a definition to a quantity. Given a sequence of vectors $\{x^k\}$ and scalars $\{\pi_k\}$, the notation $x = O(\pi)$ means that there exists a constant C (dependent on problem data) such that $\|x^k\| \leq C\pi_k$ for all values of k.

2. Partial updating. The majority of the work required of each iteration in many interior-point algorithms is in the formation and factorization of a matrix of the form ADA', or the form $A + D$, where A is an $n \times m$ (or $n \times n$) matrix which does not change from iteration to iteration and D is a diagonal matrix which may change from iteration to iteration. These matrix operations have a computational complexity of $O(n^3)$. The remainder of the work for each iteration is typically $O(n^2)$. Karmarkar showed that by using an approximation $\tilde{D}$ for D, one could update the old factorization via rank-one updates—one update for each entry of $\tilde{D}$ which changed that iteration. Each rank-one update requires $O(n^2)$ arithmetic operations, so the number of arithmetic operations performed is $O(n^2K)$, where K is the number of rank-one updates done or, equivalently, the number of times entries of $\tilde{D}$ change.

Our algorithm maintains approximation vectors $\tilde{x}$ for x and $\tilde{s}$ for s which satisfy

$$\frac{1}{\delta} \leq \frac{x_i}{\tilde{x}_i} \leq \delta \quad \text{and} \quad \frac{1}{\delta} \leq \frac{s_i}{\tilde{s}_i} \leq \delta \tag{1}$$

for a fixed constant $\delta \geq 1$ by using the following generic scheme. We assume $\tilde{D}_{ii}$ is a function of $\tilde{x}_i$ and $\tilde{s}_i$ alone.

THE PARTIAL UPDATING PROCEDURE.

STEP 1: Given $(x^k, s^k) \in F^{++}$and $(\tilde{x}^k, \tilde{s}^k)$ which satisfy (1), compute $\tilde{D}^k$, x^{k+1}, and s^{k+1};

For $i = 1$ to n,

$$\text{if} \left(\frac{x_i^{k+1}}{\tilde{x}_i^k} \geq \delta \quad \text{or} \quad \frac{x_i^{k+1}}{\tilde{x}_i^k} \leq \frac{1}{\delta} \quad \text{or} \quad \frac{s_i^{k+1}}{\tilde{s}_i^k} \geq \delta \quad \text{or} \quad \frac{s_i^{k+1}}{\tilde{s}_i^k} \leq \frac{1}{\delta} \right)$$

$$\begin{array}{lll} \text{then set} & \tilde{x}_i^{k+1} = x_i^{k+1} & \text{and} \quad \tilde{s}_i^{k+1} = s_i^{k+1}; \\ \text{else set} & \tilde{x}_i^{k+1} = \tilde{x}_i^k & \text{and} \quad \tilde{s}_i^{k+1} = \tilde{s}_i^k; \end{array}$$

STEP 2: If the termination criteria are not met, set $k = k + 1$, perform a rank-one update to the old factorization for each changed entry of $\tilde{D}$, go to STEP 1.

The following lemma appears as Lemma 3.3 in Anstreicher [1], and the subsequent theorem is based on ideas contained in the proofs of Lemma 3.1, Corollary 3.2, and Theorem 3.4 in [1].

LEMMA 2.1. *If* $0 < a < 1$ *and* $b \geq a$, *then* $|\log(b)| \leq \frac{|b-1|\log(a)}{a-1}$.

Proof. The lemma follows from the fact that the function

$$f(x) = \begin{cases} \log(x)/(x-1) & \text{for} \quad x > 0, x \neq 1, \\ 1 & \text{for} \quad x = 1 \end{cases}$$

is continuous and monotonically decreasing. □

THEOREM 2.2. *Suppose an algorithm for the LCP generates a sequence of feasible interior points* (x^0, s^0), $(x^1, s^1), \ldots, (x^K, s^K)$ *via* $x^{k+1} = x^k + d_x^k$ *and* $s^{k+1} = s^k + d_s^k$. *Suppose also that the factorization used to compute the search directions is efficiently updated by using the partial updating procedure with some* $\delta > 1$. *Let* T *represent the number of rank-one updates performed. Then*

$$T \leq \frac{\delta}{\delta - 1} \sum_{k=0}^{K-1} \left(\left\| (X^k)^{-1} d_x^k \right\|_1 + \left\| (S^k)^{-1} d_s^k \right\|_1 \right).$$

Proof. Define T^i as the ordered set of iteration indices k such that $\tilde{x}_i^k \neq \tilde{x}_i^{k-1}$. Insert 0 at the beginning of T^i. Let a and b, with $a < b$, be two consecutive elements of T^i. Note that $\tilde{x}_i^{b-1} = \tilde{x}_i^a = x_i^a$ and $\tilde{s}_i^{b-1} = \tilde{s}_i^a = s_i^a$ since $\tilde{x}_i$ and $\tilde{s}_i$ are not updated during the iterations between a and b. Since $\tilde{x}_i^b \neq \tilde{x}_i^{b-1}$, we have

$$\frac{x_i^b}{x_i^a} \geq \delta \quad \text{or} \quad \frac{x_i^b}{x_i^a} \leq \frac{1}{\delta} \quad \text{or} \quad \frac{s_i^b}{s_i^a} \geq \delta \quad \text{or} \quad \frac{s_i^b}{s_i^a} \leq \frac{1}{\delta}.$$

In any case,

$$\log(\delta) \leq \left| \log\left(\frac{x_i^b}{x_i^a} \right) \right| + \left| \log\left(\frac{s_i^b}{s_i^a} \right) \right|,$$

so

$$\begin{aligned}\log(\delta) &\le \left|\log\left(\frac{x_i^{a+1}}{x_i^a}\frac{x_i^{a+2}}{x_i^{a+1}}\cdots\frac{x_i^b}{x_i^{b-1}}\right)\right| + \left|\log\left(\frac{s_i^{a+1}}{s_i^a}\frac{s_i^{a+2}}{s_i^{a+1}}\cdots\frac{s_i^b}{s_i^{b-1}}\right)\right| \\ &= \left|\sum_{k=a}^{b-1}\log\left(\frac{x_i^{k+1}}{x_i^k}\right)\right| + \left|\sum_{k=a}^{b-1}\log\left(\frac{s_i^{k+1}}{s_i^k}\right)\right| \\ &\le \sum_{k=a}^{b-1}\left(\left|\log\left(\frac{x_i^{k+1}}{x_i^k}\right)\right| + \left|\log\left(\frac{s_i^{k+1}}{s_i^k}\right)\right|\right).\end{aligned}$$

Define

$$\begin{aligned}q_i^k &\doteq \max\left\{\frac{x_i^{k+1}}{x_i^k}, \frac{1}{\delta}\right\}, \\ r_i^k &\doteq \max\left\{\frac{s_i^{k+1}}{s_i^k}, \frac{1}{\delta}\right\}.\end{aligned}$$

Since $\log(\delta) = \left|\log\left(\frac{1}{\delta}\right)\right|$, we have

$$\log(\delta) \le \sum_{k=a}^{b-1}\left(\left|\log(q_i^k)\right| + \left|\log(r_i^k)\right|\right).$$

Now, sum over all values for $a \in T^i$ and over all values for i:

$$T\log(\delta) \le \sum_{i=1}^{n}\sum_{k=0}^{K-1}\left(\left|\log(q_i^k)\right| + \left|\log(r_i^k)\right|\right),$$

where T is the total number of updates done to $\tilde{x}$ (or $\tilde{s}$). Applying Lemma 2.1 twice, with $1/\delta$ for a and q_i^k for b, then with $1/\delta$ for a and r_i^k for b, yields

$$\begin{aligned}T\log(\delta) &\le \sum_{i=1}^{n}\sum_{k=0}^{K-1}\frac{\delta\log(\delta)}{\delta-1}\left(\left|1-q_i^k\right| + \left|1-r_i^k\right|\right) \\ &\le \sum_{i=1}^{n}\sum_{k=0}^{K-1}\frac{\delta\log(\delta)}{\delta-1}\left(\left|1-\frac{x_i^{k+1}}{x_i^k}\right| + \left|1-\frac{s_i^{k+1}}{s_i^k}\right|\right) \\ &= \frac{\delta\log(\delta)}{\delta-1}\sum_{k=0}^{K-1}\left(\left\|(X^k)^{-1}d_x^k\right\|_1 + \left\|(S^k)^{-1}d_s^k\right\|_1\right),\end{aligned}$$

which completes the proof. □

3. The algorithm. The search directions we use are based on the LSPF directions and are quite similar to the Mizuno–Todd directions. Given $(x,s) \in F^{++}$ and approximation vectors $\tilde{x}$ and $\tilde{s}$, define

$$\begin{aligned}\Delta &\doteq \frac{x's}{n}, \\ \bar{v} &\doteq -(\tilde{X}\tilde{S})^{-1/2}(XSe - \Delta e), \\ \hat{v} &\doteq -(\tilde{X}\tilde{S})^{-1/2}(XS)e, \\ \tilde{D} &\doteq \tilde{X}^{1/2}\tilde{S}^{-1/2}.\end{aligned}$$

Here, e is the n-vector of ones. Our directions are a convex combination of the centering and affine directions

$$\begin{pmatrix} \bar{d}_x \\ \bar{d}_s \end{pmatrix} \doteq \begin{pmatrix} M & -I \\ \tilde{D}^{-1} & \tilde{D} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \bar{v} \end{pmatrix}, \tag{2}$$

$$\begin{pmatrix} \hat{d}_x \\ \hat{d}_s \end{pmatrix} \doteq \begin{pmatrix} M & -I \\ \tilde{D}^{-1} & \tilde{D} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \hat{v} \end{pmatrix}; \tag{3}$$

that is, our directions are of the form

$$\begin{pmatrix} d_x(\theta) \\ d_s(\theta) \end{pmatrix} = \theta \begin{pmatrix} \bar{d}_x \\ \bar{d}_s \end{pmatrix} + (1-\theta) \begin{pmatrix} \hat{d}_x \\ \hat{d}_s \end{pmatrix}, \tag{4}$$

where $\theta \in [0,1]$. It is easily verified that the inverse matrix in (2) exists. We also define

$$\begin{aligned} D_x(\theta) &\doteq \operatorname{diag}(d_x(\theta)), \\ D_s(\theta) &\doteq \operatorname{diag}(d_s(\theta)), \\ x(\theta) &\doteq x + d_x(\theta), \\ s(\theta) &\doteq s + d_s(\theta), \\ \Delta(\theta) &\doteq \frac{x(\theta)'s(\theta)}{n}, \\ \theta^* &\doteq 1 - \frac{1}{5\sqrt{n}}. \end{aligned}$$

Note that from (3),

$$M\hat{d}_x = \hat{d}_s \tag{5}$$

and

$$\hat{d}_s = \tilde{D}^{-1}\hat{v} - \tilde{D}^{-2}\hat{d}_x.$$

So, by substitution, $\hat{d}_x$ can be formed by solving

$$(M + \tilde{D}^{-2})\hat{d}_x = \tilde{D}^{-1}\hat{v}, \tag{6}$$

and then $\hat{d}_s$ can be computed from (5). A similar argument holds for $\bar{d}_x$ and $\bar{d}_s$, so the partial updating scheme can be efficiently applied to a factorization of the matrix $M + \tilde{D}^{-2}$.

Like the LSPF algorithm, our algorithm is a path-following algorithm. All iterates (x^k, s^k) must satisfy the approximately centered condition

$$\left\| X^k S^k e - \Delta_k e \right\| \leq \beta \Delta_k, \tag{7}$$

where $\beta \in (0,1)$ is a fixed parameter.

We now propose our algorithm.

THE LCP ALGORITHM.

STEP 0: Set $\beta = .2$, $\hat{\beta} = \beta/(2+\beta)$, $\delta = 1.01$, $\theta^* = 1 - 1/(5\sqrt{n})$, and $k = 0$. Given $\varepsilon > 0$ and $(x^0, s^0) \in F^{++}$ which satisfies (7), set $\tilde{x}^0 = x^0$ and $\tilde{s}^0 = s^0$;

STEP 1: Given $(x^k, s^k) \in F^{++}$ which satisfies (7), and $\tilde{x}^k$ and $\tilde{s}^k$ generated by the partial updating procedure, compute $\bar{d}_x^k$, $\bar{d}_s^k$, $\hat{d}_x^k$, and $\hat{d}_s^k$ using the method in (6). Calculate the roots of the quartic polynomial

$$g(\theta) \doteq \left\|X^k(\theta)S^k(\theta)e - \theta\Delta_k e\right\|^2 - \hat{\beta}^2\theta^2(\Delta_k)^2. \tag{8}$$

If g has no roots in the interval $(0, \theta^*]$, set $\theta_k = 0$ and compute the search directions from (4). The updated points will be optimal for the LCP. Otherwise, let θ_k be the largest root of g in the interval $(0, \theta^*]$;

STEP 2: Set $x^{k+1} = x^k(\theta_k)$ and $s^{k+1} = s^k(\theta_k)$. If $(x^{k+1})'(s^{k+1}) \leq \varepsilon$ then terminate, else perform the partial updating procedure, set $k = k + 1$, then go to STEP 1.

4. Analysis of the algorithm. The function $g(\theta)$ in (8) is similar to the functions used to compute quantities equivalent to θ in the Mizuno–Todd algorithm and the LSPF algorithm as described in Gonzaga [9]. These papers replace Δ_k in (7) with a related parameter μ_k. The relationship between μ_k and Δ_k is described in detail in [9]. We choose not to monitor μ but instead to monitor Δ, in order to exploit the orthogonality of e and $XSe - \Delta e$. For our purposes, the function $g(\theta)$ is simply an approximation to the updated approximate centering condition

$$\left\|X^k(\theta)S^k(\theta)e - \Delta_k(\theta)e\right\| \leq \beta\Delta_k(\theta)$$

which simplifies the analysis of the algorithm. In the analysis below, we suppress the iteration index to simplify the notation.

THEOREM 4.1. *If during an iteration of the LCP algorithm, $\theta \in [0, 1]$ is chosen so that $g(\theta) \leq 0$, then $\|X(\theta)S(\theta)e - \Delta(\theta)e\| \leq \beta\Delta(\theta)$.*

Proof. Note that

$$\|X(\theta)S(\theta)e - \Delta(\theta)e\| \leq \|X(\theta)S(\theta)e - \theta\Delta e\| + \|(\theta\Delta - \Delta(\theta))e\|.$$

But $g(\theta) \leq 0$ implies $\|X(\theta)S(\theta)e - \theta\Delta e\| \leq \hat{\beta}\theta\Delta$, so

$$\|X(\theta)S(\theta)e - \Delta(\theta)e\| \leq \hat{\beta}\theta\Delta + |\theta\Delta - \Delta(\theta)|\sqrt{n}. \tag{9}$$

Following Gonzaga [9], define

$$h \doteq X(\theta)S(\theta)e - \theta\Delta e;$$

hence, $\|h\| \leq \hat{\beta}\theta\Delta$. If we left-multiply the above expression by $(1/n)e'$, the result is

$$\Delta(\theta) - \theta\Delta = \frac{1}{n}e'h; \tag{10}$$

thus,

$$|\Delta(\theta) - \theta\Delta| \leq \frac{1}{n}\|e\|\,\|h\| \leq \frac{1}{\sqrt{n}}\hat{\beta}\theta\Delta. \tag{11}$$

Also, from (10),

$$\theta\Delta = \Delta(\theta) - \frac{1}{n}e'h \leq \Delta(\theta) + \frac{1}{\sqrt{n}}\hat{\beta}\theta\Delta \leq \Delta(\theta) + \hat{\beta}\theta\Delta,$$

so therefore

$$\theta\Delta \leq \frac{1}{1-\hat{\beta}}\Delta(\theta). \tag{12}$$

If we apply inequalities (11) and (12) to (9), we see that

$$\|X(\theta)S(\theta)e - \Delta(\theta)e\| \leq \frac{2\hat{\beta}}{1-\hat{\beta}}\Delta(\theta) = \beta\Delta(\theta),$$

which completes the proof. □

The goal of this section is to show that the LCP algorithm is well defined. The following three lemmas are similar to results used in the analysis of a variety of path-following $O(n^3L)$ interior-point algorithms, [8] and [18] for example. The proofs are included for the sake of clarity.

LEMMA 4.2. *During each iteration of the LCP algorithm, the following relationships hold:*

(a) $\left\|\tilde{D}(S-\tilde{S})\right\| \leq \delta(\delta-1)\sqrt{1+\beta}\sqrt{\Delta}$,

(b) $\left\|\tilde{D}^{-1}(X-\tilde{X})\right\| \leq \delta(\delta-1)\sqrt{1+\beta}\sqrt{\Delta}$.

Proof. Observe that

$$\begin{aligned}\left\|\tilde{D}(S-\tilde{S})\right\| &= \left\|(\tilde{X}\tilde{S})^{1/2}(\tilde{S}^{-1}S-I)\right\| \\ &\leq \left\|(\tilde{X}\tilde{S})^{1/2}\right\|\left\|\tilde{S}^{-1}S-I\right\|. \end{aligned} \tag{13}$$

But (7) implies that

$$(1-\beta)\Delta \leq x_i s_i \leq (1+\beta)\Delta \tag{14}$$

so, using (1),

$$\tilde{x}_i\tilde{s}_i \leq \delta^2(1+\beta)\Delta.$$

Thus

$$\left\|(\tilde{X}\tilde{S})^{1/2}\right\| \leq \delta\sqrt{1+\beta}\sqrt{\Delta}.$$

Since (1) implies that

$$\left\|\tilde{S}^{-1}S-I\right\| \leq \delta-1,$$

with (13) we have

$$\left\|\tilde{D}(S-\tilde{S})\right\| \leq \delta(\delta-1)\sqrt{1+\beta}\sqrt{\Delta}.$$

Inequality (b) can be shown similarly. □

Define

$$\tilde{v}(\theta) \doteq \theta\bar{v} + (1-\theta)\hat{v}. \tag{15}$$

From (2)–(4) we see that

$$\tilde{D}^{-1}d_x(\theta) + \tilde{D}d_s(\theta) = \tilde{v}(\theta). \tag{16}$$

LEMMA 4.3. *During each iteration of the LCP algorithm, the following relationships hold for any* $\theta \in [0,1]$*:*

(a) $\left\|\tilde{D}^{-1}d_x\right\|^2 + \left\|\tilde{D}d_s\right\|^2 \leq \|\tilde{v}\|^2$,

(b) $\left\|\tilde{D}^{-1}d_x\right\| + \left\|\tilde{D}d_s\right\| \leq \sqrt{2}\,\|\tilde{v}\|$,

(c) $\|D_xD_se\| \leq \frac{1}{2}\|\tilde{v}\|^2$,

(d) $\left\|(X-\tilde{X})d_s + (S-\tilde{S})d_x\right\| \leq \sqrt{2}\delta(\delta-1)\sqrt{1+\beta}\sqrt{\Delta}\,\|\tilde{v}\|$,

where $d_x \doteq d_x(\theta)$, $d_s \doteq d_s(\theta)$, $D_x \doteq D_x(\theta)$, $D_s \doteq D_s(\theta)$, *and* $\tilde{v} \doteq \tilde{v}(\theta)$.

Proof. We first prove (a). From (16),

$$\left\|\tilde{D}^{-1}d_x + \tilde{D}d_s\right\|^2 = \|\tilde{v}\|^2,$$

and expanding the leftmost norm and rearranging yields

$$\left\|\tilde{D}^{-1}d_x\right\|^2 + \left\|\tilde{D}d_s\right\|^2 \leq \|\tilde{v}\|^2 - d_x'd_s.$$

But from (2)–(4),

$$d_x'd_s = d_x'Md_x \geq 0$$

because M is positive semidefinite. Therefore,

$$\left\|\tilde{D}^{-1}d_x\right\|^2 + \left\|\tilde{D}d_s\right\|^2 \leq \|\tilde{v}\|^2,$$

which completes the proof of (a).

Inequality (b) follows from (a) and the fact that

$$\left(\left\|\tilde{D}^{-1}d_x\right\| + \left\|\tilde{D}d_s\right\|\right)^2 \leq 2\left(\left\|\tilde{D}^{-1}d_x\right\|^2 + \left\|\tilde{D}d_s\right\|^2\right).$$

To show (c), note that

$$\begin{aligned}
\|D_xD_se\|^2 &= (e'D_x^2\tilde{D}^{-2})(\tilde{D}^2D_s^2e) \\
&\leq \left\|\tilde{D}^{-2}D_x^2e\right\| \left\|\tilde{D}^2D_s^2e\right\| \\
&\leq \left\|\tilde{D}^{-2}D_x^2e\right\|_1 \left\|\tilde{D}^2D_s^2e\right\|_1 \\
&= \left\|\tilde{D}^{-1}D_xe\right\|^2 \left\|\tilde{D}D_se\right\|^2 \\
&\leq \frac{1}{4}\left(\left\|\tilde{D}^{-1}d_x\right\|^2 + \left\|\tilde{D}d_s\right\|^2\right)^2
\end{aligned}$$

by the arithmetic–geometric mean inequality. Inequality (c) follows by applying (a) to this expression.

Finally,

$$\begin{aligned}
\left\|(X-\tilde{X})d_s + (S-\tilde{S})d_x\right\| &\leq \left\|(X-\tilde{S})d_s\right\| + \left\|(S-\tilde{S})d_x\right\| \\
&= \left\|(X-\tilde{X})\tilde{D}^{-1}\tilde{D}d_s\right\| + \left\|(S-\tilde{S})\tilde{D}\tilde{D}^{-1}d_x\right\| \\
&\leq \left\|\tilde{D}^{-1}(X-\tilde{X})\right\| \left\|\tilde{D}d_s\right\| + \left\|\tilde{D}(S-\tilde{S})\right\| \left\|\tilde{D}^{-1}d_x\right\|.
\end{aligned}$$

Inequality (d) follows by applying Lemma 4.2, then Lemma 4.3(b). □

LEMMA 4.4. *During each iteration of the LCP algorithm, the following relationships hold for any* $\theta \in [0,1]$:

(a) $\|\tilde{v}(\theta)\|^2 \leq \delta^2 \left(\beta^2/(1-\beta) + (1-\theta)^2 n\right) \Delta$,

(b) $\|\tilde{v}(\theta)\| \leq \delta \left(\beta/\sqrt{1-\beta} + (1-\theta)\sqrt{n}\right) \sqrt{\Delta}$.

Proof. From (15),

$$\begin{aligned}\tilde{v}(\theta) &= \theta\bar{v} + (1-\theta)\hat{v} \\ &= -(\tilde{X}\tilde{S})^{-1/2}(XS)^{1/2}\left(\theta(XS)^{-1/2}(XSe - \Delta e) + (1-\theta)(XS)^{1/2}e\right);\end{aligned}$$

therefore (using(1)),

$$\|\tilde{v}(\theta)\|^2 \leq \delta^2 \left\|\theta(XS)^{-1/2}(XSe - \Delta e) + (1-\theta)(XS)^{1/2}e\right\|^2.$$

Since the two terms above are orthogonal, and since $\theta \in [0,1]$, we have

$$\begin{aligned}\|\tilde{v}(\theta)\|^2 &\leq \delta^2 \left(\left\|(XS)^{-1/2}(XSe - \Delta e)\right\|^2 + (1-\theta)^2 \left\|(XS)^{1/2}e\right\|^2\right) \\ &\leq \delta^2 \left(\left\|(XS)^{-1}\right\| \|XSe - \Delta e\|^2 + (1-\theta)^2 n\Delta\right) \\ &\leq \delta^2 \left(\frac{1}{(1-\beta)\Delta}\beta^2\Delta^2 + (1-\theta)^2 n\Delta\right) \text{ (by (14) and (7))} \\ &= \delta^2 \left(\frac{\beta^2}{1-\beta} + (1-\theta)^2 n\right)\Delta,\end{aligned}$$

which proves (a). Inequality (b) follows trivially from (a). □

THEOREM 4.5. *During STEP* 1 *of each iteration of the LCP algorithm, for any* $\theta \in [\theta^*, 1]$, $(x(\theta), s(\theta)) \in F^{++}$. *Also,* $g(\theta^*) < 0$.

Proof. Define $d_x \doteq d_x(\theta)$, $d_s \doteq d_s(\theta)$, $D_x \doteq D_x(\theta)$, $D_s \doteq D_s(\theta)$, and $\tilde{v} \doteq \tilde{v}(\theta)$.

If we multiply equation (16) by $(\tilde{X}\tilde{S})^{1/2}$, the result is

$$\tilde{X}d_s + \tilde{S}d_x = (\tilde{X}\tilde{S})^{1/2}\tilde{v} = \theta\Delta e - XSe. \tag{17}$$

Consider the expression in the first term in the definition of $g(\theta)$:

$$\begin{aligned}X(\theta)S(\theta)e - \theta\Delta e &= XSe + Xd_s + Sd_x + D_xD_se - \theta\Delta e \\ &= XSe + \tilde{X}d_s + \tilde{S}d_x + D_xD_se - \theta\Delta e + (X-\tilde{X})d_s + (S-\tilde{S})d_x \\ &= D_xD_se + (X-\tilde{X})d_s + (S-\tilde{S})d_x \text{ (by (17)).} \qquad (18)\end{aligned}$$

Therefore,

$$\|X(\theta)S(\theta)e - \theta\Delta e\| \leq \|D_xD_se\| + \left\|(X-\tilde{X})d_s + (S-\tilde{S})d_x\right\|,$$

so applying Lemmas 4.3 and 4.4 results in

$$\begin{aligned}\|X(\theta)S(\theta)e - \theta\Delta e\| &\leq \frac{1}{2}\|\tilde{v}\|^2 + \sqrt{2}\delta(\delta-1)\sqrt{1+\beta}\sqrt{\Delta}\,\|\tilde{v}\| \\ &\leq \frac{\delta^2}{2}\left(\frac{\beta^2}{1-\beta} + (1-\theta)^2 n\right)\Delta + \alpha\left(\frac{\beta}{\sqrt{1-\beta}} + (1-\theta)\sqrt{n}\right)\Delta, \qquad (19)\end{aligned}$$

where $\alpha \doteq \sqrt{2}\delta^2(\delta-1)\sqrt{1+\beta}$.

Now, for any $\theta \in [\theta^*, 1]$ and with the choices $\beta = .2$, $\hat{\beta} = \beta/(2+\beta)$, $\delta = 1.01$, and $\theta^* = 1 - 1/(5\sqrt{n})$, it is easily verified that (19) is bounded above by $.06\Delta$. Therefore, for each index i,

$$-.06\Delta \leq x_i(\theta)s_i(\theta) - \theta\Delta \leq .06\Delta$$

so

$$-.06\Delta + \theta\Delta \leq x_i(\theta)s_i(\theta).$$

Moreover, since $\theta^* \geq .8$,

$$.7\Delta \leq x_i(\theta)s_i(\theta).$$

Consequently, by continuity, $(x(\theta), s(\theta)) \in F^{++}$ for any $\theta \in [\theta^*, 1]$. It is easily verified that $.06 < \hat{\beta}\theta^*$, which implies $g(\theta^*) < 0$. □

COROLLARY 4.6. *The LCP algorithm is well defined.*

Proof. An argument analogous to that in Theorem 4.5 shows that as long as $g(\theta)$ remains less than zero, $(x(\theta), s(\theta)) \in F^{++}$ as θ decreases from the value θ^*. If g has no roots between 0 and θ^*, by continuity, $g(0) \leq 0$ and $(x(0), s(0)) \geq 0$. But from the definition of g, $g(0) \leq 0$ implies that $\|X(0)S(0)e\| = 0$; therefore $(x(0), s(0))$ solves the LCP. Otherwise, if g has a root in the interval $[0, \theta^*]$, the choice for θ as the largest root in the interval results in strictly feasible updated iterates which, by Theorem 4.1, are approximately centered. □

5. Complexity of the algorithm.

THEOREM 5.1. *Given a value for $\varepsilon \in (0,1)$, if $x^{0\prime}s^0 \leq 1/\varepsilon$, then $x^{K\prime}s^K \leq \varepsilon$ after at most $K = O(\sqrt{n}\log(1/\varepsilon))$ iterations. The number of arithmetic operations required is $O(n^3\log(1/\varepsilon))$. If the data are integer (or rational), the algorithm requires $O(\sqrt{n}L)$ iterations and $O(n^3L)$ arithmetic operations, where L is the encoding length of the problem.*

Proof. For each iteration k, from (10),

$$\Delta_k = \Delta_{k-1}(\theta_{k-1}) = \frac{1}{n}e'h^{k-1} + \theta_{k-1}\Delta_{k-1}.$$

Thus with (11) we have

$$\Delta_k \leq \left(1 + \frac{\hat{\beta}}{\sqrt{n}}\right)\theta_{k-1}\Delta_{k-1}.$$

If we multiply this expression by n the result is

$$(x^k)'(s^k) \leq \left(1 + \frac{\hat{\beta}}{\sqrt{n}}\right)\theta_{k-1}(x^{k-1})'(s^{k-1}) \tag{20}$$

$$\leq (x^0)'(s^0)\prod_{j=0}^{k-1}\left(\left(1 + \frac{1}{10\sqrt{n}}\right)\theta_j\right). \tag{21}$$

But

$$\theta_j \leq \theta^* = 1 - \frac{1}{5\sqrt{n}};$$

thus,

$$(x^k)'(s^k) \leq (x^0)'(s^0) \prod_{j=0}^{k-1} \left(\left(1 + \frac{1}{10\sqrt{n}}\right) \left(1 - \frac{1}{5\sqrt{n}}\right) \right)$$

$$\leq \left(1 - \frac{1}{10\sqrt{n}}\right)^k (x^0)'(s^0). \tag{22}$$

Define K_1 to be the largest value for k such that $(x^k)'(s^k) > \varepsilon$. Similarly, define K_2 as the largest k such that (21) $> \varepsilon$, and define K_3 as the largest k such that (22) $> \varepsilon$. A standard argument shows that (22) implies $K_3 = O(\sqrt{n}\log(1/\varepsilon))$. From these definitions it is evident that

$$K_1 \leq K_2 \leq K_3 = O\left(\sqrt{n}\log(1/\varepsilon)\right). \tag{23}$$

Define $d_x^k \doteq d_x^k(\theta_k)$ and $d_s^k \doteq d_s^k(\theta_k)$. To determine the algorithm's computational complexity, we appeal to Theorem 2.2. The number of rank-one updates required before the last iteration is

$$T \leq \frac{\delta}{\delta - 1} \sum_{k=0}^{K_1 - 1} \left(\left\| (X^k)^{-1} d_x^k \right\|_1 + \left\| (S^k)^{-1} d_s^k \right\|_1 \right)$$

$$\leq \frac{\delta\sqrt{n}}{\delta - 1} \sum_{k=0}^{K_1 - 1} \left(\left\| (X^k)^{-1} d_x^k \right\|_2 + \left\| (S^k)^{-1} d_s^k \right\|_2 \right). \tag{24}$$

Note that

$$\left\| (X^k)^{-1} d_x^k \right\| = \left\| (X^k)^{-1} \tilde{X}^k (\tilde{X}^k \tilde{S}^k)^{-1/2} (\tilde{D}^k)^{-1} d_x^k \right\|$$

$$\leq \left\| (X^k)^{-1} \tilde{X}^k \right\| \left\| (\tilde{X}^k \tilde{S}^k)^{-1/2} \right\| \left\| (\tilde{D}^k)^{-1} d_x^k \right\|,$$

so, using (1), (7), and Lemma 4.3(a),

$$\left\| (X^k)^{-1} d_x^k \right\| \leq (\delta) \left(\frac{\delta}{\sqrt{(1-\beta)\,\Delta_k}} \right) \left\| \tilde{v}^k(\theta) \right\|.$$

Further, by Lemma 4.4,

$$\left\| (X^k)^{-1} d_x^k \right\| \leq \left(\frac{\delta^3}{\sqrt{1-\beta}} \right) \left(\frac{\beta}{\sqrt{1-\beta}} + (1 - \theta_k)\sqrt{n} \right).$$

A similar argument shows that the same bound applies to $\left\| (S^k)^{-1} d_s^k \right\|$. Thus, from (24),

$$T \leq \frac{2\delta^4\sqrt{n}}{(\delta - 1)\sqrt{1-\beta}} \sum_{k=0}^{K_1 - 1} \left(\frac{\beta}{\sqrt{1-\beta}} + (1 - \theta_k)\sqrt{n} \right).$$

Using the well-known inequality $x \leq -\log(1 - x)$ for all $x \in [0, 1)$, we see that $(1 - \theta_k) \leq -\log(\theta_k)$; therefore,

$$T \leq \frac{2\delta^4\sqrt{n}}{(\delta - 1)\sqrt{1-\beta}} \sum_{k=0}^{K_1 - 1} \left(\frac{\beta}{\sqrt{1-\beta}} - \log(\theta_k)\sqrt{n} \right)$$

$$\leq \frac{2\delta^4\beta\sqrt{n}}{(\delta-1)(1-\beta)}K_1 - \frac{2\delta^4 n}{(\delta-1)\sqrt{1-\beta}}\sum_{k=0}^{K_1-1}\log(\theta_k)$$

$$\leq O\left(n\log(1/\varepsilon)\right) - \frac{2\delta^4 n}{(\delta-1)\sqrt{1-\beta}}\sum_{k=0}^{K_1-1}\log(\theta_k) \quad \text{(by (23))}. \tag{25}$$

Now, from the definition of K_2,

$$(x^0)'(s^0)\prod_{j=0}^{K_2-1}\left(\left(1+\frac{1}{10\sqrt{n}}\right)\theta_j\right) > \varepsilon.$$

Therefore,

$$\log\left((x^0)'(s^0)\right) + K_2\log\left(1+\frac{1}{10\sqrt{n}}\right) + \sum_{j=0}^{K_2-1}\log(\theta_j) > \log(\varepsilon);$$

hence, by rearrangement and using the fact that $\log(1+x) \leq x$ for $x > -1$, we see that

$$-\sum_{j=0}^{K_2-1}\log(\theta_j) < \log\left((x^0)'(s^0)\right) - \log(\varepsilon) + \frac{K_2}{10\sqrt{n}}.$$

Now, with the hypotheses of the theorem and (23) we have

$$-\sum_{j=0}^{K_2-1}\log(\theta_j) = O\left(\log(1/\varepsilon)\right).$$

Since $K_2 \geq K_1$, substituting this into (25) implies that the number of updates required before the last iteration is

$$T = O\left(n\log(1/\varepsilon)\right).$$

Since the last iteration requires $O(n^3)$ arithmetic operations, and since each rank-one update requires $O(n^2)$ arithmetic operations, the total number of arithmetic operations required is $O\left(n^3\log(1/\varepsilon)\right)$.

In the integer case, it is well known that $\varepsilon = 2^{-2L}$ is sufficient to solve the LCP, from which the remainder of the theorem follows. □

THEOREM 5.2. *If the iterates (x^k, s^k) of the LCP algorithm converge to a strictly complementary optimal solution (x^*, s^*), then $\{(x^k)'(s^k)\} \to 0$ q-superlinearly. Moreover, after a finite number of iterations, every subsequent iteration will involve a complete factorization.*

Proof. For each iteration k, define $d_x^k \doteq d_x^k(\theta_k)$, $d_s^k \doteq d_s^k(\theta_k)$, $D_x^k \doteq D_x^k(\theta_k)$, $D_s^k \doteq D_s^k(\theta_k)$, and $\tilde{v}^k \doteq \tilde{v}^k(\theta_k)$. If the algorithm does not terminate finitely, $g(\theta_k) = 0$ for each iteration k, so from the definition of g,

$$\left\|X^k(\theta_k)S^k(\theta_k)e - \theta_k\Delta_k e\right\| = \hat{\beta}\theta_k\Delta_k.$$

That is, by (18),

$$\left\|D_x^k D_s^k e + (X^k - \tilde{X}^k)d_s^k + (S^k - \tilde{S}^k)d_x^k\right\| = \hat{\beta}\theta_k\Delta_k$$

or, by rearrangement,

$$\theta_k = \frac{\left\| D_x^k D_s^k e + (X^k - \tilde{X}^k) d_s^k + (S^k - \tilde{S}^k) d_x^k \right\|}{\hat{\beta} \Delta_k}. \tag{26}$$

To show q-superlinear convergence of the complementarity gap sequence to zero, (20) implies that it is sufficient to show that $\theta_k \to 0$ as $k \to \infty$. Due to (26), it is enough to show that

$$\frac{\left\| D_x^k D_s^k e \right\|}{\Delta_k} \to 0 \tag{27}$$

and

$$\frac{\left\| (X^k - \tilde{X}^k) d_s^k + (S^k - \tilde{S}^k) d_x^k \right\|}{\Delta_k} \to 0. \tag{28}$$

We first show (27). The proof is similar to that of Theorem 4.4 in [11].

Consider component i of x^k and s^k. Assume without loss of generality that $x_i^k \to x_i^* > 0$ and $s_i^k \to 0$ as $k \to \infty$. Since $x_i^k + (d_x^k)_i \to x_i^*$ implies $x_i^k / x_i^* + (d_x^k)_i / x_i^* \to 1$ as $k \to \infty$, it follows that

$$(d_x^k)_i \to 0 \text{ as } k \to \infty. \tag{29}$$

Let K be such that for $k \geq K$,

$$\frac{1}{2} x_i^* \leq x_i^k \leq 2 x_i^*. \tag{30}$$

For $k \geq K$, (7) and (30), followed by (1), imply

$$\frac{1 - \beta}{2\delta x_i^*} \leq \frac{\tilde{s}_i^k}{\Delta_k} \leq \frac{2(1+\beta)\delta}{x_i^*}. \tag{31}$$

Now, Lemmas 4.3(a) and 4.4(b) imply

$$\left\| \tilde{D}^k d_s^k \right\| \leq \gamma \sqrt{\Delta_k},$$

where $\gamma \doteq \delta \left(\beta / \sqrt{1 - \beta} + \sqrt{n} \right)$; therefore,

$$\left| \frac{\sqrt{\tilde{x}_i^k}}{\sqrt{\tilde{s}_i^k}} (d_s^k)_i \right| \leq \gamma \sqrt{\Delta_k}.$$

Hence (using (30), (31), and (1)),

$$\left| \frac{(d_s^k)_i}{\Delta_k} \right| \leq \frac{\gamma \sqrt{\tilde{s}_i^k}}{\sqrt{\tilde{x}_i^k} \sqrt{\Delta_k}} \leq \frac{2\delta\gamma\sqrt{1+\beta}}{x_i^*}. \tag{32}$$

Since a similar argument holds for each i, (29) and (32) imply (27).

Next, we prove (28). Again, consider component i and assume that $x_i^k \to x_i^* > 0$ and $s_i^k \to 0$ as $k \to \infty$. It is evident that

$$\tilde{s}_i^k \to 0 \text{ as } k \to \infty$$

because for any fixed index m, since $s_i^k \to 0$, there is a smallest index $p > m$ such that

$$\frac{s_i^p}{\tilde{s}_i^m} < \frac{1}{\delta};$$

hence, $\tilde{s}_i^m$ will be updated no later than iteration p. Consequently $\tilde{s}_i^k$ is updated to s_i^k infinitely often, and since $\tilde{x}_i^k$ is updated to x_i^k whenever $\tilde{s}_i^k$ is updated and $x_i^k \to x_i^*$, we have

$$\tilde{x}_i^k \to x_i^* \text{ as } k \to \infty. \tag{33}$$

Expression (32) shows that $\left|(d_s^k)_i\right| = O(\Delta_k)$ and (33) implies $\left|x_i^k - \tilde{x}_i^k\right| \to 0$; therefore,

$$\frac{\left|\left(x_i^k - \tilde{x}_i^k\right)(d_s^k)_i\right|}{\Delta_k} \to 0. \tag{34}$$

Note also that

$$\left|\left(s_i^k - \tilde{s}_i^k\right)(d_x^k)_i\right| = \left|\left(\left(s_i^k - \tilde{s}_i^k\right)\frac{\sqrt{\tilde{x}_i^k}}{\sqrt{\tilde{s}_i^k}}\right)\left(\frac{\sqrt{\tilde{s}_i^k}}{\sqrt{\tilde{x}_i^k}}\right)(d_x^k)_i\right|.$$

But from Lemma 4.2(a),

$$\left|\left(\left(s_i^k - \tilde{s}_i^k\right)\frac{\sqrt{\tilde{x}_i^k}}{\sqrt{\tilde{s}_i^k}}\right)\right| = O\left(\sqrt{\Delta_k}\right),$$

and since (32) implies that

$$\left|\left(\frac{\sqrt{\tilde{s}_i^k}}{\sqrt{\tilde{x}_i^k}}\right)\right| = O\left(\sqrt{\Delta_k}\right),$$

with (29) we conclude that

$$\frac{\left|\left(s_i^k - \tilde{s}_i^k\right)(d_x^k)_i\right|}{\Delta_k} \to 0. \tag{35}$$

Since a similar argument holds for each i, expressions (34) and (35) imply (28), so $\theta_k \to 0$ and the algorithm is q-superlinearly convergent.

Recall from (16) that for each iteration,

$$\tilde{D}^{-1}d_x + \tilde{D}d_s = \tilde{v}.$$

If we multiply this equation by $(\tilde{X}\tilde{S})^{-1/2}$, we see that

$$\tilde{X}^{-1}d_x + \tilde{S}^{-1}d_s = (\tilde{X}\tilde{S})^{-1/2}\tilde{v}. \tag{36}$$

Once again, consider component i, and suppose $x_i^k \to x_i^* > 0$ and $s_i^k \to 0$ as $k \to \infty$. Since $\theta_k \to 0$ and $\left((d_x^k)_i / x_i^k\right) \to 0$, let $\bar{M}$ be such that $k \geq \bar{M}$ implies

$$\frac{\theta_k}{1-\beta} + \left|\frac{(d_x^k)_i}{x_i^k}\right| < \frac{1}{\delta}. \tag{37}$$

From the argument above, there exists an index $m > \bar{M}$ such that $\tilde{s}_i^m = s_i^m$ and $\tilde{x}_i^m = x_i^m$. This implies that, for iteration m, the ith row of (36) simplifies to

$$\frac{(d_x^m)_i}{x_i^m} + \frac{(d_s^m)_i}{s_i^m} = \frac{1}{\sqrt{x_i^m s_i^m}} \tilde{v}_i^m,$$

which, after solving for $(d_s^m)_i / s_i^m$, yields

$$\frac{(d_s^m)_i}{s_i^m} = \frac{\theta_m \Delta_m}{x_i^m s_i^m} - 1 - \frac{(d_x^m)_i}{x_i^m}. \tag{38}$$

Now,

$$\begin{aligned} \frac{s_i^{m+1}}{\tilde{s}_i^m} &= \frac{s_i^m + (d_s^m)_i}{s_i^m} \\ &= \frac{\theta_m \Delta_m}{x_i^m s_i^m} - \frac{(d_x^m)_i}{x_i^m} \text{ (by (38))} \\ &\leq \frac{\theta_m}{1-\beta} + \left| \frac{(d_x^m)_i}{x_i^m} \right| \text{ (by (7))} \\ &< \frac{1}{\delta} \text{ (by (37)),} \end{aligned}$$

so the partial updating scheme will update $\tilde{s}_i^{m+1} = s_i^{m+1}$ and $\tilde{x}_i^{m+1} = x_i^{m+1}$. Repeating the argument by replacing m with $m+1$ and so on shows that $\tilde{s}_i^k = s_i^k$ and $\tilde{x}_i^k = x_i^k$ for all $k \geq m$. A similar argument holds for each i; therefore, from some point on, complete factorizations will be performed every successive iteration. □

COROLLARY 5.3. *If the LCP is nondegenerate and has a strictly complementary solution* (x^*, s^*), *the complementarity gap sequence generated by the LCP algorithm converges to zero q-quadratically.*

Proof. The fact that the complementarity gap converges to zero is enough to guarantee that $(x^k, s^k) \to (x^*, s^*)$, so we do not need to assume that the iterates converge in this case. The previous theorem shows that from some point on complete factorizations will be performed, so the algorithm becomes in essence the LSPF algorithm, which is known to converge q-quadratically in the nondegenerate case with a strictly complementary optimal solution (see [11]). □

6. Concluding remarks. We have shown that, in the nondegenerate case with a strictly complementary optimal solution, a particular partial updating variant of the LSPF algorithm is q-quadratically convergent and has a computational complexity bound for achieving ε-optimality which is a factor of $\sqrt{n}$ smaller than that of the original algorithm. A similar q-superlinear convergence result is obtained for the degenerate case under the assumption that the iterates converge to a strictly complementary optimal solution.

REFERENCES

[1] K. M. ANSTREICHER, *A standard form variant, and safeguarded linesearch, for the modified Karmarkar algorithm*, Math. Programming, 47 (1990), pp. 337–351.

[2] K. M. ANSTREICHER AND R. A. BOSCH, *Long steps in an $O(n^3 L)$ algorithm for linear programming*, Math. Programming, 54 (1992), pp. 251–265.

[3] ———, *A new infinity-norm path following algorithm for linear programming*, SIAM J. Optim., 5 (1995), pp. 236–246.

[4] J. F. BONNANS AND C. C. GONZAGA, *Convergence of interior point algorithms for the monotone linear complementarity problem*, Math. Oper. Res., 21 (1996), pp. 1–25.

[5] R. A. BOSCH, *On Mizuno's rank-one updating algorithm for linear programming*, SIAM J. Optim., 3 (1993), pp. 861–867.

[6] R. A. BOSCH AND K. M. ANSTREICHER, *On partial updating in a potential reduction linear programming algorithm of Kojima, Mizuno, and Yoshise*, Algorithmica, 9 (1993), pp. 184–197.

[7] D. DEN HERTOG, C. ROOS, AND J. P. VIAL, *A complexity reduction for the long-step path following algorithm for linear programming*, SIAM J. Optim., 2 (1992), pp. 71–87.

[8] C. C. GONZAGA, *An algorithm for solving linear programming problems in $O(n^3L)$ operations*, in Progress in Mathematical Programming: Interior-point and Related Methods, N. Megiddo, ed., Springer-Verlag, 1989, pp. 1–28.

[9] ———, *The Largest Step Path Following Algorithm for Monotone Linear Complementarity Problems*, Report of the Faculty of Technical Mathematics and Informatics 94-07, Delft University of Technology, Delft, the Netherlands, 1994; Math. Programming, to appear.

[10] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[11] K. A. MCSHANE, *Superlinearly convergent $O(\sqrt{n}L)$-iteration interior point algorithms for linear programming and the monotone linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 247–261.

[12] S. MEHROTRA, *Deferred Rank One Updates in $O(n^3L)$ Interior Point Algorithms*, Technical Report 90-11, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1990.

[13] S. MIZUNO, *A rank one updating interior algorithm for linear programming*, Arabian J. Sci. Engrg., 15 (1990), pp. 671–677.

[14] ———, *$O(n^pL)$ iteration $O(n^3L)$ potential reduction algorithms for linear programming*, Linear Algebra Appl., 152 (1991), pp. 155–168.

[15] S. MIZUNO AND M. J. TODD, *An $O(n^3L)$ adaptive path following algorithm for a linear complementarity problem*, Math. Programming, 52 (1991), pp. 587–595.

[16] S. MIZUNO, A. YOSHISE, AND T. KIKUCHI, *Practical polynomial time algorithms for linear complementarity problems*, J. Oper. Res. Soc. Japan, 32 (1989), pp. 75–92.

[17] R. D. C. MONTEIRO AND I. ADLER, *Interior path following primal-dual algorithms, Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–42.

[18] ———, *Interior path following primal-dual algorithms, Part* II: *Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[19] Y. YE, *An $O(n^3L)$ potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 994–1006, November 1996
© 1996 Society for Industrial and Applied Mathematics
006

ALGORITHMS FOR THE GENERALIZED LINEAR COMPLEMENTARITY PROBLEM WITH A VERTICAL BLOCK Z-MATRIX*

S. R. MOHAN† AND S. K. NEOGY†

Abstract. In this paper, we consider the generalized linear complementarity problem VLCP (q, A) where A is a vertical block Z-matrix. We prove that the Cottle–Dantzig pivoting algorithm can process this problem by showing that this algorithm generates the same sequence of bases that a modified simplex algorithm for minimizing the artificial variable z_0 does. We also show that a modified version of the Cottle–Dantzig algorithm can be used for determining whether a given vertical block Z-matrix is a vertical block P_0-matrix.

Key words. generalized linear complementarity problem, vertical block Z-matrix, vertical block M_0-matrix, almost proper basis matrix, secondary proper ray

AMS subject classification. 90C33

1. Introduction.

We begin by introducing some basic notations used in this paper. We consider matrices and vectors with real entries. For any matrix $A \in R^{m\times n}$, a_{ij} denotes its ith row and jth column entry. $A_{.j}$ denotes the jth column and $A_{i.}$, the ith row of A. If A is a matrix of order $m \times n$, $J \subseteq \{1, 2, \ldots, m\}$, and $K \subseteq \{1, 2, \ldots, n\}$ then A_{JK} denotes the submatrix of A consisting of only the rows and columns of A whose indices are in J and K, respectively. For any set K, $|K|$ denotes its cardinality. For any set $J \subseteq \{1, 2, \ldots, m\}$, $\bar{J}$ denotes its complement in $\{1, 2, \ldots, m\}$. Any vector $x \in R^n$ is a column vector unless otherwise specified and x^t denotes the row transpose of x. e_i denotes the vector whose ith coordinate is 1 and whose other coordinates are 0's. We say that a square matrix $A = ((a_{ij}))$ of order n is a Z-matrix if $a_{ij} \leq 0$, $\forall i \neq j$. This class of matrices has been introduced by Fiedler and Pták [10]. An $n \times n$ matrix A is said to be a P (P_0)-matrix if all its principal minors are positive (nonnegative). An $n \times n$ matrix A is an M (M_0)-matrix if it is in $Z \cap P$ ($Z \cap P_0$).

Given a real square matrix M of order n and a vector $q \in R^n$, the linear complementarity problem is to find $w \in R^n$ and $z \in R^n$ such that

$$w - Mz = q, \quad w \geq 0, \; z \geq 0, \tag{1.1}$$

$$w^t z = 0. \tag{1.2}$$

This problem is denoted as LCP(q, M). It is well studied in the literature on mathematical programming and arises in a number of applications in operations research, mathematical economics, and engineering. For recent books on this problem, see Cottle, Pang, and Stone [3] and Murty [15].

We say that an $m \times k$ $(m \geq k)$ matrix A is a vertical block matrix of type $(m_1, m_2, \ldots, m_k)$ if the rows of A are partitioned into k blocks so that the first block contains the rows 1 through m_1 and in general the jth block contains the rows $\sum_{r=1}^{j-1} m_r + 1$ through $\sum_{r=1}^{j} m_r$ of A for $j = 2, \ldots, k$. The jth block of A is

* Received by the editors October 14, 1994; accepted for publication (in revised form) July 3, 1995.

† Statistical Quality Control and Operations Research Unit, Indian Statistical Institute, 7, S.J.S. Sansanwal Marg, New Dehli 110016, India (srm@isid.ernet.in, skn@isid.ernet.in).

denoted by A^j and is a matrix of order $m_j \times k$. In [2] Cottle and Dantzig consider a generalization of the linear complementarity problem as follows.

Given an $m \times k$ $(m \geq k)$ vertical block matrix A of type $(m_1, m_2, \ldots, m_k)$ and $q \in R^m$ where $m = \sum_{j=1}^{k} m_j$, find $w \in R^m$ and $z \in R^k$ such that

$$w - Az = q, \quad w \geq 0, \quad z \geq 0, \tag{1.3}$$

$$z_j \prod_{i=1}^{m_j} w_i^j = 0 \text{ for } j = 1, 2, \ldots, k, \tag{1.4}$$

where w_i^1 is the ith component of w for $1 \leq i \leq m_1$, and for $j > 1$, w_i^j is the $(\sum_{r=1}^{j-1} m_r + i)$th component of w for $1 \leq i \leq m_j$. This generalization is known as the *vertical linear complementarity problem* (see [3], p. 32) and it is denoted by VLCP(q, A). Lemke's algorithm for the LCP(q, M) (see [3]) has been extended with some modifications to the VLCP(q, A) by Cottle and Dantzig in [2].

In recent years a number of applications of the VLCP(q, A) have been reported in the literature. In [9] Ebiefung and Kostreva introduce a generalized Leontief input-output linear model and formulate it as a VLCP. This model can be effectively used for the problem of choosing a new technology and also for solving problems related to energy commodity demands, international trade, multinational army personnel assignment, and pollution control. Oh [16] has formulated a mixed lubrication problem as a generalized nonlinear complementarity problem. A slightly more general form of the VLCP(q, A) also arises in different areas of control theory through discretization of Hamilton–Jacobi–Bellman equations [19], [20]. In [11] Gowda and Sznajder report an interesting extension of the bimatrix game model, and the problem of computing a pair of equilibrium strategies for this extended model has a VLCP formulation. This sort of applications and the potential future applications have motivated the study of the VLCP, especially the study of algorithms for the VLCP.

The LCP(q, M) where M is Z-matrix has been extensively studied in the literature [1], [10], and [13] and arises in a number of applications (see [5] and [22]). In [7] Ebiefung and Kostreva introduce the class of vertical block Z-matrices. They present a generalization of Chandrasekaran's pivoting algorithm to solve LCP(q, M) where M is a Z-matrix (see [1]), for the VLCP(q, A) when A is a vertical block Z-matrix.

Ebiefung and Kostreva [9] point out that the Cottle–Dantzig algorithm can process the VLCP(q, A) where A is a vertical block M-matrix [2] but are silent about whether this algorithm can process the VLCP(q, A) where A is a vertical block Z-matrix. We say that an algorithm processes a problem if the algorithm can either compute a solution to it, if one exists, or show that no solution exists. For more explanations see [3]. The VLCP with a vertical block Z-matrix arises in some important practical problems (see [9], [16], [19], [20]). This paper is partially motivated by the question of whether the algorithm given by Cottle and Dantzig can solve the VLCP(q, A) where A is a vertical block Z-matrix. This is a generalization of a similar result due to Saigal [18] on the applicability of Lemke's method for LCP(q, M) when $M \in Z$. This we do by presenting a simplex type algorithm for **Problem P**, which is as follows:

$$\text{Minimize } z_0$$

$$\text{subject to } w - Az - dz_0 = q, \quad w \geq 0, \; z \geq 0, \; z_0 \geq 0, \tag{1.5}$$

where $d \in R^m$ is any positive vector and showing that the sequence of almost proper solutions generated by our modified simplex method for the above problem is the same as the sequence of almost proper solutions generated by the Cottle–Dantzig algorithm when (1.3) has a solution. Actually, this idea is again a generalization of the algorithm presented in [13]. The analysis presented in this paper leads to a greater understanding of the properties of a vertical block Z-matrix.

In §2, we present the required definitions and our algorithm. In §3, we show that when the VLCP(q, A) does not have a solution, the Cottle–Dantzig algorithm takes more iterations to detect the infeasibility than the modified simplex algorithm. In §4, we show that a modified version of the Cottle–Dantzig algorithm can be used for checking whether a given vertical block Z-matrix is a vertical block P_0-matrix or not. For the square Z-matrices such a checking algorithm has been given by Ramamurthy [17].

2. Preliminaries.

DEFINITION 2.1. *Let A be a vertical block matrix of type $(m_1, m_2, \ldots, m_k)$. A submatrix of size k of A is called a* representative submatrix *if its jth row is drawn from the jth block A^j of A. Clearly, a vertical block matrix of type $(m_1, m_2, \ldots, m_k)$ has $\prod_{j=1}^{k} m_j$ representative submatrices.*

Let $J_1 = \{1, 2, \ldots, m_1\}$ be the set of row indices in A corresponding to A^1 and let $J_r = \{\sum_{j=1}^{r-1} m_j + 1, \sum_{j=1}^{r-1} m_j + 2, \ldots, \sum_{j=1}^{r} m_j\}$ be the set of row indices in A corresponding to A^r, $r = 2, 3, \ldots, k$.

DEFINITION 2.2. *Let A be a vertical block matrix of type $(m_1, m_2, \ldots, m_k)$. Suppose G is a representative submatrix of A. Further suppose that the jth row of G is the r_jth row of A, where $r_j \in J_j$. Given the VLCP(q, A), let $q(G)$ be the $k \times 1$ vector whose jth component is q_{r_j}. The problem LCP$(q(G),G)$ is known as a* representative subproblem *associated with the VLCP(q, A).*

DEFINITION 2.3. *A vertical block matrix A of type $(m_1, m_2, \ldots, m_k)$ is called a* vertical block Z-matrix *if all its representative submatrices are Z-matrices.*

Vertical block $P(P_0)$-matrices are defined in a similar manner (see [8]).

DEFINITION 2.4. *A vertical block matrix A of type $(m_1, m_2, \ldots, m_k)$ is called a* vertical block M (M_0)-matrix *if it is a vertical block $Z \cap P$ $(Z \cap P_0)$-matrix.*

DEFINITION 2.5. *We call the set of variables $\{z_p\} \cup \{w_r,\ r \in J_p\}$ the pth set of* related variables.

Given a vertical block matrix A of type $(m_1, m_2, \ldots, m_k)$ consider the matrix $(I, -A)$ where I is the identity matrix of order $m \times m$.

DEFINITION 2.6. *We call B, a submatrix of order $m \times m$ of $(I, -A)$, a* proper basis *matrix if*

(i) *$-A_{\cdot j}$ is a column of $B \Rightarrow \exists$, an index $p \in J_j$ such that $I_{\cdot p}$ is not a column of B;*

(ii) *the columns of B form a linearly independent set.*

DEFINITION 2.7. *We call B, a submatrix of order $m \times m$ of $(I, -A, -d)$, an* almost proper basis *matrix if*

(i) *$-d$ is a column of B;*

(ii) *$-A_{\cdot j}$ is a column of $B \Rightarrow \exists$, an index $p \in J_j$ such that $I_{\cdot p}$ is not a column of B;*

(iii) *the columns of B form a linearly independent set.*

The terms proper and almost proper were introduced in [2].

DEFINITION 2.8. *Let B be an almost proper or proper basis matrix. We say that B is* feasible *for the VLCP(q, A) if $B^{-1}q \geq 0$.*

We say that an almost proper or proper basis matrix B which is feasible for the VLCP(q, A) is *nondegenerate* if $B^{-1}q > 0$.

We now look at the set

$$S = \{(w, z, z_0) \mid w \in R^m, z \in R^k, z_0 \in R,\ w - Az - dz_0 = q,\ w \geq 0,\ z \geq 0, z_0 \geq 0\}.$$

We say that a $(w, z, z_0) \in S$ is a *basic feasible solution* to (1.5) if the set of columns of $(I, -A, -d)$ corresponding to the positive coordinates of (w, z, z_0) are linearly independent and a *nondegenerate basic feasible solution* if, in addition, the number of positive coordinates of (w, z, z_0) is m.

Assume that all the basic feasible solutions to (1.5) are nondegenerate. Under this assumption the extreme points of S are in one-one correspondence with the basic feasible solutions to (1.5). Their correspondence is well discussed in many text books on linear programming. For instance, see Hadley ([12], p. 100). In particular, if B is a proper basis matrix, which is feasible, the extreme point corresponding to it is a solution to the VLCP(q, A).

DEFINITION 2.9. *We say that an extreme point of S is* almost proper *if the corresponding basis matrix is almost proper.*

DEFINITION 2.10. *Let $y = (u^*, \bar{v}, v_0)$ be an extreme direction of S and let $(w^*, \bar{z}, z_0) \in S$. Consider the ray of points*

$$(w(\lambda), z(\lambda), z_0(\lambda)) := (w^*, \bar{z}, z_0) + \lambda\,(u^*, \bar{v}, v_0) \in S \ \ \forall\, \lambda \geq 0.$$

We call such a ray an almost proper ray *if*

$$z_j(\lambda) \prod_{i=1}^{m_j} w_i^j(\lambda) = 0, \quad \forall\, \lambda \geq 0,\ \ j = 1, \ldots, k.$$

Remark 2.1. It is easy to verify that if $(w(\lambda), z(\lambda), z_0(\lambda)) := (w^*, \bar{z}, z_0) + \lambda\,(u^*, \bar{v}, v_0)$ is an almost proper ray, then $(u^*, \bar{v}, v_0) \geq 0$, $u^* - A\bar{v} - dv_0 = 0$, and

$$\bar{v}_j \prod_{i \in J_j} u^{*j}_i = \bar{z}_j \prod_{i \in J_j} w^{*j}_i = \bar{v}_j \prod_{i \in J_j} w^{*j}_i = \bar{z}_j \prod_{i \in J_j} u^{*j}_i = 0,\ \ j = 1, 2, \ldots, k.$$

In what follows we make the standard nondegeneracy assumption that all the almost proper basic feasible solutions to (1.5) are nondegenerate. We now state the Cottle–Dantzig algorithm for solving the VLCP(q, A). The Cottle–Dantzig algorithm for the VLCP(q, A) starts with the initial solution to (1.5) as

$$w = q + d\,z_0;\ z = 0,$$

where z_0 is large enough so that $w > 0$.

Step 1: Decrease z_0 to $\bar{z_0} = \min\{z_0 \mid q + d\,z_0 \geq 0,\ z_0 \geq 0\}$ so that one of the variables w_i, $1 \leq i \leq m$, say w_p, is reduced to zero. We now have a basic feasible solution with z_0 in place of w_p. This is the initial almost proper basic feasible solution. Now let r be the unique index, $1 \leq r \leq k$, such that $p \in J_r$. We have exactly one pair of nonbasic variables (z_r, w_p) which belong to the same set of related variables.

Step 2: At each iteration, there is exactly one pair of nonbasic variables belonging to the same set of related variables. Of these, one has been eliminated from the basis in the previous iteration; the other is now selected to be included in the basis. For example, in the second iteration z_r is selected to be included in the basis.

Step 3: If the variable selected at Step 2 to enter the basis can be arbitrarily increased, then the procedure terminates in an almost proper ray, to be called a *secondary proper ray.* If a new basic feasible solution is obtained with $z_0 = 0$ or z_0 is nonbasic, then we have solved (1.3) and (1.4) and have a solution for the VLCP(q, A). Otherwise, we have obtained a new almost proper basic feasible solution and a new pair of nonbasic variables (x_β, y_r) belonging to the same set of related variables, say the sth set, where either $(x_\beta, y_r) = (z_s, w_t)$ with $t \in J_s$ or $(x_\beta, y_r) = (w_{t_1}, w_{t_2})$ with $t_1, t_2 \in J_s$. We repeat Step 2.

The Cottle–Dantzig algorithm (Algorithm CD) consists of the repeated application of Steps 2 and 3. Under our nondegeneracy assumption, the procedure either terminates in a *solution* to the VLCP(q, A) or in a *secondary proper ray.*

Given a vertical block Z-matrix of type $(m_1, m_2, \ldots, m_k)$ and a vector $q \in R^m$ we can also apply a modified simplex method to Problem P.

The modifications we introduce are as follows:

(i) We start with the initial basic feasible solution as $w = q + d\bar{z}_0$ and $z = 0$ where $\bar{z}_0 = \min\{z_0 \mid q + d z_0 \geq 0,\ z_0 \geq 0\}$.

(ii) At any iteration if of the pair of nonbasic variables belonging to the same set of related variables there is one with a positive simplex multiplier (negative of the relative cost coefficient as in [14, p. 58]), then that variable is selected for inclusion in the set of basic variables at the next iteration.

We call the simplex method with the above modifications *Algorithm S.* While applying the above algorithm, we may encounter two types of almost proper basis matrices. We define these types below:

DEFINITION 2.11. *We say that an almost proper basis matrix B is of* type I *if for each r, $1 \leq r \leq k$, the fact that $-A_{\cdot r}$ is a column of B implies that for exactly one index $j \in J_r$, $I_{\cdot j}$ is not a column of B.*

DEFINITION 2.12. *An almost proper basis matrix generated by the algorithm is said to be of* type II *if it is not of* type I.

Remark 2.2. Note that if an almost proper basis matrix B is of type II, then there is exactly one index s, $1 \leq s \leq k$ such that $-A_{\cdot s}$ is a column of B and exactly two indices l and p in J_s such that $I_{\cdot l}$ and $I_{\cdot p}$ are not included in B.

DEFINITION 2.13. *Let B be an almost proper basis matrix of $(I, -A, -d)$ and let π be a permutation of $\{1, 2, \ldots, m\}$; i.e. $\pi(1), \pi(2), \ldots, \pi(m)$ is a permutation of $1, 2, \ldots, m$. By permuting the rows and columns of B using the same permutation π we obtain the matrix $\bar{B}$. We call $\bar{B}$ a principal rearrangement of B.*

DEFINITION 2.14. *Let B be an almost proper basis matrix of $(I, -A, -d)$. Suppose $B_{\cdot p} = -d$, for some $p \in J_t$. We say that B is in standard form if $\forall\ j \neq p$ $B_{\cdot j}$ is either $I_{\cdot j}$ or $-A_{\cdot r}$ where $j \in J_r$. Further if B is of type I then neither $I_{\cdot p}$ nor $-A_{\cdot t}$ are columns of B. If B is of type II then there exists an index $l \in J_t$ such that neither $I_{\cdot p}$ nor $I_{\cdot l}$ are columns of B.*

Remark 2.3. We note that by permuting the columns of B, we can obtain its standard form. We note the following result, which will be useful in §3.

LEMMA 2.1. *Let A be a vertical block Z-matrix. Suppose B is an almost proper*

basis matrix of $(I, -A, -d)$ *which is in the standard form. By a principal rearrangement of the rows and columns of* B, *we can write* B *in the partitioned form as* C, *where*

$$C = \begin{bmatrix} I & D_1 & -\bar{d}^1 \\ 0 & D_2 & -\bar{d}^2 \\ 0 & g^t & -\bar{d}_m \end{bmatrix}.$$

Further, (i) $-D_2$ *is a* Z*-matrix, and* (ii) $g^t \geq 0$ *if* B *is of type* I *and at most one coordinate of* g^t *is negative if* B *is of type* II.

Proof. Let p be an index such that $B_{\cdot p} = -d$. Suppose B includes exactly r columns from $-A$. Then B includes exactly $(m-r-1)$ columns from I. Let $t_1 < t_2 < \cdots < t_{m-r-1}$ be the indices such that $I_{\cdot t_j}$ is a column of B. Actually, since B is in the standard form, $B_{\cdot t_j} = I_{\cdot t_j}$. Let $L = \{l \mid -A_{\cdot l} \text{ is a column of } B\}$, and let $l_1 < l_2 < \cdots < l_r$ be the ordered set of indices in L.

Case 1. Suppose B is of type I. Then given any $l_i \in L$ there exists exactly one index $p_i \in J_{l_i}$ such that $I_{\cdot p_i}$ is not a column of B and $B_{\cdot p_i} = -A_{\cdot l_i}$. Now let $\pi(t_i) = i$ for $i = 1, 2, \ldots, (m-r-1)$, $\pi(p_i) = m-r-1+i$ for $i = 1, 2, \ldots, r$, and $\pi(p) = m$. Then the matrix C obtained by applying the principal rearrangement using π has the above form. Now, note that D_2 is actually a submatrix of $-A$, containing the columns $l_1, l_2, \ldots, l_r$ and the rows $p_1, p_2, \ldots, p_r$ of $-A$. Note that as $p_i \in J_{l_i}$, this is a principal submatrix of a representative submatrix of $-A$. Thus $-D_2$ is a Z-matrix. Also as g^t is obtained by a permutation of the elements of the pth row of $-A$, where $p \notin J_{l_i}$, $i = 1, 2, \ldots, r$, $g^t \geq 0$. This completes the proof in this case.

Case 2. Suppose B is of type II. In this case there is exactly one index s and exactly two indices p and l in J_s such that $I_{\cdot l}$ and $I_{\cdot p}$ are not columns of B and $-A_{\cdot s}$ is a column of B. We note that $s = l_i$ for some i. Since B is in the standard form there exists exactly one index $p_j \in J_{l_j}$ such that $I_{\cdot p_j}$ is not a column of B for all j, $1 \leq j \leq r$, $j \neq i$. For $j = i$, however, $l_i = s$, and we have two indices p and $l \in J_s$ such that $I_{\cdot p}$ and $I_{\cdot l}$ are not columns of B. We have $B_{\cdot p_j} = -A_{\cdot l_j}$ and either $B_{\cdot p} = -d$ or $B_{\cdot l} = -d$. Now we take $\pi(t_\alpha) = \alpha$, $\alpha = 1, 2, \ldots, (m-r-1)$, $\pi(p_j) = m-r-1+j$, $j \neq i$. Without loss of generality suppose that $B_{\cdot p} = -d$ then take $\pi(l) = m-r-1+i$ and $\pi(p) = m$. It is easy to see as in the proof of Case 1, that by permuting the rows and columns of B using π, we obtain C, which has the above form. As before we note that $-D_2$ is a principal submatrix of a representative matrix of A and therefore $-D_2 \in Z$. Note that g^t is obtained by a permutation of the pth row of $-A$, where $p \notin J_{l_j}$, $1 \leq j \leq r$, $j \neq i$. But $p \in J_{l_i}$ and hence it follows that $g_j \geq 0$, $j \neq i$, and g_i may be negative. Thus at most one coordinate of g^t is negative. This concludes the proof. □

3. Validity of the algorithms. In order to prove that both Algorithm CD and Algorithm S can process the VLCP(q, A) under the standard nondegeneracy assumption where A is a vertical block Z-matrix, we require the following lemmas.

LEMMA 3.1. *Suppose* B *is an almost proper basis matrix of* $(I, -A, -d)$ *of type* I *which is in the standard form. Let* C *be obtained as in Lemma* 2.1. *Suppose* $-D_2$ *is a* P*-matrix. Let* f^t *be the last row of* C^{-1}. *Then* $f \leq 0$.

Proof. Since $f^t C = e_m^t$ it follows that $f_j = 0$, $j = 1, 2, \ldots, (m-r-1)$. Also, we obtain the equations

$$\bar{f}^t D_2 + f_m g^t = 0, \tag{3.1}$$

$$\sum_{j=m-r}^{m} f_j(-\bar{d}_j) = 1, \tag{3.2}$$

where $\bar{f}^t = (f_{m-r}, f_{m-r+1}, \ldots, f_{m-1})$ and $\bar{d}$ is the last column of C.

From (3.1) we note that

$$\begin{aligned}\bar{f}^t &= -f_m\, g^t (D_2)^{-1} \\ &= f_m\, g^t (-D_2)^{-1}.\end{aligned}$$

Since $-D_2$ is an M-matrix and $g \geq 0$, it follows that f_m and the coordinates of $\bar{f}$ have the same sign. It now follows from (3.2) that $f \leq 0$. □

LEMMA 3.2. *Let B be an almost proper basis matrix of $(I, -A, -d)$ of type* II *which is in the standard form. Let p be the index such that $B_{\cdot p} = -d$, $l_1 < l_2 < \cdots < l_r$ be the indices such that $-A_{\cdot l_j}$ is a column of B, for $j = 1, 2, \ldots, r$ and l_i be the index such that $p \in J_{l_i}$. Let C be the partitioned form obtained after a principal rearrangement of the rows and columns of B as in Lemma* 2.1. *Suppose $-D_2$ is a P-matrix. Let f^t be the last row of C^{-1}. If $f \not\leq 0$ then at least one of the two coordinates f_m and $f_{m-r-1+i}$ is positive.*

Proof. Since $f^t C = e_m^t$ we obtain the equations (3.1) and (3.2). Further, $f_j = 0$, $j = 1, 2, \ldots, (m - r - 1)$.

From (3.1) we have

$$\begin{aligned}\bar{f}^t &= -f_m\, g^t (D_2)^{-1} \\ &= f_m\, g^t (-D_2)^{-1}.\end{aligned}$$

If $f_m > 0$ then we have nothing to prove. So let $f_m \leq 0$ and $f_{m-r-1+i} \leq 0$.

Let $\tilde{f}$ be obtained from $\bar{f}$ by omitting its ith coordinate and let $\tilde{D}_2$ be the matrix obtained by omitting the ith row and column of D_2. Also let $\tilde{g}$ be obtained from g by omitting its ith coordinate. Then (3.1) leads to

$$\tilde{f}^t(-\tilde{D}_2) = f_m\, \tilde{g}^t + f_{m-r-1+i}\, (\tilde{D}_2)_{i\cdot}$$

Note that $\tilde{g}^t$ and $(\tilde{D}_2)_{i\cdot}$ are nonnegative vectors. As $-\tilde{D}_2$ is also an M-matrix it follows that $\tilde{f} \leq 0$ and hence $f \leq 0$.

Note also that if $f_m = 0$, then $f = 0$, which contradicts the nonsingularity of C^{-1}. This completes the proof of the lemma. □

We now state and prove our main theorem which establishes the validity of the algorithms presented in §2.

THEOREM 3.1. *Let A be a vertical block Z-matrix and consider the VLCP(q, A). Suppose* (1.3) *has a solution. If we apply Algorithm S to Problem P then the sequence of almost proper basis matrices $B(t)$ generated is the same as the sequence of almost proper basis matrices generated by Algorithm CD. Further, Algorithm S and hence Algorithm CD compute a solution to the VLCP(q, A).*

Proof. Note that the initial almost proper basis matrix $B(0)$ used by both of the algorithms is the same and contains $I_{\cdot j}$ for $1 \leq j \leq m$, $j \neq p$, for some p, $1 \leq p \leq m$. Further $(B(0))_{\cdot p} = -d$. Suppose $p \in J_u$. Note that $B(0)$ is of type I and let π be the permutation as in Lemma 2.1. Let f^t be the last row of $(C(0))^{-1}$. Let $y_{\cdot j}$ be the jth column of $(I, -A, -d)$ and let $\bar{y}_{\cdot j}$ be obtained from $y_{\cdot j}$ by permutation of its coordinates using π. Now since $f \leq 0$, if $f^t \bar{y}_{\cdot j} > 0$ for some j, then $f^t \bar{y}_{\cdot j} = \frac{a_{pu}}{d_p}$ and $y_{\cdot j} = -A_{\cdot u}$. Note also that as (1.3) has a feasible solution, there is such a $y_{\cdot j}$

and $-A_{.u}$ is selected to be included in the basis. Thus both of the algorithms select the same column, $-A_{.u}$, to be included in the basis at the next iteration. Further, as $B(1)$ in the standard form has $(B(1))_{.p} = -A_{.u}$, $D_2(1)$ is a 1×1 matrix which is $(-a_{pu}) < 0$. Hence $-D_2(1)$ is a P-matrix.

We use the same notation as used in the proof of Lemma 2.1. In particular if $B(t)$ is the almost proper basis matrix at the tth iteration which is in the standard form, $C(t)$ is obtained by using a permutation π_t as in Lemma 2.1.

Let us make the induction hypothesis that for all $t \leq v$ the almost proper basis matrices generated by both of the algorithms are the same and that $-D_2(t)$ is a P-matrix. Let f^t be the last row of $(C(v))^{-1}$. We have two cases to consider.

Case (i). $B(v)$ is an almost proper basis matrix of type I. By Lemma 3.1, we have $f \leq 0$. Let $y_{.j}$ be the jth column of $(I, -A, -d)$. Let $\bar{y}_{.j}$ be obtained by permuting the coordinates of $y_{.j}$ using π_v. Now let p be as in the proof of Lemma 2.1, i.e., $(B(v))_{.p} = -d$ and let l be such that $p \in J_l$. Now by Lemma 3.1 we have $f^t \leq 0$, as $-D_2(v)$ is an M-matrix. Suppose $y_{.j} = I_{.t}$ is a column of $(I, -A, -d)$ not in $B(v)$. Then clearly $f^t \bar{y}_{.j} \leq 0$. If $y_{.j} = -A_{.s}$ for $s \neq l$ then $f^t \bar{y}_{.j} = \sum f_{m-r-1+\alpha}\,(-a_{\pi_v(p_\alpha)s}) \leq 0$ as $(-a_{\pi_v(p_\alpha)s}) \geq 0 \ \forall\ \alpha$. Therefore, only when $y_{.j} = -A_{.l}$, $f^t \bar{y}_{.j}$ may be positive. If $f^t \bar{y}_{.j}$ is not positive when $y_{.j} = -A_{.l}$, then Algorithm S terminates with the conclusion that (1.3) does not have a solution, contradicting our hypothesis. Hence $f^t \bar{y}_{.j} > 0$, for j such that $y_{.j} = -A_{.l}$, and Algorithm S selects $-A_{.l}$ to be included in $B(v+1)$. Algorithm CD also selects the same column, as w_p and z_l are the pair of nonbasic variables belonging to the same set of related variables.

Let j be the index such that $y_{.j} = -A_{.l}$ and let $x = (C(v))^{-1}\,\bar{y}_{.j}$. Let $x^t = (x_1, x_2, \eta)^t$ be the partition of x induced by π_v where $\eta \in R$. We have the equation

$$I\,x_1 + D_1(v)\,x_2 + (-\bar{d}^1)\,\eta = -\bar{y}_{Tj},$$

where $T = (t_1, t_2, \ldots, t_{m-r-1})$ is the index set as given in the proof of Lemma 2.1. We also have

$$D_2(v)\,x_2 + (-\bar{d}^2)\,\eta = h,$$

where $h = (-a_{p_1 l}, -a_{p_2 l}, \ldots, -a_{p_r l})$.

As $\eta = f^t\,(\bar{y}_{.j})$, we have $\eta > 0$ and we have

$$D_2(v)\,x_2 = h + \eta\,\bar{d}^2.$$

As $-D_2(v)$ is an M-matrix and as $h + \eta\,\bar{d}^2 \geq 0$, we have $x_2 \leq 0$. This shows that $-A_{.l}$ does not replace any $-A_{.l_i}$ which is in $B(v)$. Therefore, $D_2(v+1)$ is given by

$$D_2(v+1) = \begin{bmatrix} D_2(v) & h \\ f & -a_{pl} \end{bmatrix}.$$

Note that

$$D_2(v+1)\begin{bmatrix} x_2 \\ -1 \end{bmatrix} = \eta \begin{bmatrix} \bar{d}^2 \\ \bar{d}_m \end{bmatrix} \quad \text{or} \quad [-D_2(v+1)]\begin{bmatrix} -x_2 \\ 1 \end{bmatrix} = \eta \begin{bmatrix} \bar{d}^2 \\ \bar{d}_m \end{bmatrix} > 0.$$

As $-D_2(v+1)$ is a Z-matrix, it follows from a theorem of Fiedler and Pták [10, p. 387] that $-D_2(v+1)$ is a P-matrix.

Note that both of the algorithms use the same rule for determining the column to be replaced from $B(v)$.

If the column replaced by $-A_{.l}$ is $-d$, then both of the algorithms terminate with a solution to the VLCP(q, A). Otherwise they generate the same almost proper basis matrix $B(v+1)$ in the next iteration. Further, $-D_2(v+1)$ is a P-matrix.

Case (ii). $B(v)$ is an almost proper basis matrix of type II. Let $B(v)_{.p} = -d$. Suppose $f \leq 0$. Then $f^t \bar{y}_{.j} \leq 0$ if $y_{.j} = I_{.t}$ for some t. Thus, none of the columns of I will be selected to enter the basis by Algorithm S.

Also, as in *Case* (i), for t such that $p \notin J_t$, the simplex multiplier corresponding to the nonbasic column $-A_{.t}$ is nonpositive, since for $y_{.j} = -A_{.t}$ we have

$$f^t \bar{y}_{.j} = \sum_{\alpha=1}^{r} f_{m-r-1+\alpha}\,(-a_{\pi_v(p_\alpha)t}),$$

where $-a_{\pi_v(p_\alpha)t} \geq 0$.

Now as in the proof of Lemma 2.1 let s be such that p and $l \in J_s$, and $s = l_i$. Since $-A_{.l_i}$ is already a column of $B(v)$, we find that if $f \leq 0$, there is no column whose simplex multiplier is positive. It implies that z_0 cannot be further reduced and Algorithm S terminates with the conclusion that there is no solution to (1.3) contradicting our hypothesis. Thus $f \not\leq 0$. By Lemma 3.2 at least one of the two coordinates f_m and $f_{m-r-1+i}$ is positive. Note also that of the nonbasic pair $(I_{.p}, I_{.l_i})$ corresponding to the same set of related variables one must have been eliminated from the basis matrix $B(v-1)$. Since a column eliminated from the basis matrix cannot re-enter it immediately (see [12, p. 144]), it follows that either $f_m > 0$ and $f_{m-r-1+i} \leq 0$ or $f_m \leq 0$ and $f_{m-r-1+i} > 0$. Thus both of the algorithms select the same column to be included in the basis. Now since the column to leave $B(v)$ is determined by both of the algorithms using the same minimum ratio rule, if this is $-d$, both of the algorithms terminate with a solution to the VLCP(q, A). Otherwise they both obtain the same $B(v+1)$. Let b be either $I_{.p}$ or $I_{.l_i}$, whichever is selected by the algorithm to be included in $B(v+1)$, and let $\bar{b}$ be the permutation of its rows using π_v.

Let $x = (C(v))^{-1}\bar{b}$. Let $x^t = (x_1, x_2, \eta)^t$ be the partition of x induced by the partitioned form of $C(v)$, where $\eta \in R$.

We then have

$$C(v)\,x = \bar{b} \text{ or}$$

$$x_1 + D_1(v)\,x_2 + \eta\,(-\bar{d}^1) \geq 0, \tag{3.3}$$

$$D_2(v)\,x_2 + \eta\,(-\bar{d}^2) \geq 0. \tag{3.4}$$

Thus we have $D_2(v)\,x_2 \geq \eta\,\bar{d}^2$.

Since $\eta > 0$ and $-D_2(v)$ is an M-matrix, it follows that $x_2 \leq 0$. Thus none of the columns $-A_{.l_j}$ included in $B(v)$ will be selected to leave the basis.

This shows that $D_2(v+1) = D_2(v)$.

As $-D_2(v)$ is a P-matrix it follows that $-D_2(v+1)$ is also a P-matrix. It now follows, by the principle of induction, that the sequence of almost proper basis matrices generated by both of the algorithms is the same when (1.3) has a solution, and they both compute a solution to the VLCP(q, A). When (1.3) does not have a solution, then Algorithm S terminates with a $z_0 > 0$, thus showing that there is no solution to the VLCP(q, A). In this case, Algorithm CD will terminate in a secondary proper ray. This concludes the proof. □

Remark 3.1. From Theorem 3.1, it follows that when (1.3) has a solution, both Algorithm S and Algorithm CD generate the same sequence of basis matrices. However, when (1.3) does not have a solution, Algorithm S never requires more iterations than Algorithm CD requires to detect the infeasibility. There are instances when Algorithm CD may take a few more iterations than Algorithm S does. The following example illustrates this. The fact that Algorithm S never requires more iterations than Algorithm CD requires is clear from the conditions for terminating these algorithms. Thus Algorithm S is clearly preferable to Algorithm CD for processing the VLCP(q, A) when A is a vertical block Z-matrix.

Example 3.1. Consider the vertical block Z-matrix

$$A = \begin{bmatrix} 3 & -1 & -2 \\ 4 & -3 & -2 \\ \\ -1 & 3 & -1 \\ -2 & 4 & -3 \\ -1 & 6 & -1 \\ \\ -1 & -2 & 3 \\ -3 & -1 & 5 \\ -2 & -3 & 6 \\ -1 & -1 & 4 \end{bmatrix}, \quad q = \begin{bmatrix} -4 \\ 2 \\ \\ 3 \\ -1 \\ -3 \\ \\ -10 \\ -5 \\ -6 \\ -8 \end{bmatrix},$$

where $m_1 = 2$, $m_2 = 3$, $m_3 = 4$. Algorithm CD takes four iterations to terminate in a secondary proper ray. Algorithm S terminates in three iterations with $\min z_0 > 0$, demonstrating infeasibility.

Remark 3.2. When (1.3) has a solution, we note that Algorithm S generates at most k almost proper type I basis matrices since at every iteration with a type I basis, the algorithm introduces one of the z_j variables into the basis, and a basic z_j variable never leaves the basis. However, there may be a number of almost proper type II basis matrices generated between two successive almost proper type I basis matrices. The question of complexity is being studied.

Remark 3.3. Note that when Algorithm CD or Algorithm S terminates with a solution, the corresponding proper basis matrix B can be written (after a principal rearrangement of its rows and columns) as

$$B = \begin{bmatrix} I & D_1 \\ 0 & D_2 \end{bmatrix},$$

where $-D_2$ is an M-matrix. This is a consequence of our Theorem 3.1. Now it is easy to show that B is an optimal basis for the following class of linear programs as well:

$$\text{Minimize} \quad \sum_{j=1}^{k} c_j z_j$$

$$\text{subject to } w - Az = q, \quad (w, z) \geq 0,$$

where $c = (c_1, c_2, \ldots, c_k)$ is any nonnegative vector. This shows that the solution corresponding to B is actually the least element of the set

$$X(q) = \{(w, z) \mid w - Az = q, w \geq 0, z \geq 0\}.$$

(See Cottle and Veinott [4] for the definition and properties of the least element.) Thus Algorithms CD and S both compute the least element of $X(q)$ whenever $X(q) \neq \phi$. This gives us an independent and constructive proof for the results observed by Sznajder [21] and Ebiefung and Kostreva [7].

4. An application of the Cottle–Dantzig algorithm. In this section we present a modification of Algorithm CD to determine whether a given vertical block Z-matrix A of type $(m_1, m_2, \ldots, m_k)$ is a vertical block M_0-matrix or not. We first state a few results.

LEMMA 4.1. *Suppose* (w^*, z^*) *solves the VLCP*$(0, A)$. *Let* $K = \{j \mid z_j^* > 0\}$ *and* $J = \cup_{j \in K} J_j$. *Then* (i) $A_{\bar{J}K} = 0$, (ii) $A_{\bar{J}\bar{K}} z_{\bar{K}}^* = 0$, (iii) $w_{\bar{J}}^* = 0$, *and* (iv) $A_{JK} z_K^* \geq 0$.

Proof. Since (w^*, z^*) solves the VLCP$(0, A)$ we have $w^* - A z^* = 0$. This implies that for $j \in K$ $\exists\, i \in J_j$ such that $w_i^* = 0$ and $\sum_{j \in K} a_{ij} z_j^* = 0$. But for $i \in \bar{J}$, we have $w_i^* - \sum_{j \in K} a_{ij} z_j^* = 0$, which implies $w_i^* = 0$ and $\sum_{j \in K} a_{ij} z_j^* = 0$, as $a_{ij} \leq 0$ for $i \in \bar{J}$ and $j \in K$. It follows from here that $A_{\bar{J}K} = 0$ and $w_{\bar{J}}^* = 0$ as $z_j^* > 0$ for $j \in K$. This proves (i) and (iii). Also, since for $i \in J$, $w_i^* = \sum_{j \in K} a_{ij} z_j^* \geq 0$, as $w^* \geq 0$. Hence (iv) holds. (ii) holds trivially. This completes the proof. □

LEMMA 4.2. *Suppose* A *is a vertical block* Z*-matrix of type* $(m_1, m_2, \ldots, m_k)$ *and let* $q \in R^m$ *be such that the VLCP*(q, A) *has no solution. Let* $\Re = \{(\bar{w}, \bar{z}, \bar{z}_0) + \lambda(\hat{w}, \hat{z}, \hat{z}_0),\ \lambda \geq 0\}$ *be the secondary proper ray generated by Algorithm CD. Let* $K = \{j \mid \bar{z}_j \neq 0\}$ *and* $\bar{K} = \{j \mid \bar{z}_j = 0\}$. *If* $\hat{z}_0 \neq 0$ *then* A *is not a vertical block* M_0*-matrix. If* $\hat{z}_0 = 0$ *then* (i) $A\hat{z} \geq 0$; (ii) $A_{\bar{J}K} = 0$, *where* $\bar{J} = \cup_{j \in \bar{K}} J_j$; (iii) A_{JK} *is a vertical block* M_0*-matrix, where* $J = \cup_{j \in K} J_j$.

Proof. It is easy to verify from Definition 2.10 and Remark 2.1 that $(\hat{w}, \hat{z}, \hat{z}_0) \geq 0$, $\hat{z}_j \prod_{i \in J_j} \hat{w}_i^j = 0$, and $\hat{w} - A\hat{z} - d\hat{z}_0 = 0$. If $\hat{z}_0 \neq 0$, then $(\hat{w}, \hat{z})$ is a nontrivial solution to the VLCP$(\hat{z}_0\, d, A)$. This implies that there is a representative matrix G of A such that $\hat{z}$ is a nontrivial solution of the LCP$(\hat{z}_0\, d(G), G)$ (see [9, Prop. 4.3]). It now follows from [6, Eqn. (3), lemma, p. 620 and Eqn. (18), lemma, p. 622] that G is not a P_0-matrix. Hence A is not a vertical block P_0-matrix.

If $\hat{z}_0 = 0$, it is easy to verify that $(\hat{w}, \hat{z})$ is a nontrivial solution to the VLCP$(0, A)$. Now the conclusions (i) and (ii) follow from Lemma 4.1. By Lemma 4.1, we also have $A_{JK} \hat{z}_K \geq 0$ and $\hat{z}_K > 0$. If G is any representative submatrix of A_{JK} of order $|K|$, then $G\hat{z}_K \geq 0$, $\hat{z}_K > 0$. Since G is a square Z-matrix it follows from a theorem of Fiedler and Pták [10, Thm. (5.4)] that G is a P_0-matrix. Hence A_{JK} is a vertical block M_0-matrix. □

LEMMA 4.3. *Let* A *be a vertical block* Z*-matrix of type* $(m_1, m_2, \ldots, m_k)$ *and* $q \in R^m$ *be such that* $q < 0$. *Then the VLCP*(q, A) *has a solution if and only if* A *is a vertical block* M*-matrix.*

Proof. Suppose $q < 0$. Let (w^*, z^*) be a solution to the VLCP(q, A). It follows that $w^* - q = Az^* > 0$. Now from Theorem 3 [8, p. 169] it follows that A is a vertical block M-matrix. Conversely, suppose A is a vertical block M-matrix. By Theorem 3 [8, p. 169], it follows that there is an $x \geq 0$, $x \in R^k$ such that $Ax > 0$. Given any $q \in R^m$ choose $\mu > 0$, so that $w = q + A(\mu x) > 0$. Now it follows from Theorem 3.1 that VLCP(q, A) has a solution. This holds for each $q \in R^m$. □

We now present an algorithm to verify whether a given vertical block Z-matrix A of type $(m_1, m_2, \ldots, m_k)$ is a vertical block P_0-matrix or not.

Step 0: Let A be the given vertical block matrix of type $(m_1, m_2, \ldots, m_k)$. Let $\gamma = \{1, 2, \ldots, k\}$. Choose any $q < 0$ and go to Step 1.

Step 1: Let $\theta = \gamma$ and apply the following version of Algorithm CD to the VLCP(q_J, $A_{J\theta}$) where $J = \cup_{j\in\theta} J_j$. Execute the steps of the Algorithm CD until either a basic z_j variable begins to decrease or a secondary proper ray is encountered or a solution to the VLCP(q_J, $A_{J\theta}$) is obtained. If the algorithm is terminated due to a decrease in a basic z_j variable, then we say that the algorithm is terminated prematurely.

(a) If the algorithm terminates with a solution to the VLCP(q_J, $A_{J\theta}$), stop. A is a vertical block M_0-matrix.

(b) If there is a premature termination, stop. A is not a vertical block M_0-matrix.

(c) If the algorithm terminates with a secondary proper ray, then go to Step 2.

Step 2: Let $\Re = \{(\bar{w}, \bar{z}, \bar{z}_0) + \lambda(\hat{w}, \hat{z}, \hat{z}_0),\ \lambda \geq 0\}$ be the secondary proper ray generated by Algorithm CD when applied to the VLCP($q_J, A_{J\theta}$). If $\hat{z}_0 \neq 0$, stop. A is not a vertical block M_0-matrix. Otherwise go to Step 3.

Step 3: Let $K = \{j \mid \hat{z}_j \neq 0\}$ and $\bar{K} = \{j \mid \hat{z}_j = 0\}$. If $\bar{K} = \phi$ stop. A is a vertical block M_0-matrix, otherwise $\gamma = \bar{K}$ and return to Step 1.

THEOREM 4.1. *The above algorithm verifies whether A is a vertical block M_0-matrix or not in a finite number of steps.*

Proof. The matrix $A_{J\theta}$ is a vertical block Z-matrix at any stage of the algorithm. Now if Case (a) at Step 1 occurs, then $A_{J\theta}$ is a vertical block M-matrix by Lemma 4.3. However, if Case (b) occurs at Step 1, $A_{J\theta}$ is not a vertical block M_0-matrix by our results in the previous section. If we stop with Step 2 at any stage of the algorithm, then it follows from the first part of Lemma 4.2 that $A_{J\theta}$ is not a vertical block M_0-matrix. If we stop at Step 3, with $\bar{K} = \phi$, then for $\hat{z} > 0$, we have $A_{J\theta}\,\hat{z} \geq 0$, which implies that $A_{J\theta}$ is a vertical block P_0-matrix (see [8]). Hence A is a vertical block M_0-matrix. If $\bar{K} \neq \phi$, by our Lemma 4.2 $A_{\bar{L}K} = 0$, where $L = \cup_{j\in K} J_j \subseteq J$ and $\bar{L} = \cup_{j\in\bar{K}} J_j \subseteq J$, and it is easy to verify that $A_{J\theta}$ is a vertical block P_0-matrix if and only if $A_{\bar{L}\bar{K}}$ is a P_0-matrix. We also note that $|\bar{K}| < |\theta|$. Hence the above algorithm verifies whether A is a vertical block M_0-matrix or not in at most k repetitions of Algorithm CD. Hence our theorem follows. □

Acknowledgments. The authors wish to thank the unknown referees who have patiently gone through more than one revision of this paper and whose suggestions have improved its presentation and readability considerably.

REFERENCES

[1] R. CHANDRASEKARAN, *A special case of the complementary pivot problem*, Opsearch, 7 (1970), pp. 263–268.

[2] R. W. COTTLE AND G. B. DANTZIG, *A generalization of the linear complementarity problem*, J. Combin. Theory, 8 (1970), pp. 79–90.

[3] R. W. COTTLE, J. S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[4] R. W. COTTLE AND A. F. VEINOTT, JR., *Polyhedral sets having a least element*, Math. Programming, 3 (1972), pp. 238–249.

[5] C. W. CRYER, *The method of Christopherson for solving free boundary problems for infinite journal bearings by means of finite differences*, Math. Comput., 25 (1971), pp. 435–443.

[6] B. C. EAVES, *The linear complementarity problem*, Management Sci., 17 (1971), pp. 612–634.

[7] A. A. EBIEFUNG AND M. M. KOSTREVA, *Z-matrices and the Generalized Linear Complementarity Problem*, Technical report #608, Department of Mathematical Sciences, Clemson University, Clemson, SC, 1991.

[8] ———, *Generalized P_0- and Z-matrices*, Linear Algebra Appl., 195 (1993), pp. 165–179.
[9] ———, *The generalized Leontief input-output model and its application to the choice of new technology*, Ann. Oper. Res., 44 (1993), pp. 161–172.
[10] M. FIEDLER AND V. PTÁK, *On matrices with non-positive off-diagonal elements and positive principal minors*, Czechoslovak Math. J., 12 (1962), pp. 382–400.
[11] M. S. GOWDA AND R. SZNAJDER, *A Generalization of the Nash equilibrium theorem on bimatrix games*, Research Report No.93-11, Department of Mathematics, University of Maryland, Baltimore, MD, 1993.
[12] G. HADLEY, *Linear Programming*, Addison-Wesley Publishing Co., Reading, MA, 1962.
[13] S. R. MOHAN, *On the simplex method and a class of linear complementarity problems*, Linear Algebra Appl., 14 (1976), pp. 1–9.
[14] K. G. MURTY, *Linear Programming*, John Wiley, New York, 1983.
[15] ———, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, West Berlin, 1988.
[16] K. P. OH, *The formulation of the mixed lubrication problem as a generalized nonlinear complementarity problem*, Trans. ASME J. Tribology, 108 (1986), pp. 598–604.
[17] K. G. RAMAMURTHY, *A polynomial algorithm for testing the nonnegativity of principal minors of Z-matrices*, Linear Algebra Appl., 83 (1986), pp. 39–47.
[18] R. SAIGAL, *Lemke's algorithm and a special linear complementarity problem*, Opsearch, 8 (1971), pp. 201–208.
[19] M. SUN, *Monotonicity of Mangasarian's iterative algorithm for generalized linear complementarity problems*, J. Math. Anal. Appl., 144 (1989), pp. 474–485.
[20] ———, *Singular control problems in bounded intervals*, Stochastics, 21 (1987), pp. 303–344.
[21] R. SZNAJDER, *Degree-theoretic Analysis of the Vertical and Horizontal Linear Complementarity Problems*, Ph.D. dissertation, Department of Mathematics, University of Maryland, Baltimore, MD, 1994.
[22] A. TAMIR, *An application of Z-matrices to a class of resource allocation problems*, Management Sci., 23 (1976), pp. 317–323.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1007–1024, November 1996
© 1996 Society for Industrial and Applied Mathematics

SEMIDEFINITE PROGRAMMING: A PATH-FOLLOWING ALGORITHM FOR A LINEAR–QUADRATIC FUNCTIONAL*

LEONID FAYBUSOVICH†

Abstract. We construct a class of path-following algorithms for solving semidefinite problems with linear–quadratic functionals. Complexity estimates similar to the best known for the case of the standard convex quadratic programming problem are obtained. Complete proofs of all results are included.

Key words. quadratic programming, interior-point methods, semidefinite programming

AMS subject classifications. 90C25, 34C35

1. Introduction. Let $S(n)$ be the vector space of symmetric n-by-n matrices with real entries. We consider the standard scalar product

$$\langle A, B\rangle = \mathrm{Tr}(AB), \tag{1.1}$$

where $A, B \in S(n)$ and $\mathrm{Tr}(X)$ stands for the trace of the matrix X. We use standard notations $K \geq 0$ (respectively, $K > 0$) for the nonnegative-definite (respectively, positive-definite) symmetric matrix K. Let $Q : S(n) \to S(n)$ be a symmetric nonnegative-definite linear operator, i.e., $\langle Q(A), B\rangle = \langle A, Q(B)\rangle, \langle Q(A), A\rangle \geq 0$ for any $A, B \in S(n)$. We consider the problem of the minimization of a quadratic functional of the following form:

$$f(K) = \langle C, K\rangle + \frac{1}{2}\langle Q(K), K\rangle \to \min, \tag{1.2}$$

$$\langle A_i, K\rangle = b_i, \qquad i = 1, \ldots, m, \tag{1.3}$$

$$K \in S(n), \qquad K \geq 0. \tag{1.4}$$

Here $C, A_i, i = 1, \ldots, m$, are symmetric n-by-n matrices and $b_i, i = 1, \ldots, m$ are real numbers. Throughout this paper we will assume that the set P determined by constraints (1.3),(1.4) is compact and has a nonempty interior, i.e., $\mathrm{int}(P) = \{K \in P : K > 0\}$ is not empty. We will use the notation ∂P for $P \setminus \mathrm{int}(P)$. We will assume that the matrices $A_1, \ldots, A_m$ are linearly independent over the field of real numbers R.

In this paper we present two versions of path-following algorithms for solving (1.2)–(1.4). More precisely, we will consider a path $K(\beta), \beta \geq 0$ in $\mathrm{int}(P)$ formed by optimal solutions of the family of problems:

$$f_\beta(K) = \beta\left(\langle C, K\rangle + \frac{1}{2}\langle K, Q(K)\rangle\right) - \ln\det(K) \to \min, \tag{1.5}$$

$$K \in \mathrm{int}(P). \tag{1.6}$$

* Received by the editors July 6, 1994; accepted for publication (in revised form) June 13, 1995. This research was supported in part by a grant from the Gustaf Sigurd Magnuson Foundation of Sweden and by NSF grant DMS 94-23279 while the author visited the Division of Optimization and System Theory of the Royal Institute of Technology in Stockholm, Sweden.

† Department of Mathematics, University of Notre Dame, Mail Distribution Center, Notre Dame, IN 46556-5683.

Here $\beta \geq 0$ is a real parameter. Observe, first of all, that $\ln \det(K)$ is a self-concordant barrier function [14] on the cone of nonnegative-definite symmetric matrices. Hence, there is a standard way of constructing path-following algorithms for the problem (1.2)–(1.4). Here we consider a different approach. Namely, we show that a version of a path-following algorithm for the standard convex quadratic programming problem considered in [7] can be generalized to the semidefinite situation. This approach has certain advantages. In particular, it enables us to take into account the concrete structure of the problem. As a result, many estimates are sharper in comparison with the general Nesterov–Nemirovsky scheme (e.g., the estimate of the size of the domain of the quadratic rate of convergence for the Newton method). We refer to [7] where such a comparison was made for the standard quadratic programming problem. A relative simplification of the analysis in [7] is based on the remarkable idea of Roos and Vial [18]. See also a review paper [10]. In the context of the semidefinite programming this idea can be explained in the following way. The Hessian of the barrier function $\ln \det(K)$ defines a Riemannian metric on the manifold of positive-definite symmetric matrices. This metric admits a transitive group of linear isometries. As a result, the vector fields arising as gradients of cost functions relative to this metric (generalized affine–scaling vector fields), possess certain invariance properties relative to this group action. This enables one to reduce many considerations to the situation where $K = E$ (the identity matrix) where the metric is Euclidean. We refer to [8] where this point of view was considered in detail. The linear–quadratic semidefinite programming problem (1.2)–(1.4) has numerous applications to control theory [5], [12]. Other important sources of applications include subgradient projection methods and recursive quadratic programming methods for problems involving semidefinite constraints. We refer to [1], [4], [8], [2] for the interior-point approaches to solving the linear semidefinite programming problem and to [3], [16], [13], [17] for eigenvalue problems. For a complete bibliography we refer to the forthcoming review paper [19]. Observe that the standard convex quadratic programming problem is a particular case of (1.2)–(1.4). In this case our algorithm and many proofs practically coincide with those in [7]. Our parametrization of the central path is different. We decided to include all proofs in the paper. Thus the paper is practically self-contained. The only substantial prerequisite is the knowledge of basic properties of Fréchet derivatives.

2. Duality, central trajectories, and dynamical systems. Consider the following optimization problem:

$$h(K, Z, \lambda) = \sum_{i=1}^{m} b_i \lambda_i - \frac{1}{2}\langle K, Q(K)\rangle \to \max, \tag{2.1}$$

$$\sum_{i=1}^{m} \lambda_i A_i - Q(K) + Z = C, \quad Z \geq 0, \tag{2.2}$$

$$K, \quad Z \in S(n), \qquad \lambda = (\lambda_1, \ldots \lambda_m) \in R^m. \tag{2.3}$$

LEMMA 2.1. *Suppose that* $K \in P$ *and* $(\bar{K}, Z, \lambda)$ *satisfies* (2.2), (2.3). *Then*

$$f(K) - h(\bar{K}, Z, \lambda) = \frac{1}{2}\langle K - \bar{K}, Q(K - \bar{K})\rangle + \langle Z, K\rangle \geq 0. \tag{2.4}$$

Proof. Since $b_i = \langle A_i, K\rangle, i = 1, \ldots, m$, we have

$$\Delta = f(K) - h(\bar{K}, Z, \lambda) \tag{2.5}$$

$$= \left\langle C - \sum_{i=1}^{m} \lambda_i A_i, K \right\rangle + \frac{1}{2}(\langle K, Q(K)\rangle + \langle \bar{K}, Q(\bar{K})\rangle).$$

By (2.2),

$$\left\langle C - \sum_{i=1}^{m} \lambda_i A_i, K \right\rangle = \langle Z, K\rangle - \langle Q(\bar{K}), K\rangle.$$

Substituting this into (2.5), we obtain (2.4). □

Remark 2.2. The problem (2.1)–(2.3) is usually called the dual of (1.2)–(1.4). As will follow from our results, the standard duality properties hold for the pair (1.2)–(1.4), (2.1)–(2.3) of dual problems.

Consider now the problem (1.5),(1.6). Observe that

$$Df_\beta(K) \bullet \xi = \langle \beta(C + Q(K)), \xi\rangle - \langle K^{-1}, \xi\rangle, \tag{2.6}$$

$$D^2 f_\beta(K)(\xi, \eta) = \langle \beta Q(\xi), \eta\rangle + \langle K^{-1}\xi, K^{-1}\eta\rangle. \tag{2.7}$$

Here $D^k f_\beta(K)(\xi_1, \ldots, \xi_k)$ stands for the kth Fréchet derivative of f_β at the point K evaluated at the point $(\xi_1, \ldots, \xi_k) \in S(n) \times \cdots \times S(n)$. We can conclude from (2.7) that f_β is a strictly convex function on the set of positive-definite symmetric matrices. Introduce a family of Riemannian metrics

$$g_\beta(K; \xi, \eta) = D^2 f_\beta(K)(\xi, \eta), \qquad \xi, \eta \in S(n), \tag{2.8}$$

for any $\beta \geq 0$.

PROPOSITION 2.3. *For any* $\beta \geq 0$ *the problem* (1.5), (1.6) *has a unique solution* $K(\beta) \in \operatorname{int}(P)$.

Proof. Let $\bar{K} \in \operatorname{int}(P)$ and $f_\beta(\bar{K}) = \sigma$. Consider $P_\sigma = \{K \in \operatorname{int}(P) : f_\beta(K) \leq \sigma\}$. It is clear that $\inf\{f_\beta(K) : K \in \operatorname{int}(P)\} = \inf\{f_\beta(K) : K \in P_\sigma\}$. Observe now that P_σ is closed in $S(n)$. Indeed, let $K_i, i \in N$, be a sequence of points in P_σ and $K_i \to \tilde{K}, i \to \infty$. If $\tilde{K} \in \partial P$, then we should have $\lim f_\beta(K_i) = +\infty, i \to +\infty$. But $f_\beta(K_i) \leq \sigma$ for all i, which is a contradiction. Hence, $\tilde{K} \in \operatorname{int}(P)$. We conclude that $f_\beta(\tilde{K}) = \lim f_\beta(K_i), i \to +\infty$. Consequently, $f_\beta(\tilde{K}) \leq \sigma$, i.e., $\tilde{K} \in P_\sigma$. Since $P_\sigma \subset P$ and P is compact, it is clear that $\min\{f_\beta(K) : K \in P_\sigma\}$ is attained at some point $K(\beta) \in P_\sigma$. The minimum is unique since f_β is strictly convex on $\operatorname{int}(P)$. □

Remark 2.4. The curve $K(\beta), \beta \geq 0$, is usually called the central trajectory.

Let us introduce some further notation. Set

$$T = span_R(A_1, \ldots, A_m), \tag{2.9}$$

$$T^\perp = \{K \in S(n) : \langle A_i, K\rangle = 0, i = 1, \ldots, m\}. \tag{2.10}$$

It is clear that

$$S(n) = T \oplus T^\perp.$$

Denote by π the orthogonal projection of $S(n)$ onto $T^\perp$ along T. The point $K(\beta)$ is uniquely determined by the condition

$$\nabla f_\beta(K(\beta)) \in T; \tag{2.11}$$

i.e.,

$$\pi(\beta C + \beta Q(K(\beta)) - K(\beta)^{-1}) = 0. \tag{2.12}$$

Remark 2.5. For a differentiable function φ defined on an open neighborhood of P in $S(n)$, the Euclidean gradient $\nabla\varphi(K), K \in P$, is uniquely determined by the condition

$$D\varphi(K) \bullet \xi = \langle \nabla\varphi(K), \xi \rangle, \qquad \xi \in S(n). \tag{2.13}$$

For any $\beta \geq 0$ consider the map $\psi_\beta : \mathrm{int}(P) \to T^\perp$:

$$\psi_\beta(K) = \pi(\beta Q(K) - K^{-1}). \tag{2.14}$$

The map ψ_β is a natural generalization for the case of a nonlinear cost function of the version of the Legendre transform considered in [6]. For details related to the semidefinite case, see [9]. We will use ψ_β for the analysis of various properties of the central trajectory. We can rewrite (2.12) in the form

$$\psi_\beta(K(\beta)) = -\beta\pi(C), \qquad \beta \geq 0. \tag{2.15}$$

PROPOSITION 2.6. *For each $\beta \geq 0$, the map ψ_β is a smooth isomorphism of* $\mathrm{int}(P)$ *onto $T^\perp$ such that the function $(\beta, K) \to \psi_\beta^{-1}(K) : (0, +\infty) \times T^\perp \to \mathrm{int}(P)$ is smooth.*

Proof. Indeed, for any $C \in T^\perp$ the problem

$$-\langle C, K \rangle + \frac{\beta}{2}\langle Q(K), K \rangle - \ln\det(K) \to \min, \qquad K \in \mathrm{int}(P), \tag{2.16}$$

has a unique solution $K_C \in \mathrm{int}(P)$ and

$$\psi_\beta(K_C) = C$$

(see Proposition 2.3 and (2.15)). Hence, ψ_β is surjective. On the other hand, if $\psi_\beta(K_1) = \psi_\beta(K_2) = C, K_1, K_2 \in \mathrm{int}(P)$, then again by (2.15) both K_1 and K_2 are solutions to (2.16). Hence, by Proposition 2.3, $K_1 = K_2$. In other words, ψ_β is injective. We must now prove that ψ_β^{-1} is smooth. Observe that for every $\xi \in T^\perp, K \in \mathrm{int}(P)$,

$$D\psi_\beta(K) \bullet \xi = \pi(\beta Q(\xi) + K^{-1}\xi K^{-1}). \tag{2.17}$$

Let $D\psi_\beta(K) \bullet \xi = 0$. This means

$$\beta Q(\xi) + K^{-1}\xi K^{-1} = \sum_{i=1}^{m} \lambda_i A_i$$

for some $\lambda_i \in R$. Hence, taking into account $\langle A_i, \xi \rangle = 0, i = 1, \ldots, m$, we have

$$\beta\langle Q(\xi), \xi \rangle + \langle K^{-1}\xi, K^{-1}\xi \rangle = 0.$$

Since $\beta \geq 0, \langle Q(\xi), \xi \rangle \geq 0$, we conclude that $K^{-1}\xi = 0$ or $\xi = 0$. In other words, $D\psi_\beta(K)$ defines an injective (and hence a bijective) map from $T^\perp$ onto $T^\perp$. Hence, by the inverse function theorem applied to the function $(\beta, K) \to (\beta, \psi_\beta(K)) : (0, +\infty) \times$

$\operatorname{int}(P) \to (0,+\infty) \times T^{\perp}$, the function $(\beta, K) \to \psi_{\beta}^{-1}(K)$ is a smooth function of the pair $(\beta, K) \in (0,+\infty) \times T^{\perp}$. □

COROLLARY 2.7. *$K(\beta)$ is a smooth function of β.*

Proof. By (2.15), $K(\beta) = \psi_{\beta}^{-1}(-\beta\pi(C))$. The result follows by Proposition 2.6. □

Consider the family $\mathcal{L}_{\beta,K}$ of symmetric positive-definite operators on $S(n)$:

$$\mathcal{L}_{\beta,K}(\xi) = \beta Q(\xi) + K^{-1}\xi K^{-1}. \tag{2.18}$$

Here $\beta \geq 0$ and K is a positive-definite symmetric matrix. We clearly have (see (2.8))

$$g_{\beta}(K;\xi,\eta) = \langle \mathcal{L}_{\beta,K}(\xi), \eta \rangle, \tag{2.19}$$

$\beta \geq 0, K > 0, \xi, \eta \in S(n)$. We can think of $\operatorname{int}(P)$ as a submanifold of the manifold of positive-definite symmetric matrices. Thus each metric g_{β} induces a Riemannian metric on $\operatorname{int}(P)$. Given a smooth function φ defined on an open neighborhood of $\operatorname{int}(P)$ in $S(n)$, denote by $\nabla_{\beta}\varphi(K) \in T^{\perp}, K \in \operatorname{int}(P)$, the gradient of the restriction of φ to the manifold $\operatorname{int}(P)$ with respect to the metric g_{β}. In other words, for any $\xi \in T^{\perp}, K \in \operatorname{int}(P)$,

$$D\varphi(K) \bullet \xi = \langle \nabla\varphi(K), \xi \rangle = g_{\beta}(K;\xi, \nabla_{\beta}\varphi(K)), \tag{2.20}$$

where $\nabla_{\beta}\varphi(K) \in T^{\perp}$.

PROPOSITION 2.8.

$$\nabla_{\beta}\varphi(K) = \mathcal{L}_{\beta,K}^{-1}(\nabla\varphi(K) - \Lambda(K)), \tag{2.21}$$

where $\Lambda(K) \in T$ is uniquely defined by conditions

$$\langle \nabla_{\beta}\varphi(K), A_i \rangle = 0, \qquad i = 1, \ldots, m.$$

More precisely, if

$$\Lambda(K) = \sum_{i=1}^{m} \lambda_i(K) A_i,$$

then $\lambda(K) = (\lambda_1(K), \ldots, \lambda_m(K))$ is a unique solution to the system of linear equations

$$\langle \mathcal{L}_{\beta,K}^{-1}(A_j), \nabla\varphi(K) \rangle = \sum_{i=1}^{m} \lambda_i(K) \langle \mathcal{L}_{\beta,K}^{-1}(A_i), A_j \rangle, \tag{2.22}$$

where $j = 1, \ldots, m$.

Proof. We have for $K \in \operatorname{int}(P), \xi \in T^{\perp}$,

$$g_{\beta}(K;\xi, \nabla_{\beta}\varphi(K)) = \langle \mathcal{L}_{\beta,K}(\xi), \mathcal{L}_{\beta,K}^{-1}(\nabla\varphi(K) - \Lambda(K)) \rangle = \langle \xi, \nabla\varphi(K) \rangle - \langle \xi, \Lambda(K) \rangle.$$

But $\langle \xi, \Lambda(K) \rangle = 0$, since $\xi \in T^{\perp}, \Lambda(K) \in T$. □

COROLLARY 2.9. *Given a smooth function φ defined on an open neighborhood of $\operatorname{int}(P)$ in $S(n)$, we have*

$$D\psi_{\beta}(K) \bullet \nabla_{\beta}\varphi(K) = \pi(\nabla\varphi(K)). \tag{2.23}$$

Proof. By (2.17), $D\psi_{\beta}(K) \bullet \xi = \pi(\mathcal{L}_{\beta,K}(\xi))$. In particular, by (2.21),

$$D\psi_{\beta}(K) \bullet \nabla_{\beta}\varphi(K) = \pi(\mathcal{L}_{\beta,K}(\mathcal{L}_{\beta,K}^{-1}(\nabla\varphi(K) - \Lambda(K)))) = \pi(\nabla\varphi(K)),$$

since $\pi(\Lambda(K)) = 0 (\Lambda(K) \in T)$. □

The next proposition describes a differential equation which has the central trajectory as its solution.

PROPOSITION 2.10. *Let* $\varphi(K) = -f(K) = -\langle C, K\rangle - \frac{1}{2}\langle Q(K), K\rangle$. *Then*

$$\frac{dK}{d\beta}(\beta) = \nabla_\beta \varphi(K(\beta)). \tag{2.24}$$

Proof. Let $\alpha(\beta) = \psi_\beta(K(\beta))$. We have

$$\frac{d\alpha}{d\beta}(\beta) = D\psi_\beta(K(\beta)) \bullet \frac{dK(\beta)}{d\beta} + \pi(Q(K(\beta))).$$

On the other hand, by (2.15),

$$\frac{d\alpha}{d\beta}(\beta) = -\pi(C).$$

Hence,

$$D\psi_\beta(K(\beta)) \bullet \frac{dK(\beta)}{d\beta} = -\pi(C + Q(K(\beta))). \tag{2.25}$$

Comparing this with (2.23) and using the fact that $D\psi_\beta(K)$ is bijective on $T^\perp$ (see Proposition 2.6), we arrive at (2.24). □

COROLLARY 2.11. *Let* $f(K) = \langle C, K\rangle + \frac{1}{2}\langle Q(K), K\rangle$. *Then the function* $f(K(\beta))$ *is a monotonically decreasing function of* β *for* $\beta \geq 0$.

Proof. We have

$$\begin{aligned}\frac{d}{d\beta}(f(K(\beta))) &= \left\langle \nabla f(K(\beta)), \frac{dK}{d\beta}(\beta)\right\rangle = g_\beta\left(K(\beta); \nabla_\beta f(K(\beta)), \frac{dK}{d\beta}(\beta)\right)\\ &= -g_\beta(K(\beta); \nabla_\beta f(K(\beta)), \nabla_\beta f(K(\beta))) \leq 0.\end{aligned}$$

In the last equality we used (2.24). □

Consider the following optimization problem:

$$\chi_\beta(K, Z, \lambda) = \beta\left(\sum_{i=1}^m \lambda_i b_i - \frac{1}{2}\langle Q(K), K\rangle\right) + \ln\det(Z) \to \max, \tag{2.26}$$

$$Z = C + Q(K) - \sum_{i=1}^m \lambda_i A_i > 0. \tag{2.27}$$

Since χ_β is a concave function of its variables, we clearly have the following necessary and sufficient conditions for the optimal solution of (2.26), (2.27):

$$\frac{\partial \chi_\beta}{\partial \lambda_i}(K, Z, \lambda) = \beta b_i - \langle A_i, Z^{-1}\rangle = 0, \qquad i = 1, \ldots, m, \tag{2.28}$$

$$\frac{\partial \chi_\beta}{\partial K}(K, Z, \lambda) = Q(Z^{-1} - \beta K) = 0, \qquad Z > 0. \tag{2.29}$$

Let $K(\beta)$ be a point on the central trajectory with $\beta > 0$. By (2.12),

$$\frac{K(\beta)^{-1}}{\beta} = C + Q(K(\beta)) - \sum_{i=1}^m \lambda_i(\beta) A_i$$

for some smooth functions $\lambda_i(\beta)$. Set $Z(\beta) = K(\beta)^{-1}/\beta$. Since $(K(\beta), Z(\beta), \lambda(\beta))$ satisfies (2.27)–(2.29), we conclude that this triple is an optimal solution to (2.26), (2.27). Recall (see (2.1)) that we defined the function $h(K, Z, \lambda)$ as follows:

$$h(K, Z, \lambda) = \sum_{i=1}^{m} b_i \lambda_i - \frac{1}{2}\langle Q(K), K\rangle.$$

COROLLARY 2.12.

$$f(K(\beta)) - h(K(\beta), Z(\beta), \lambda(\beta)) = \frac{n}{\beta}. \tag{2.30}$$

If K^* is a solution to (1.2)–(1.4), *then* $\lim f(K(\beta)) = f(K^*), \beta \to +\infty$.

Proof. Observe that $(K(\beta), Z(\beta), \lambda(\beta))$ is a feasible solution to (2.1)–(2.3). Hence, by Lemma 2.1,

$$f(K(\beta)) - h(K(\beta), Z(\beta), \lambda(\beta)) = \langle Z(\beta), K(\beta)\rangle = \frac{n}{\beta}. \tag{2.31}$$

Since $f(K(\beta)) \geq f(K^*) \geq h(K(\beta), Z(\beta), \lambda(\beta))$, again by Lemma 2.1 we conclude that $\lim f(K(\beta)), \beta \to +\infty$, exists and is equal to $f(K^*)$. Observe that the existence of this limit follows from the fact that $f(K(\beta))$ is a decreasing function of β (see Corollary 2.11). □

PROPOSITION 2.13. *The function $\alpha(\beta) = h(K(\beta), Z(\beta), \lambda(\beta))$ is a monotonically increasing function of β.*

Proof. Set $\Delta(\beta) = \chi_\beta(K(\beta), Z(\beta), \lambda(\beta)), \beta > 0$. Let $\beta_2 > \beta_1 > 0$. Since $(K(\beta), Z(\beta), \lambda(\beta))$ is an optimal solution to (2.26), (2.27), we have

$$\Delta(\beta_1) \geq \chi_{\beta_1}(K(\beta_2), Z(\beta_2), \lambda(\beta_2)),$$
$$\Delta(\beta_2) \geq \chi_{\beta_2}(K(\beta_1), Z(\beta_1), \lambda(\beta_1)),$$

or (since $\chi_\beta = \beta h + \ln\det(Z)$)

$$\beta_1\alpha(\beta_1) + \ln\det Z(\beta_1) \geq \beta_1\alpha(\beta_2) + \ln\det Z(\beta_2),$$
$$\beta_2\alpha(\beta_2) + \ln\det Z(\beta_2) \geq \beta_2\alpha(\beta_1) + \ln\det Z(\beta_1).$$

Adding these two inequalities, we obtain

$$(\beta_2 - \beta_1)(\alpha(\beta_2) - \alpha(\beta_1)) \geq 0.$$

Since $\beta_2 > \beta_1$, the above inequality implies $\alpha(\beta_2) \geq \alpha(\beta_1)$. □

COROLLARY 2.14. *If $\beta_2 > \beta_1 > 0$, then*

$$f(K(\beta_1)) - f(K(\beta_2)) \leq \frac{n}{\beta_1} - \frac{n}{\beta_2}. \tag{2.32}$$

Proof. By Proposition 2.13, $h(K(\beta_2), Z(\beta_2), \lambda(\beta_2)) \geq h(K(\beta_1), Z(\beta_1), \lambda(\beta_1))$. Hence,

$$f(K(\beta_1)) - f(K(\beta_2))$$
$$\leq [f(K(\beta_1)) - h(K(\beta_1), Z(\beta_1), \lambda(\beta_1))] - [f(K(\beta_2)) - h(K(\beta_2), Z(\beta_2), \lambda(\beta_2))].$$

The result follows from (2.31). □

Earlier (see (2.18)–(2.20)) we introduced a family of Riemannian metrics g_β on the manifold $\mathrm{int}(P)$ and corresponding gradient vector fields $\nabla_\beta\varphi$ (see (2.21)) of smooth functions on $\mathrm{int}(P)$. We now introduce quantities $\delta_\beta(K;\beta_1)$ which denote the length of certain gradient vector fields measured relative to the metric g_β. These quantities will play the role of the proximity measure of a given point $K \in \mathrm{int}(P)$ to the point $K(\beta)$ on the central trajectory.

Recall (see (1.5)) that

$$f_\beta(K) = \beta\left(\langle C, K\rangle + \frac{1}{2}\langle Q(K), K\rangle\right) - \ln\det(K),$$

where $K > 0$. Given $K \in \mathrm{int}(P), \beta, \beta_1 \geq 0$, set

$$\delta_\beta(K;\beta_1) = g_\beta(K; \nabla_\beta f_{\beta_1}(K), \nabla_\beta f_{\beta_1}(K))^{1/2}. \tag{2.33}$$

We first of all give a very useful characterization of $\delta_0(K;\beta_1)$.

PROPOSITION 2.15. *Let φ be a smooth function defined on an open neighborhood of* $\mathrm{int}(P)$. *Then for any $K \in \mathrm{int}(P)$,*

$$\begin{aligned} l_K(\varphi) &= g_0(K; \nabla_0\varphi(K), \nabla_0\varphi(K))^{1/2} \\ &= \min\{\| K^{1/2}(\nabla\varphi(K) - M)K^{1/2} \|_F : M \in T\}. \end{aligned} \tag{2.34}$$

Remark 2.16. Given n-by-n square matrix A, we denote by $\|A\|_F$ the Frobenius norm of A; i.e.,

$$\| A \|_F = \mathrm{Tr}(AA^T)^{1/2}. \tag{2.35}$$

Proof. We know (see Proposition 2.8) that $\nabla_0\varphi(K) = K(\nabla\varphi(K) - \Lambda(K))K$, where $\Lambda(K) \in T$ is uniquely determined by the condition $\nabla_0\varphi(K) \in T^\perp$. Given $K \in \mathrm{int}(P)$, consider the following optimization problem:

$$\gamma(M) = \langle \eta(M), \eta(M)\rangle \to \min,$$

$$\eta(M) = K^{1/2}(\nabla\varphi(K) - M)K^{1/2}, \qquad M \in T.$$

The optimality condition for this problem is $D\gamma(M) \bullet \xi = 0$ for every $\xi \in T$ or $\langle K^{1/2}\eta(M)K^{1/2}, K^{1/2}\xi K^{1/2}\rangle = 0$ for every $\xi \in T$. In other words, $K(\nabla\varphi(K) - M)K \in T^\perp$. Hence (see Proposition 2.8), $M = \Lambda(K)$ is the solution to our problem. We have

$$l_K(\varphi) = \| K^{1/2}(\nabla\varphi(K) - \Lambda(K))K^{1/2} \|_F = \gamma(\Lambda(K))^{1/2}. \qquad \square$$

3. A large-step path-following algorithm. We start with some auxiliary results needed to evaluate the total number of Newton steps in a path-following algorithm.

LEMMA 3.1. *Let $\langle,\rangle_1, \langle,\rangle_2$ be two Euclidean scalar products on the vector space V such that $\langle v, v\rangle_1 \leq \langle v, v\rangle_2$ for any $v \in V$. Suppose that φ is a smooth function on V and $\nabla_1\varphi(v), \nabla_2\varphi(v)$ are gradients of φ relative to $\langle,\rangle_1, \langle,\rangle_2$, respectively. Then*

$$\langle \nabla_1\varphi(v), \nabla_1\varphi(v)\rangle_1 \geq \langle \nabla_2\varphi(v), \nabla_2\varphi(v)\rangle_2 \tag{3.1}$$

for any $v \in V$.

Proof. By the definition of the gradient relative to a metric, we have

$$D\varphi(v) \bullet \xi = \langle \nabla_1 \varphi(v), \xi \rangle_1 = \langle \nabla_2 \varphi(v), \xi \rangle_2$$

for any $v, \xi \in V$. In particular, for $\xi = \nabla_2 \varphi(v)$, we obtain

$$\begin{aligned} \langle \nabla_2 \varphi(v), \nabla_2 \varphi(v) \rangle_2 &= \langle \nabla_1 \varphi(v), \nabla_2 \varphi(v) \rangle_1 \\ &\leq \langle \nabla_1 \varphi(v), \nabla_1 \varphi(v) \rangle_1^{1/2} \langle \nabla_2 \varphi(v), \nabla_2 \varphi(v) \rangle_1^{1/2} \\ &\leq \langle \nabla_1 \varphi(v), \nabla_1 \varphi(v) \rangle_1^{1/2} \langle \nabla_2 \varphi(v), \nabla_2 \varphi(v) \rangle_2^{1/2}. \end{aligned}$$

Here the first inequality is the Cauchy–Schwarz inequality for the scalar product $\langle , \rangle_1$ and the second inequality follows from the imposed condition $\langle \xi, \xi \rangle_1 \leq \langle \xi, \xi \rangle_2$ with $\xi = \nabla_2 \varphi(v)$. The result follows. □

If we apply Lemma 3.1 to Euclidean scalar products $g_0(K;,), g_\beta(K;,), \beta \geq 0$, on $T^\perp$ and $\varphi = f_{\beta_1}$, we immediately obtain the following result.

COROLLARY 3.2.

$$\delta_0(K;\beta_1) \geq \delta_\beta(K;\beta_1) \tag{3.2}$$

for any $K \in \operatorname{int}(P), \beta, \beta_1 \geq 0$.

Let us introduce "the Newton direction":

$$p_\beta(K) = -\nabla_\beta f_\beta(K), \qquad \beta \geq 0, K \in \operatorname{int}(P). \tag{3.3}$$

By Proposition 2.8, $p_\beta(K)$ admits the following characterization:

$$p_\beta(K) = -\mathcal{L}_{\beta,K}^{-1}(\beta C + \beta Q(K) - K^{-1} - \Lambda(K)), \tag{3.4}$$

where $\Lambda(K) \in T$ is uniquely determined by the condition $p_\beta(K) \in T^\perp$ (see (2.21)).

LEMMA 3.3.

$$\delta_\beta(K;\beta)^2 = -\langle p_\beta(K), \nabla f_\beta(K) \rangle. \tag{3.5}$$

Proof. By (2.33), (3.4)

$$\begin{aligned} \delta_\beta(K;\beta)^2 &= \langle p_\beta(K), \mathcal{L}_{\beta,K}(p_\beta(K)) \rangle = \langle p_\beta(K), -\nabla f_\beta(K) + \Lambda(K) \rangle \\ &= -\langle p_\beta(K), \nabla f_\beta(K) \rangle. \end{aligned}$$

The last equality follows from the fact that $p_\beta(K) \in T^\perp, \Lambda(K) \in T$. □

PROPOSITION 3.4. *Given* $K \in \operatorname{int}(P), \beta \geq 0$, *we have*

$$\| K^{-1/2} p_\beta(K) K^{-1/2} \|_F^2 \leq \delta_\beta(K;\beta)^2 \leq \delta_0(K;\beta)^2. \tag{3.6}$$

Proof. Indeed, by (2.33),

$$\begin{aligned} \delta_\beta(K;\beta)^2 &= \langle p_\beta(K), \mathcal{L}_{\beta,K}(p_\beta(K)) \rangle = \langle p_\beta(K), K^{-1} p_\beta(K) K^{-1} + \beta Q(p_\beta(K)) \rangle \\ &\geq \langle p_\beta(K), K^{-1} p_\beta(K) K^{-1} \rangle = \| K^{-1/2} p_\beta(K) K^{-1/2} \|_F^2. \end{aligned}$$

Further, $\delta_\beta(K;\beta) \leq \delta_0(K;\beta)$ by Corollary 3.2. □

LEMMA 3.5. *Let* $K > 0$ *and* $p \in S(n)$ *be such that* $\| K^{-1/2} p K^{-1/2} \|_F < 1$. *Then* $K + p > 0$.

Proof. Set, in this proof, $K^{-1/2}pK^{-1/2} = D$. We have

$$K + p = K^{1/2}(E + D)K^{1/2}. \tag{3.7}$$

Observe that $\mathrm{Tr}(D^2) < 1$ implies that the eigenvalues $\mu_1, \dots, \mu_n$ of D satisfy

$$\mu_1^2 + \cdots + \mu_n^2 < 1.$$

From this and (3.7) the result immediately follows. □

The next proposition gives an estimate for the size of the domain of the quadratic convergence for the Newton method.

PROPOSITION 3.6. *Given* $K \in \mathrm{int}(P), \beta \geq 0$, *if* $\|K^{-1/2}p_\beta(K)K^{-1/2}\|_F < 1$, *then* $K_+ = K + p_\beta(K) \in \mathrm{int}(P)$. *Moreover,*

$$\begin{aligned}\|K_+^{-1/2}p_\beta(K_+)K_+^{-1/2}\|_F &\leq \delta_\beta(K_+;\beta) \leq \delta_0(K_+;\beta) \\ &\leq \|K^{-1/2}p_\beta(K)K^{-1/2}\|_F^2 \leq \delta_\beta(K;\beta)^2 \leq \delta_0(K;\beta)^2.\end{aligned} \tag{3.8}$$

Proof. Set, in this proof $p_\beta(K) = p$. Observe that by Lemma 3.5, $K_+ > 0$. Since $p \in T^\perp$, we have $\langle A_i, K_+\rangle = \langle A_i, K\rangle, i = 1, \dots, m$. Hence, $K_+ \in \mathrm{int}(P)$. By Proposition 2.15,

$$\delta_0(K_+;\beta) = \min\{\| K_+^{1/2}(\beta C + \beta Q(K_+) - K_+^{-1} - M)K_+^{1/2} \|_F \colon M \in T\}.$$

Take $M = \Lambda(K)$, where $\Lambda(K)$ is defined in (3.4). From (3.4),

$$\Lambda(K) = K^{-1}pK^{-1} + \beta Q(p) + \beta C + \beta Q(K) - K^{-1}.$$

Hence,

$$\beta C + \beta Q(K_+) - K_+^{-1} - \Lambda(K) = K^{-1} - K_+^{-1} - K^{-1}pK^{-1}.$$

Here we used

$$Q(K_+) = Q(K + p) = Q(K) + Q(p).$$

Hence,

$$\begin{aligned}\delta_0(K_+;\beta)^2 &\leq \| K_+^{1/2}(K^{-1} - K_+^{-1} - K^{-1}pK^{-1})K_+^{1/2} \|_F^2 \\ &= \mathrm{Tr}([K_+(K^{-1} - K_+^{-1} - K^{-1}pK^{-1})]^2) \\ &= \mathrm{Tr}([E + pK^{-1} - E - pK^{-1} - pK^{-1}pK^{-1}]^2) = \mathrm{Tr}([pK^{-1}pK^{-1}]^2).\end{aligned}$$

Here we used $K_+ = K + p$. Further,

$$\begin{aligned}\mathrm{Tr}([pK^{-1}pK^{-1}]^2) &= \mathrm{Tr}(pK^{-1}pK^{-1}pK^{-1}pK^{-1}) \\ &= \mathrm{Tr}[(K^{-1/2}pK^{-1/2})^4] = \|(K^{-1/2}pK^{-1/2})^2\|_F^2 \leq \|K^{-1/2}pK^{-1/2}\|_F^4.\end{aligned}$$

In other words,

$$\delta_0(K_+;\beta) \leq \|K^{-1/2}pK^{-1/2}\|_F^2.$$

The rest immediately follows by Proposition 3.4. □

PROPOSITION 3.7. *Let $K \in \operatorname{int}(P)$ be such that $K + p_\beta(K) \in \operatorname{int}(P)$ for some $\beta \geq 0$. Then*

$$f_\beta(K) - f_\beta(K + \bar{\alpha} p_\beta(K)) \geq \delta_\beta(K;\beta) - \ln(1 + \delta_\beta(K,\beta)), \tag{3.9}$$

where $\bar{\alpha} = (1 + \delta_\beta(K;\beta))^{-1}$.

Proof. Set, in this proof, $p_\beta(K) = p, \delta_\beta(K;\beta) = \delta$. Consider the function $\varphi(\alpha) = f_\beta(K + \alpha p)$. The Taylor series for φ near $\alpha = 0$ has the form

$$\varphi(\alpha) = f_\beta(K) + \alpha\langle \nabla f_\beta(K), p\rangle + \frac{1}{2}\alpha^2 g_\beta(K;p,p) + \sum_{k=3}^{\infty} t_k, \tag{3.10}$$

where

$$t_k = \frac{\alpha^k}{k!} D^k f_\beta(K)(p,\dots,p).$$

We have

$$D^k f_\beta(K)(p,\dots,p) = (-1)^k (k-1)! \operatorname{Tr}((K^{-1/2} p K^{-1/2})^k), \qquad k \geq 3.$$

Let $\mu_1,\dots,\mu_n$ be eigenvalues of the symmetric matrix $K^{-1/2} p K^{-1/2}$. We have

$$\begin{aligned} |\operatorname{Tr}((K^{-1/2} p K^{-1/2})^k)| &\leq |\mu_1|^k + \cdots |\mu_n|^k \leq (|\mu_1|^2 + \cdots + |\mu_n|^2)^{k/2} \\ &= \|K^{-1/2} p K^{-1/2}\|_F^k \leq \delta^k. \end{aligned}$$

In the last inequality we used (3.6). Hence, taking into account $\langle \nabla f_\beta(K), p\rangle = -\delta^2$ (see Lemma 3.3), we obtain

$$f_\beta(K + \alpha p) \leq f_\beta(K) + \left(\frac{1}{2}\alpha^2 - \alpha\right)\delta^2 + \sum_{k=3}^{\infty} \frac{\alpha^k}{k}\delta^k = f_\beta(K) - \alpha\delta^2 - \ln(1 - \alpha\delta) - \alpha\delta.$$

Consequently,

$$f_\beta(K) - f_\beta(K + \alpha p) \geq \alpha(\delta^2 + \delta) + \ln(1 - \alpha\delta). \tag{3.11}$$

For $\alpha = \bar{\alpha}$ we obtain (3.9). □

COROLLARY 3.8. *Under the assumptions of Proposition 3.7, if $0 < \delta_\beta(K;\beta) < \frac{1}{2}$, then*

$$f_\beta(K + p_\beta(K)) < f_\beta(K).$$

Proof. Indeed, by (3.11),

$$f_\beta(K + p_\beta(K)) \leq f_\beta(K) - \delta^2 - \delta - \ln(1 - \delta),$$

where $\delta = \delta_\beta(K;\beta)$. The function $\delta^2 + \delta + \ln(1 - \delta)$ is monotonically increasing for $0 < \delta < \frac{1}{2}$ and is equal to zero at zero. The result follows. □

The next proposition states the convergence of the Newton method for the function f_β.

PROPOSITION 3.9. *Suppose that $K \in \operatorname{int}(P)$ and $\delta_\beta(K;\beta) < 1$. Consider the sequence $K_0 = K, K_1 = K_0 + p_\beta(K_0), \dots, K_i = K_{i-1} + p_\beta(K_{i-1}), i \geq 2$. Then $K_i \in \operatorname{int}(P)$ for any i and, moreover, $\lim K_i = K(\beta), i \to +\infty$.*

Proof. Since

$$\delta_\beta(K_i;\beta) \le \delta_\beta(K_0;\beta)^{2^i}, \tag{3.12}$$

we first of all conclude by Lemma 3.5 and Proposition 3.4 that $K_i \in \operatorname{int}(P)$ for all i. Moreover, $\delta_\beta(K_i;\beta) < \frac{1}{2}$ for all sufficiently large i. Consequently, by Corollary 3.8 $f_\beta(K_{i+1}) \le f_\beta(K_i)$ for sufficiently large i. Hence, $\sup\{f_\beta(K_i) : i = 0,1,\ldots\} = \gamma < +\infty$. Let $\overline{K}$ be a limit point of the sequence K_i. Then $f_\beta(\overline{K}) \in \operatorname{int}(P)$, since $f_\beta(K) \to +\infty, K \to \partial P$. By (3.12), $\delta_\beta(K_i;\beta) \to 0, i \to +\infty$. We have $\delta_\beta(\overline{K};\beta) = 0, \overline{K} \in \operatorname{int}(P)$. By (2.33) and Proposition 2.8, we have $\nabla f_\beta(\overline{K}) \in T$, which implies (see (2.11)) that $\overline{K} = K(\beta)$. Hence, $K(\beta)$ is the only limit point of the sequence $K_i, i \ge 0$.

Observe that at least one limit point exists, since $K_i \in P$ for all i and P is compact. We conclude $K_i \to K(\beta), i \to +\infty$. □

PROPOSITION 3.10. *Suppose* $K \in \operatorname{int}(P), \delta_\beta(K;\beta) < 1$. *Then*

$$f_\beta(K) - f_\beta(K(\beta)) \le \frac{\delta_\beta(K;\beta)^2}{1-\delta_\beta(K;\beta)^2}. \tag{3.13}$$

Proof. We set, in this proof, $\delta = \delta_\beta(K;\beta)$. Since f_β is convex on $\operatorname{int}(P)$, we have

$$f_\beta(K + p_\beta(K)) - f_\beta(K) \ge \langle \nabla f_\beta(K), p_\beta(K)\rangle.$$

By Lemma 3.3 we obtain

$$f_\beta(K) - f_\beta(K + p_\beta(K)) \le \delta^2. \tag{3.14}$$

Set $K_0 = K$ and let $K_1, K_2, \ldots$ be the sequence obtained by repeating Newton steps:

$$K_{i+1} = K_i + p_\beta(K_i).$$

By Proposition 3.9, $K_i \to K(\beta), i \to +\infty$. By Proposition 3.6,

$$\delta_\beta(K_i;\beta) \le \delta^{2^i}, \qquad i = 0,1,\ldots. \tag{3.15}$$

Hence,

$$\begin{aligned} f_\beta(K) - f_\beta(K(\beta)) &= \sum_{j=0}^{\infty}[f_\beta(K_j) - f_\beta(K_{j+1})] \le \sum_{j=0}^{\infty}\delta_\beta(K_j;\beta)^2 \\ &\le \sum_{j=0}^{\infty}\delta^{2^{j+1}} \le \frac{\delta^2}{1-\delta^2}. \end{aligned}$$

Here the first inequality follows from (3.14) and the second inequality is a consequence of (3.15). □

PROPOSITION 3.11. *If* $\beta > 0, \delta = \delta_\beta(K;\beta) < 1$, *then*

$$f(K) - f(K(\beta)) \le \frac{\delta(1+\delta)}{1-\delta}\frac{\sqrt{n}}{\beta}. \tag{3.16}$$

Proof. In this proof we set $\delta_\beta(K;\beta) = \delta, p_\beta(K) = p$. Recall (see (1.2)) that

$$f(K) = \langle C, K\rangle + \frac{1}{2}\langle Q(K), K\rangle.$$

Using the convexity of f, we obtain

$$\langle \nabla f(K+p), p\rangle \geq f(K+p) - f(K) \geq \langle \nabla f(K), p\rangle. \tag{3.17}$$

Further,

$$\nabla f(K) = C + Q(K) = \frac{1}{\beta}(\nabla f_\beta(K) + K^{-1}).$$

Hence,

$$\langle \nabla f(K), p\rangle = \frac{\langle \nabla f_\beta(K), p\rangle}{\beta} + \frac{\langle K^{-1}, p\rangle}{\beta}. \tag{3.18}$$

Now by Lemma 3.3, $\langle \nabla f_\beta(K), p\rangle = -\delta^2$. Further,

$$\langle K^{-1}, p\rangle = \operatorname{Tr}(K^{-1}p) = \operatorname{Tr}(K^{-1/2}pK^{-1/2}).$$

By the Cauchy–Schwarz inequality

$$|\langle K^{-1}, p\rangle| \leq \|K^{-1/2}pK^{-1/2}\|_F \|E\|_F = \|K^{-1/2}pK^{-1/2}\|_F \sqrt{n}. \tag{3.19}$$

Thus from (3.18),

$$\langle \nabla f(K), p\rangle \geq \frac{-(\delta^2 + \sqrt{n}\delta)}{\beta} \geq \frac{-\sqrt{n}\delta(1+\delta)}{\beta}. \tag{3.20}$$

Here we used $\|K^{-1/2}pK^{-1/2}\|_F \leq \delta$ (see Proposition 3.4). On the other hand,

$$\langle \nabla f(K+p), p\rangle = \langle C, p\rangle + \langle p, Q(K+p)\rangle.$$

By (3.4),

$$-\beta Q(p) - K^{-1}pK^{-1} = \beta C + \beta Q(K) - K^{-1} - \Lambda(K), \qquad \Lambda(K) \in T.$$

Hence,

$$C + Q(K+p) = \frac{1}{\beta}(K^{-1} + \Lambda(K) - K^{-1}pK^{-1}).$$

Thus,

$$\langle \nabla f(K+p), p\rangle = \frac{\langle K^{-1}, p\rangle - \langle K^{-1}pK^{-1}, p\rangle}{\beta}.$$

Here we used $\langle \Lambda(K), p\rangle = 0$. Now $\langle K^{-1}pK^{-1}, p\rangle = \|K^{-1/2}pK^{-1/2}\|_F^2$ and

$$|\langle p, K^{-1}\rangle| \leq \delta_\beta(K;\beta)\sqrt{n}$$

(see 3.19). We conclude that

$$\langle \nabla f(K+p), p\rangle \leq \frac{\delta\sqrt{n}}{\beta}. \tag{3.21}$$

Combining (3.20), (3.21), and (3.17), we obtain

$$|f(K) - f(K+p)| \leq \frac{\delta(1+\delta)\sqrt{n}}{\beta}. \tag{3.22}$$

Consider the sequence $K_0 = K, K_1 = K_0 + p, \ldots$ of Newton iterates. By Proposition 3.9, $K_i \to K(\beta), \beta \to +\infty$. Now

$$|f(K) - f(K(\beta))| = |\sum_{i=0}^{\infty}(f(K_i) - f(K_{i+1}))| \leq \sum_{i=0}^{\infty}|f(K_i) - f(K_{i+1})|$$

$$\leq \sum_{i=0}^{\infty}\frac{\delta_\beta(K_i;\beta)\sqrt{n}(1+\delta_\beta(K_i;\beta))}{\beta} \leq \frac{\sqrt{n}}{\beta}(1+\delta_\beta(K;\beta))\sum_{i=0}^{\infty}\delta^{2^i} \leq \frac{\sqrt{n}(1+\delta)\delta}{\beta(1-\delta)}.$$

Here we used $\delta_\beta(K_i;\beta) \leq \delta^{2^i}$ (see Proposition 3.6). □

We are now in position to describe a version of the long-step path-following algorithm (compare with [7]). Suppose we are given $\beta_0 > 0$ and $K_0 \in \text{int}(P)$ such that $\delta_{\beta_0}(K_0;\beta_0) \leq 1/2$. Given $\theta > 0$, we perform what are called outer iterations: $\beta_1 = (1+\theta)\beta_0, \ldots, \beta_i = (1+\theta)^i\beta_0$. Suppose that we are able to construct a sequence of points $K_1, K_2, \ldots$ in $\text{int}(P)$ such that $\delta_{\beta_i}(K_i;\beta_i) \leq \frac{1}{2}$.

THEOREM 3.12. *Suppose that $\epsilon > 0$ is given and*

$$i \geq \frac{\ln(\frac{4n}{\epsilon\beta_0})}{\ln(1+\theta)}. \tag{3.23}$$

Then

$$f(K_i) - f(K^*) \leq \epsilon,$$

where K^* is an optimal solution to the problem (1.2)–(1.4).

Proof. Condition (3.23) is equivalent to

$$\frac{n}{\beta_i} \leq \frac{\epsilon}{4}. \tag{3.24}$$

We have

$$f(K_i) - f(K^*) = [f(K_i) - f(K(\beta_i))] + [f(K(\beta_i)) - f(K^*)].$$

By Proposition 3.11,

$$f(K_i) - f(K(\beta_i)) \leq \frac{\delta(1+\delta)}{1-\delta}\frac{\sqrt{n}}{\beta_i},$$

where $\delta = \delta_{\beta_i}(K_i;\beta_i) \leq \frac{1}{2}$. Hence,

$$f(K_i) - f(K(\beta_i)) \leq \frac{3\sqrt{n}}{2\beta_i}. \tag{3.25}$$

Here we use that $\delta \to \frac{\delta(1+\delta)}{1-\delta}$ is a monotonically increasing function of δ for $0 < \delta < 1$. Further,

$$f(K(\beta_i)) - f(K^*) \leq \frac{n}{\beta_i} \tag{3.26}$$

by (2.30). Combining (3.24)–(3.26), we obtain

$$f(K_i) - f(K^*) \leq \frac{\frac{3}{2}\sqrt{n}+n}{\beta_i} \leq \frac{\epsilon}{4n}\left(\frac{3}{2}\sqrt{n}+n\right) \leq \epsilon. \qquad \square$$

We must now describe a procedure for updating K_i. Suppose we are given $K_i \in \operatorname{int}(P)$ such that $\delta_{\beta_i}(K_i;\beta_i) \leq \frac{1}{2}$. Set $\beta_{i+1} = (1+\theta)\beta_i$. We have to find $K_{i+1} \in \operatorname{int}(P)$ such that $\delta_{\beta_{i+1}}(K_{i+1};\beta_{i+1}) \leq \frac{1}{2}$. To do that we perform several Newton steps for the function $f_{\beta_{i+1}}$ starting from the point K_i. Each such Newton step is called the inner iteration.

THEOREM 3.13. *Each outer iteration requires at most*

$$\frac{11}{3} + 11\theta\left(\frac{3}{2}\sqrt{n} + \frac{\theta n}{\theta+1}\right)$$

inner iterations.

Proof. We start with a point $K_i \in \operatorname{int}(P)$ such that $\delta_{\beta_i}(K_i;\beta_i) \leq \frac{1}{2}$. Observe that, as it easily follows from Lemma 3.5, the point

$$\overline{K} = K_i + \bar{\alpha} p_{\beta_{i+1}}(K_i), \qquad \bar{\alpha} = \frac{1}{1+\delta_{\beta_{i+1}}(K_i;\beta_{i+1})},$$

belongs to $\operatorname{int}(P)$. By Proposition 3.7,

$$\begin{aligned} f_{\beta_{i+1}}(K_i) - f_{\beta_{i+1}}(\overline{K}) &\geq \delta_{\beta_{i+1}}(K_i,\beta_{i+1}) - \ln(1+\delta_{\beta_{i+1}}(K_i,\beta_{i+1})) \\ &\geq \frac{1}{2} - \ln\left(\frac{3}{2}\right) > \frac{1}{11}, \end{aligned} \tag{3.27}$$

provided $\delta_{\beta_{i+1}}(K_i;\beta_{i+1}) \geq \frac{1}{2}$. If $\delta_{\beta_{i+1}}(K_i,\beta_{i+1}) < \frac{1}{2}$, set $K_{i+1} = K_i$. We continue to perform inner iterations described above until $\delta_{\beta_{i+1}}(K_i;\beta_{i+1}) < \frac{1}{2}$. By (3.27) the value of the function $f_{\beta_{i+1}}$ is decreased by $\frac{1}{11}$ after such an iteration. If N is the number of such iterations, we obviously have

$$\frac{N}{11} \leq f_{\beta_{i+1}}(K_i) - f_{\beta_{i+1}}(K(\beta_{i+1})). \tag{3.28}$$

Let

$$\eta(K,\beta) = f_\beta(K) - f_\beta(K(\beta)).$$

We wish to estimate $\eta(K_i,\beta_{i+1})$. By the mean value theorem

$$\eta(K,\beta_{i+1}) = \eta(K,\beta_i) + \frac{\partial\eta}{\partial\beta}(K,\hat{\beta})\theta\beta_i \tag{3.29}$$

for some $\hat{\beta} \in (\beta_i,\beta_{i+1})$. We have

$$\frac{\partial\eta}{\partial\beta}(K,\beta) = f(K) - f(K(\beta)) - \left\langle \nabla f_\beta(K(\beta)), \frac{dK(\beta)}{d\beta}\right\rangle.$$

(Recall that $f_\beta(K) = \beta f(K) - \ln\det(K)$.) But

$$\frac{dK(\beta)}{d\beta} \in T^\perp, \quad \nabla f_\beta(K(\beta)) \in T$$

(see (2.11)). Hence,

$$\frac{\partial \eta}{\partial \beta}(K,\beta) = f(K) - f(K(\beta)). \tag{3.30}$$

Combining (3.29), (3.30) for $K = K_i$, we obtain

(3.31)
$$\begin{aligned}\eta(K_i,\beta_{i+1}) - \eta(K_i,\beta_i) &= \theta\beta_i(f(K_i) - f(K(\hat{\beta}))) \le \theta\beta_i(f(K_i) - f(K(\beta_{i+1})))\\ &\le \theta\beta_i(|\, f(K_i) - f(K(\beta_i))\,| + |\, f(K(\beta_i)) - f(K(\beta_{i+1}))\,|),\end{aligned}$$

where in the first inequality we used the fact that $f(K(\beta))$ is a monotonically decreasing function of β (see Corollary 2.11). By Proposition 3.10,

$$\eta(K_i,\beta_i) = f_{\beta_i}(K_i) - f_{\beta_i}(K(\beta_i)) \le \frac{(\frac{1}{2})^2}{1-(\frac{1}{2})^2} = \frac{1}{3}. \tag{3.32}$$

(Recall that $\delta_{\beta_i}(K_i,\beta_i) \le \frac{1}{2}$.) By Proposition 3.11,

$$|f(K_i) - f(K(\beta_i))| \le \frac{\frac{1}{2}\frac{3}{2}}{\frac{1}{2}}\frac{\sqrt{n}}{\beta_i} = \frac{3}{2}\frac{\sqrt{n}}{\beta_i}. \tag{3.33}$$

By (2.32),

$$|f(K(\beta_i)) - f(K(\beta_{i+1}))| \le \frac{n}{\beta_i} - \frac{n}{\beta_{i+1}} = \frac{\theta n}{\beta_{i+1}}. \tag{3.34}$$

Combining (3.31)–(3.34), we finally obtain

$$\begin{aligned}f_{\beta_{i+1}}(K_i) - f_{\beta_{i+1}}(K(\beta_{i+1})) &\le \frac{1}{3} + \theta\beta_i\left(\frac{3}{2}\frac{\sqrt{n}}{\beta_i} + \frac{\theta n}{\beta_{i+1}}\right)\\ &= \frac{1}{3} + \theta\left(\frac{3\sqrt{n}}{2} + \frac{\theta n}{\theta+1}\right).\end{aligned} \tag{3.35}$$

Hence, by (3.35), (3.28),

$$N \le \frac{11}{3} + 11\theta\left(\frac{3\sqrt{n}}{2} + \frac{\theta n}{\theta+1}\right). \qquad \square$$

From Theorems 3.12 and 3.13 we immediately obtain the following result.

THEOREM 3.14. *An upper bound for the total number of Newton iterations is given by*

$$\frac{\ln(\frac{4n}{\epsilon\beta_0})}{\ln(1+\theta)}\left(\frac{11}{3} + 11\theta\left(\frac{3\sqrt{n}}{2} + \frac{\theta n}{\theta+1}\right)\right).$$

4. A Shor step algorithm. In this section we briefly consider the situation where the number of inner iterations (per one outer iteration) does not exceed 1.

PROPOSITION 4.1. *Let* $\beta_+ = (1+\vartheta)\beta, \vartheta > 0, \beta > 0, K \in \operatorname{int}(P)$. *Then*

$$\delta_0(K;\beta_+) \le (1+\theta)\delta_0(K;\beta) + \theta\sqrt{n}. \tag{4.1}$$

Proof. According to Proposition 2.15,

$$\delta_0(K;\beta_+) = \min\{\| \beta_+ K^{1/2}(C + Q(K) - M)K^{1/2} - E \|_F : M \in T\}. \tag{4.2}$$

Let

$$\nabla_0 f_\beta = \beta K(C + Q(K) - \Lambda(K))K - K \tag{4.3}$$

for some $\Lambda(K) \in T$ (see (2.21)). We conclude from (4.2), (4.3) that

$$\begin{aligned}\delta_0(K,\beta_+) &\le \|\beta_+ K^{1/2}(C + Q(K) - \Lambda(K))K^{1/2} - E\|_F \\ &= \|(1+\theta)[\beta K^{1/2}(C + Q(K) - \Lambda(K))K^{1/2} - E] + \theta E\|_F \le (1+\theta)\delta_0(K,\beta) + \theta\sqrt{n}. \ \square\end{aligned}$$

PROPOSITION 4.2. *Under the assumptions of Proposition* 4.1 *let* $K_+ = K + p_\beta(K)$ *and* $\theta = \frac{3}{32\sqrt{n}}$. *If* $\delta_0(K;\beta) \le \frac{1}{2}$, *then* $\delta_0(K_+;\beta_+) \le \frac{1}{2}$.

Proof. Due to Proposition 4.1,

$$\begin{aligned}\delta_0(K;\beta_+) &\le \frac{1+\theta}{2} + \theta\sqrt{n} \le \frac{1}{2}\left(1 + \frac{3}{32\sqrt{n}}\right) + \frac{3}{32} \\ &\le \frac{35+6}{64} = \frac{41}{64} < \frac{11}{16}.\end{aligned}$$

Now by Proposition 3.6,

$$\delta_0(K_+;\beta_+) \le \delta_0(K;\beta_+)^2 \le \frac{121}{256} < \frac{1}{2}. \qquad \square$$

5. Concluding remarks. In the present paper we have considered large-step and short-step versions of a path-following algorithm for the semidefinite programming problem with an arbitrary (convex) quadratic functional. We have shown that the results for the standard quadratic problem can be extended without essential changes to this more general situation. While our barrier function is self-concordant we never explicitly used this fact. One of the most important contributions of the work of Nesterov and Nemirovsky is the construction of the so-called barrier calculus. In their fundamental book [14] they in particular show how to obtain the barrier function $\ln\det(K)$ from general principles. One can then apply a general scheme to obtain various versions of path-following algorithms. However, very frequently, the use of the concrete structure of a problem enables one to improve the general results. In particular, this is the case in our situation. For example, the conic reformulation of the problem considered is certainly possible and may be useful (e.g., for the construction of primal–dual algorithms). However, in our opinion, it is quite unnatural and computationally expensive in the situation considered. The performance of the Newton step is, of course, the most expensive part of any path-following algorithm. In many applications it is possible to use the concrete structure of the problem to perform this step. If one uses various artificial transformations, this structure may be lost, which causes computational problems. This gives an additional reason to keep the structure of the problem in constructing path-following algorithms.

The results of this paper admit a further generalization to the case of an arbitrary self-dual homogeneous cone. The barrier function should be chosen as a properly normalized logarithm of the characteristic function of the cone [11]. This requires, however, a quite advanced version of the technique of self-concordant barriers [15].

REFERENCES

[1] F. ALIZADEH, *Optimization over the positive-definite cone: Interior-point methods and combinatorial applications,* in Advances in Optimization and Parallel Computing, P. Pardalos, ed., North-Holland, Amsterdam, 1992.

[2] F. ALIZADEH, J.-P.A. HAEBERLY, AND M. L. OVERTON, *Primal-Dual Interior-Point Methods for Semidefinite Programming,* preprint, 1994.

[3] S. BOYD AND L. EL GHAOUI, *Method of centers for minimizing generalized eigenvalues,* Linear Algebra Appl., 188, 189 (1993), pp. 63–111.

[4] S. BOYD AND L. VANDENBERGHE, *Primal-dual potential reduction method for problems involving matrix inequalities,* Math. Programming, 69 (1995), pp. 205–236.

[5] S. BOYD, L. EL GHAOUI, E. FERON, AND V. BALAKRISHNAN, *Linear Matrix Inequalities in System and Control Theory,* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[6] D. A. BAYER AND J. C. LAGARIAS, *The nonlinear geometry of linear programming* I, Trans. Amer. Math. Soc., 314 (1989), pp. 499–526.

[7] D. DEN HERTOG, *Interior Point Approach to Linear, Quadratic and Convex Programming,* Kluwer Academic Publishers, Norwell, MA, 1994.

[8] L. FAYBUSOVICH, *On a matrix generalization of affine-scaling vector fields,* SIAM J. Matrix Anal. Appl., 16 (1995), pp. 886–897.

[9] ———, *A Hamiltonian structure for generalized affine-scaling vector fields,* J. Nonlinear Sci., 5 (1995), pp. 11–28.

[10] C. GONZAGA, *Path-following methods for linear programming,* SIAM Rev., 34 (1992), pp. 167–224.

[11] O. GULER, *Barrier Functions in Interior Point Methods,* Math. Oper. Res., to appear.

[12] U. HELMKE AND J. MOORE, *Optimization and Dynamical Systems,* Springer, Berlin, New York, 1994.

[13] F. JARRE, *Interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices,* SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[14] YU. E. NESTEROV AND A. S. NEMIROVSKY, *Interior Point Polynomial Algorithms in Convex Programming,* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[15] YU. NESTEROV AND M. TODD, *Self-Scaled Cones and Interior-Point Methods in Nonlinear Programming,* Math. Oper. Res., to appear.

[16] M. L. OVERTON AND R. S. WOMERSLEY, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices,* Math. Programming, 62 (1993), pp. 321–357.

[17] F. RENDL, R. VANDERBEI, AND H. WOLKOWITZ, *A Primal-Dual Interior-Point Method for the Max-Min Eigenvalue Problem,* Tech. Report, University of Waterloo, Ontario, Canada, 1993.

[18] C. ROOS AND J.-PH. VIAL, *A polynomial method of approximate centers for linear programming,* Math. Programming, 54 (1992), pp. 295–305.

[19] L. VANDENBERGHE AND S. BOYD, *Semidefinite programming,* SIAM Rev., 38 (1996), pp. 49–95.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1025–1039, November 1996
© 1996 Society for Industrial and Applied Mathematics
008

ANALYSIS OF A SYMMETRIC RANK-ONE TRUST REGION METHOD*

RICHARD H. BYRD†, HUMAID FAYEZ KHALFAN‡, AND ROBERT B. SCHNABEL†

Abstract. Several recent computational studies have shown that the symmetric rank-one (SR1) update is a very competitive quasi-Newton update in optimization algorithms. This paper gives a new analysis of a trust region SR1 method for unconstrained optimization and shows that the method has an $n+1$ step q-superlinear rate of convergence. The analysis makes neither of the assumptions of uniform linear independence of the iterates nor positive definiteness of the Hessian approximations that have been made in other recent analyses of SR1 methods. The trust region method that is analyzed is fairly standard, except that it includes the feature that the Hessian approximation is updated after all steps, including rejected steps. We also present computational results that show that this feature, safeguarded in a way that is consistent with the convergence analysis, does not harm the efficiency of the SR1 trust region method.

Key words. quasi-Newton method, symmetric rank-one update, superlinear convergence

AMS subject classifications. 65, 49

1. Introduction.

The symmetric rank-one (SR1) secant update has been shown to be very effective in optimization calculations, especially when used in conjunction with a trust region method. The theoretical understanding of its convergence behavior on nonquadratic problems is still rather incomplete, however. This paper improves that understanding by providing a local convergence analysis of a trust region, SR1 method for unconstrained optimization under fairly mild assumptions. In doing so it extends the work in Conn, Gould, and Toint [1991] and in Khalfan, Byrd, and Schnabel [1993]. The new results are particularly interesting, first because a trust region method can be argued to be the appropriate context for the SR1 and second because the trust region structure allows some of the less desirable assumptions made by Khalfan, Byrd, and Schnabel [1993] in analyzing the SR1 for a line search method to be relaxed. In particular, no assumption about the positive definiteness of the Hessian approximations is made.

In solving the nonlinear unconstrained minimization problem

$$\text{minimize } f(x),\ x \in R^n \tag{1.1}$$

the SR1 update can be used to approximate the Hessian matrix of the objective function $\nabla^2 f(x)$. Given an approximation B_k to the Hessian, the SR1 formula updates it by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}, \tag{1.2}$$

where s_k is a step and $y_k = \nabla f(x_k + s_k) - \nabla f(x_k)$. Minimization algorithms using this update in both a line search and a trust region context have been shown in

* Received by the editors July 30, 1993; accepted for publication (in revised form) May 12, 1995. This research was supported by Air Force Office of Scientific Research grants AFOSR-90-0109 and F49620-94-1-0101, Army Research Office grants DAAL 03-91-G-0151 and DAAH04-94-0228, and National Science Foundation grant CCR-9101795.

† Department of Computer Science, Campus Box 430, University of Colorado, Boulder, CO 80309 (richard@hessian.cs.colorado.edu, bobby@hessian.cs.colorado.edu).

‡ Department of Mathematics and Computer Science, United Arab Emirates University, Al-Ain, United Arab Emirates.

computational experiments by Conn, Gould, and Toint [1988, 1992] and by Khalfan, Byrd, and Schnabel [1993] to be competitive with methods using the widely accepted Broyden–Fletcher–Goldfarb–Shanno (BFGS) update. The convergence of such algorithms is not as well understood, however, as convergence of the BFGS method. For example, no convergence results that are similar to the global results for the line search BFGS method given by Powell [1975] or the local superlinear results of Broyden, Dennis, and Moré [1973] exist, so far, for the SR1 method when applied to a nonquadratic function. A significant difference between the BFGS and SR1 updates that contributes to this situation is that although the BFGS algorithm is guaranteed to produce a positive definite B_{k+1} if B_k is positive definite and $s_k^T y_k > 0$, the SR1 update does not have this property. In practical implementations of an SR1 method, the Hessian approximations B_k may be indefinite at some iterations.

Conn, Gould, and Toint [1991] proved that the sequence of matrices generated by the SR1 formula converges to the actual Hessian at the solution $\nabla^2 f(x_*)$, provided that the sequence of steps taken, $\{s_k\}$, is uniformly linearly independent, that the denominator in (1.2) is sufficiently different from zero, and that the iterates converge to x_*. Using this result it is simple to prove that the rate of convergence under these assumptions is q-superlinear. (Interestingly, for the BFGS method Ge and Powell [1983] proved, under a different set of assumptions, that the sequence of generated matrices converges but not necessarily to $\nabla^2 f(x_*)$.) The assumption of uniform linear independence is unusual and fairly strong, but Conn, Gould, and Toint [1991] present experiments on minimization of randomly generated quartics in which it holds. On the other hand, experiments by Khalfan, Byrd, and Schnabel [1993] showed that nearly dependent steps occurred commonly when using the SR1 in both a line search and a trust region method on a standard set of unconstrained optimization test problems.

The study of the SR1 in a line search context by Khalfan, Byrd, and Schnabel [1993] did not make a uniform linear independence assumption and instead made the weaker assumptions that the matrices B_k are positive definite and uniformly bounded. This analysis showed the convergence rate of the iterates to be $n+1$-step q-superlinear and $2n$-step q-quadratic. Although the matrices B_k did appear in experiments to stay bounded, they were not always positive definite at every iterate. In fact, an example was constructed in that paper in which B_k is indefinite at every iterate and convergence is only linear.

In this paper we show that when used with a trust region method, indefiniteness does not cause such problems to the local convergence properties of the SR1 method. In §2 we describe in detail a class of SR1 trust region methods that is fairly standard with one exception and show that the superlinear convergence result proved in Khalfan, Byrd, and Schnabel [1993] holds for this method without any assumption of positive definiteness. The one unusual feature of the method that is analyzed is that it updates the Hessian approximation B_k at all steps, including rejected steps. In §3 we briefly present computational results that show that this feature, when safeguarded in a manner that is consistent with the convergence analysis, is reasonable in practice in comparison with the more common alternative of updating only at successful steps.

We will use the symbol $\|x\|$ to denote an arbitrary norm of a vector x. When the argument is a matrix the norm is the corresponding induced matrix norm.

2. Convergence results. In this section we establish convergence properties of the SR1 update in the following general trust region algorithm. In a trust region method, an attempt is made to search for an acceptable step in a neighborhood around x_k, instead of searching only in one direction as is done in line search algorithms.

Specifically the quadratic model problem

$$\text{minimize } m(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \tag{2.1}$$

$$\text{subject to } \|s\| \leq \Delta_k \tag{2.2}$$

is solved approximately, where B_k is an approximation to the Hessian matrix and Δ_k is a positive number representing the radius of the ball around the current point in which $m(x_k + s)$ is trusted to represent the objective function. The radius of the trust region, Δ_k, is adjusted from iteration to iteration by monitoring the actual reduction in functional values,

$$\text{ared}_k = f(x_k) - f(x_k + s_k),$$

and the reduction predicted by the quadratic model,

$$\text{pred}_k = f(x_k) - m(x_k + s_k),$$

where s_k is an approximate minimizer of $m(x_k + s)$ over $\|s\| \leq \Delta_k$. If $\frac{\text{ared}_k}{\text{pred}_k}$ is close to 1, this indicates that the quadratic model represents $f(x)$ well enough and the trust region radius may be increased. If, on the other hand, this ratio is smaller than a prescribed small number, the size of the trust region is reduced. For a background on trust region algorithms and their global convergence, see, for example, Moré and Sorensen [1983] and Shultz, Schnabel, and Byrd [1985].

Below we give a formal description of the algorithm that we will analyze. It is meant to be a fairly typical trust region method, with the exception of Step 6. The step computed is a possibly approximate solution to the trust region problem, which is required to be accurate enough that for all k

$$\text{pred}_k \geq \sigma_1 \|\nabla f(x_k)\| \min\{\Delta_k, \sigma_2 \|\nabla f(x_k)\| / \|B_k\|\} \tag{2.3}$$

for positive constants σ_1 and σ_2 and such that

$$\text{whenever } \|s_k\| < 0.8\Delta_k, \text{ then } B_k s_k = -\nabla f(x_k). \tag{2.4}$$

These conditions are satisfied if the trust region problem (2.1, 2.2) is solved exactly and are satisfied by most approximate trust region strategies that are used in practice (see, e.g., Shultz, Schnabel, and Byrd [1985]). Note that in the notation used, it can happen that $x_{k+1} = x_k$ if the step s_k is rejected even though the update (1.2) is made.

As described in Khalfan, Byrd, and Schnabel [1993], we skip the update if the denominator in (1.2) is too small. Specifically, we update by (1.2) only if

$$|s_k^T (y_k - B_k s_k)| \geq r \|s_k\| \|y_k - B_k s_k\|, \tag{2.5}$$

where r is a constant $\in (0, 1)$; otherwise, we set $B_{k+1} = B_k$.

ALGORITHM 2.1.

0) Choose an initial point x_0, an initial symmetric matrix B_0, an initial trust region radius Δ_0, $\eta \in (0, 0.1)$, $\tau_1 \in (0, 1)$, $\tau_2 > 1$, and set $k = 0$.
1) Compute $f(x_k)$ and $\nabla f(x_k)$. If $\nabla f(x_k) = 0$, then stop.

2) Compute a trust region step: find a feasible approximate solution s_k to

$$\min_{s \in R^n} \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \quad \text{subject to } \|s\| \leq \Delta_k$$

such that (2.3) and (2.4) are satisfied for some positive constants, σ_1 and σ_2.

3) Compute $\text{ared}_k = f(x_k) - f(x_k + s_k)$ and $\text{pred}_k = -(\nabla f(x_k)^T s_k + \frac{1}{2} s_k^T B_k s_k)$.

4) If $\dfrac{\text{ared}_k}{\text{pred}_k} > \eta$, then
   set $x_{k+1} = x_k + s_k$;
   else set $x_{k+1} = x_k$.

5) If $\dfrac{\text{ared}_k}{\text{pred}_k} > 0.75$, then
   if $\|s_k\| < 0.8\Delta_k$, then set $\Delta_{k+1} = \Delta_k$;
   else set $\Delta_{k+1} = \tau_2 \Delta_k$;
   else
   if $0.1 \leq \dfrac{\text{ared}_k}{\text{pred}_k} \leq 0.75$, then set $\Delta_{k+1} = \Delta_k$;
   else set $\Delta_{k+1} = \tau_1 \Delta_k$.

6) Compute the next Hessian approximation B_{k+1} using the SR1 update along the direction s_k. (Observe here that the update is made even if $x_{k+1} = x_k$.)

7) Set $k = k + 1$ and go to Step 1.

Note that the trust region updating procedure in Step 5 requires increasing the trust region radius if there is good model-function agreement. Such a feature is usually present in implemented algorithms and is essential to our analysis. We specify the increase and decrease factors, τ_2 and τ_1, to be constants. Actually it is only essential to our results that the decrease factor be bounded away from zero, and the increase factor be bounded above, and bounded below by a number greater than one. However, considering such possibilities would complicate some proofs significantly. Also for simplicity we have specified reasonable numerical values, 0.1 and 0.75, for the algorithmic constants used in deciding when to decrease or increase the trust region radius, but in our analysis any values in (0,1) will work. We have not included the empirically efficient device of internal doubling (i.e., letting $x_{k+1} = x_k$ while increasing the radius when model-function agreement is very good) in order to keep the analysis simpler, even though such a device could be handled by similar analysis.

It is essential to the convergence analysis that even when $x_{k+1} = x_k$, the matrix B_k is updated along the failed direction s_k to get B_{k+1}. Analyses of Powell–Symmetric–Broyden (PSB) trust region methods by Powell [1970, 1975] make a similar provision, as does Khalfan [1989] in analyzing a related method. As far as we know these are the only other analyses proving superlinear convergence of a trust region secant method. Such updates along failed directions seem to be necessary to the convergence analysis because if the Hessian approximation is incorrect along such a direction and is not updated along that direction very similar directions could, in principle, be generated repeatedly at later iterations. Such steps would be rejected, resulting in the trust region being reduced at these iterations, potentially keeping it small enough that it would interfere with making a superlinear step even if the Hessian approximation is accurate enough to provide one. In §3 we present some numerical experiments that assess the effects and costs of this strategy of updating at failed points. It is possible to include safeguards on this strategy that do not make updates at "very bad" failed points which do not interfere with the theoretical results

in this paper. We have included such a safeguard in our implementation in §3 but for simplicity have not used it in the algorithm that we analyze.

In this section we will show that under certain assumptions the rate of convergence of Algorithm 2.1 is $n+1$-step q-superlinear. This is the same kind of superlinear convergence established for a line search SR1 method in Khalfan, Byrd, and Schnabel [1993]. As in that paper we will first show that for a large fraction of the iterates near the solution $B_k s_k$ is quite close to $\nabla^2 f(x_*)s_k$ and that if such an s_k satisfies $B_k s_k = -\nabla f(x_k)$ good progress is made toward the solution.

Throughout this section the following assumptions will frequently be made.

Assumptions.

(A1) The sequence of iterates does not terminate and remains in a closed, bounded, convex set D on which the function f is twice continuously differentiable and in which f has a unique stationary point x_*. The Hessian $\nabla^2 f(x_*)$ is positive definite, and $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x_*; that is, there exists a constant $\gamma > 0$ such that for all x, y in some neighborhood of x_*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq \gamma \|x - y\|.$$

(A2) The sequence of matrices $\{B_k\}$ is generated from each iterate x_k by the SR1 update, using s_k, and for each iteration

$$|s_k^T(y_k - B_k s_k)| \geq r\|s_k\|\|y_k - B_k s_k\|, \tag{2.6}$$

where r is a constant $\in (0,1)$.

(A3) The sequence of matrices $\{B_k\}$ is bounded by a constant M such that $\|B_k\| \leq M$ for all k.

The assumptions that the iterates remain in a bounded set and that the denominator of the SR1 update obeys (2.6) appear reasonable in practice and have been made in the analyses of Conn, Gould, and Toint [1991] and Khalfan, Byrd, and Schnabel [1993]. Assumption (A2) implies that no updates are skipped due to violation of condition (2.6). As described in Khalfan, Byrd, and Schnabel [1993], this is usually the case in practice. The assumption that $\|B_k\|$ is bounded above is made in Khalfan, Byrd, and Schnabel [1993] (although not in Conn, Gould, and Toint [1991] due to their stronger assumption of uniform linear independence) and also appears to be reasonable in practice. Note that the latter two assumptions are not needed in analyses of the BFGS method.

Note that Assumption (A1) implies that if we choose positive constants

$$\beta_1 < \frac{1}{\|\nabla^2 f(x_*)^{-1}\|} \text{ and } \beta_2 > \|\nabla^2 f(x_*)\| \tag{2.7}$$

then if $\|x_k - x_*\|$ is sufficiently small

$$\beta_1\|x_k - x_*\| \leq \|\nabla f(x_k)\| \leq \beta_2\|x_k - x_*\| \tag{2.8}$$

and

$$\frac{2}{\beta_2}[f(x_k) - f(x_*)] \leq \|x_k - x_*\|^2 \leq \frac{2}{\beta_1}[f(x_k) - f(x_*)]. \tag{2.9}$$

First we point out that these assumptions are enough to imply convergence to x_* using standard global convergence theory.

THEOREM 2.1. *If the sequence* $\{x_k\}$ *is generated by Algorithm* 2.1 *and Assumptions* (A1) *and* (A3) *hold, then* $x_k \to x_*$.

Proof. Note that Algorithm 2.1 is a trust region method of the type considered by Theorem 2.2 in Shultz, Schnabel, and Byrd [1985]. Since the assumption that $\{B_k\}$ be bounded is satisfied, it follows from that theorem that all cluster points of $\{x_k\}$ are stationary points of f. By Assumption (A1) the sequence stays in a compact set and thus has at least one cluster point, which is a stationary point. Since there is only one stationary point, x_*, the entire sequence must converge to x_*. □

Now we establish a general relation between steplength and trust region which in fact holds for essentially any trust region method, provided the iterates converge.

LEMMA 2.2. *Suppose that the sequence of iterates* $\{x_k\}$, *generated by Algorithm* 2.1, *converges to* x_*. *Then either*

$$\Delta_k \to 0 \tag{2.10}$$

or there exist $K > 0$ *and* $\Delta > 0$ *such that for all* $k > K$

$$\Delta_k = \Delta. \tag{2.11}$$

In either case $s_k \to 0$.

Proof. Let $\Delta = \liminf \Delta_k$ and suppose first that $\Delta > 0$. From Step 5 of Algorithm 2.1, Δ_k is increased only if $\|x_{k+1} - x_k\| \geq 0.8\Delta_k$. Therefore since $\|x_{k+1} - x_k\| \to 0$, there exists $K \geq 0$ such that Δ_k is not increased for any $k \geq K$. Since $\Delta > 0$, this implies that $\Delta_k \geq \Delta$ for all $k \geq K$. Also we know that for some $K_1 > K$, $\Delta_{K_1} < \frac{1}{\tau_1}\Delta$. If the trust region radius were to be decreased we would have $\Delta_{K_1+1} < \Delta$, which we have ruled out. Since neither increase nor decrease can occur we must have $\Delta_k = \Delta$ for all $k \geq K_1$.

Suppose now that $\Delta = 0$. Since $x_k \to x_*$, for every $\epsilon > 0$ there exists $K_\epsilon \geq 0$ such that $\|x_{k+1} - x_k\| < \epsilon$ for all $k \geq K_\epsilon$. Since $\liminf \Delta_k = 0$, there exists $j \geq K_\epsilon$ such that $\Delta_j < \epsilon$. But since Δ_k is increased only if $\Delta_k \leq \frac{1}{0.8}\|x_{k+1} - x_k\| < \frac{\epsilon}{0.8}$, and the increase factor is τ_2, we have that $\Delta_k < \frac{\tau_2 \epsilon}{0.8}$ for all $k \geq j$. Therefore (2.10) follows.

To show that $s_k \to 0$, note that if (2.10) is true, then clearly $s_k \to 0$. If (2.11) is true, then for all $k \geq K$, the step s_k is accepted and $s_k = \|x_{k+1} - x_k\|$. Thus $s_k \to 0$ by convergence. □

In analyzing the convergence rate of our algorithm we will make reference to the following result, which is a slight variation of Lemma 3.2 from Khalfan, Byrd, and Schnabel [1993]. It states that if the sequence of steps generated by an iterative process using the SR1 update satisfies Assumptions (A1) and (A2) and the sequence of matrices, $\{B_k\}$, is bounded, then out of any set of $n+1$ steps, B_k is accurate along at least one step.

LEMMA 2.3. *Let* $\{x_k\}$ *be a sequence of iterates which converges to the local minimizer* x_* *of* f, *and let* s_k *be a sequence of vectors such that* $x_k + s_k \to x_*$. *Suppose that Assumptions* (A1)–(A3) *hold. Then there exists* $K \geq 0$ *such that for any set of* $n+1$ *steps* $\mathcal{S} = \{s_{k_j} : K \leq k_1 < \cdots < k_{n+1}\}$ *there exists an index* k_m *with* $m \in \{2, 3, \ldots, n+1\}$ *such that*

$$\frac{\|(B_{k_m} - \nabla^2 f(x_*))s_{k_m}\|}{\|s_{k_m}\|} < \bar{c}\epsilon_{\mathcal{S}}^{\frac{1}{n}}, \tag{2.12}$$

where

$$\epsilon_{\mathcal{S}} = \max_{1 \leq j \leq n+1} \{\|x_{k_j} - x_*\|, \|x_{k_j} + s_{k_j} - x_*\|\}$$

and

$$\bar{c} = 4\left[\gamma + \sqrt{n}\frac{\gamma}{2}\left(\frac{2}{r}+1\right)^{k_{n+1}-k_1-2} + M + \|\nabla^2 f(x_*)\|\right].$$

Actually Lemma 3.2 from Khalfan, Byrd, and Schnabel [1993] assumed in addition that $x_{k+1} = x_k + s_k$, but this fact is not used except in the definition of ϵ_S, and we have taken this difference into account in defining ϵ_S in (2.12).

Applying Lemma 2.3 in a trust region context we can show that out of any set of p steps generated by Algorithm 2.1 where $p > n$ there are at least $p - n$ steps that are accepted by Algorithm 2.1, and for which the approximate Hessian is accurate.

LEMMA 2.4. *Suppose that Assumptions* (A1)–(A3) *are satisfied. Then there exists N such that for any set of $p > n$ consecutive steps $s_{k+1}, s_{k+2}, \ldots, s_{k+p}$ with $k \geq N$, there exists a set, $\mathcal{G}_k$, of at least $p-n$ indices contained in the set $\{i : k+1 \leq i \leq k+p\}$ such that for all $j \in \mathcal{G}_k$,*

$$\frac{\|(B_j - \nabla^2 f(x_*))s_j\|}{\|s_j\|} < c\epsilon_k^{\frac{1}{n}}, \tag{2.13}$$

where

$$\epsilon_k = \max_{k+1\leq j\leq k+p}\{\|x_j - x_*\|, \|x_j + s_j - x_*\|\}$$

and

$$c = 4\left[\gamma + \sqrt{n}\frac{\gamma}{2}\left(\frac{2}{r}+1\right)^{p-2} + M + \|\nabla^2 f(x_*)\|\right].$$

Furthermore, for k sufficiently large, if $j \in \mathcal{G}_k$, then

$$\|s_j\| < \frac{2\beta_2}{\beta_1}\|x_j - x_*\| \tag{2.14}$$

and

$$\frac{\text{ared}_j}{\text{pred}_j} \geq 0.75. \tag{2.15}$$

Proof. By Theorem 2.1 $x_k \to x_*$ and by Lemma 2.2 $s_k \to 0$. Therefore, by Lemma 2.3, applied to the set

$$\{s_k, s_{k+1}, \ldots, s_{k+p}\}, \tag{2.16}$$

there exists N such that for any $k \geq N$ there exists an index l_1, with $k+1 \leq l_1 \leq k+p$ satisfying

$$\|(B_{l_1} - \nabla^2 f(x_*))s_{l_1}\| < c\epsilon_k^{\frac{1}{n}}\|s_{l_1}\|, \tag{2.17}$$

where the fact that the set (2.16) consists of $p+1$ consecutive steps allows us to replace $\bar{c}$ by c. Now we can apply Lemma 2.3 to the set $\{s_k, s_{k+1}, \ldots, s_{k+p}\} - s_{l_1}$ to

get l_2. Repeating this $p-n$ times we get a set of $p-n$ indices $\mathcal{G}_k = \{l_1, l_2, \ldots, l_{p-n}\}$ such that if $j \in \mathcal{G}_k$, then

$$\|(B_j - \nabla^2 f(x_*))s_j\| < c\epsilon_k^{\frac{1}{n}}\|s_j\|. \tag{2.18}$$

Now consider $j \in \mathcal{G}_k$. By (2.7) and (2.18) for k sufficiently large $s_j^T B_j s_j > \beta_1 \|s_j\|^2$. Therefore using (2.8) with j large enough we have

$$\begin{aligned} 0 \leq \text{pred}_j &= -\nabla f(x_j)^T s_j - \frac{1}{2} s_j^T B_j s_j \\ &< \|\nabla f(x_j)\|\|s_j\| - \frac{1}{2}\beta_1 \|s_j\|^2 \\ &\leq \beta_2 \|x_j - x_*\|\|s_j\| - \frac{1}{2}\beta_1 \|s_j\|^2. \end{aligned}$$

Therefore (2.14) holds for $j \in \mathcal{G}_k$, for k sufficiently large.

Now let $j \in \mathcal{G}_k$. Then using Taylor's theorem (see, e.g., Ortega and Rheinboldt [1970]), the Lipschitz continuity of $\nabla^2 f(x)$, (2.18) and (2.14) and assuming k is large enough we have that

$$\begin{aligned} |\text{ared}_j - \text{pred}_j| &= \left| f(x_j) - f(x_j + s_j) - \left(f(x_j) - f(x_j) - \nabla f(x_j)^T s_j - \frac{1}{2} s_j^T B_j s_j \right) \right| \\ &\leq \left| \frac{1}{2} s_j^T B_j s_j - \int_0^1 s_j^T \nabla^2 f(x_j + \zeta s_j) s_j (1-\zeta) d\zeta \right| \\ &\leq \|s_j\| \int_0^1 \|(B_j - \nabla^2 f(x_j + \zeta s_j))s_j\|(1-\zeta) d\zeta \\ &\leq \|s_j\|\|(B_j - \nabla^2 f(x_*))s_j\| \\ &\quad + \|s_j\|^2 \int_0^1 \|\nabla^2 f(x_*) - \nabla^2 f(x_j + \zeta s_j)\|(1-\zeta) d\zeta \\ &\leq \|s_j\|\|(B_j - \nabla^2 f(x_*))s_j\| + \gamma \|s_j\|^2 (\|x_j - x_*\| + \|s_j\|) \\ &\leq c\epsilon_k^{\frac{1}{n}} \|s_j\|^2 + \gamma(\|s_j\|^2 \|x_j - x_*\| + \|s_j\|^3) \\ &\leq \left[c\epsilon_k^{\frac{1}{n}} + \gamma \left(1 + \frac{2\beta_2}{\beta_1} \right) \|x_j - x_*\| \right] \|s_j\|^2 \\ &\leq 2c\epsilon_k^{\frac{1}{n}} \|s_j\|^2, \end{aligned} \tag{2.19}$$

where in the last step we note that for k, and thus j, large enough $\|x_j - x_*\|/c\epsilon_k^{\frac{1}{n}}$ becomes arbitrarily small.

Since $\|s_j\| < \left(\frac{2\beta_2}{\beta_1}\right)\|x_j - x_*\| \leq \left(\frac{2\beta_2}{\beta_1^2}\right)\|\nabla f(x_j)\|$ and $\|s_j\| \leq \Delta_j$ it follows that

$$\|s_j\|^2 \leq \left(\frac{2\beta_2}{\beta_1^2}\right)\|\nabla f(x_j)\| \min\left\{\Delta_j, \left(\frac{2\beta_2}{\beta_1^2}\right)\|\nabla f(x_j)\|\right\}. \tag{2.20}$$

From the condition (2.3) on Step 2 of the algorithm, we have that

$$\text{pred}_j \geq \sigma_1 \|\nabla f(x_j)\| \min\left\{\Delta_j, \sigma_2 \frac{\|\nabla f(x_j)\|}{\|B_j\|}\right\}. \tag{2.21}$$

Therefore using (2.19), (2.20), and (2.21) we have that

$$|\text{ared}_j - \text{pred}_j| \leq \alpha \epsilon_k^{\frac{1}{n}} |\text{pred}_j|,$$

where α is a constant, which implies (2.15). □

The following lemma is a slight extension of Lemma 3.3 given in Khalfan, Byrd, and Schnabel [1993], which is a sort of single step version of the Dennis–Moré superlinear convergence condition.

LEMMA 2.5. *Suppose the function f satisfies Assumption* (A1). *If the quantities*

$$e_k = \|x_k - x_*\| \text{ and } \frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|}$$

are sufficiently small and if $B_k s_k = -\nabla f(x_k)$, then

$$\|x_k + s_k - x_*\| \leq \|\nabla^2 f(x_*)^{-1}\| \left[2\frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|}e_k + \frac{\gamma}{2}e_k^2\right] \tag{2.22}$$

and

$$h(x_k + s_k) \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\nabla^2 f(x_*)^{-1}\| \left[2\frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|}h(x_k) + \frac{\gamma}{\sqrt{2\beta_1}}h(x_k)^2\right], \tag{2.23}$$

where $h(x) = [f(x) - f(x_*)]^{\frac{1}{2}}$.

Proof. Inequality (2.22) is proved in Lemma 3.3 of Khalfan, Byrd, and Schnabel [1993], and (2.23) follows from (2.22) using (2.9) three times. □

Up to now the analysis has been quite similar to the SR1 line search analysis of Khalfan, Byrd, and Schnabel [1993]. We have shown that for many steps $B_k s_k$ is accurate and the step is accepted, and Lemma 2.5 says that when the trust region is inactive such a step results in good progress toward the solution. However, since B_k can be indefinite, we must worry about the possibility that the trust region interferes with progress toward the solution. In the following lemma we show that in fact the trust region radius is much larger than the distance to the solution (measured, for convenience, by $h(x_k)$) for all iterates with k sufficiently large, so that the trust region will not interfere with rapid progress when $B_k s_k$ is accurate. This is the case because, since at least $p - n$ out of p steps are accurate, trust radius decreases occur less often than trust radius increases and very good steps.

LEMMA 2.6. *Suppose the assumptions of Lemma* 2.4 *are satisfied and the computed step satisfies* (2.4). *Then, letting $h_k = h(x_k)$,*

$$\lim_{k\to\infty} \frac{h_k}{\Delta_k} = 0. \tag{2.24}$$

Proof. Let p be the smallest integer greater than $2n + n(-\log\tau_1/\log\tau_2)$. Then

$$\tau_1^n \tau_2^{p-2n} \geq 1, \tag{2.25}$$

and applying Lemma 2.4 with this value of p there exists N such that if $k \geq N$, then there exists a set of at least $p - n$ indices, $\mathcal{G}_k \subset \{j : k+1 \leq j \leq k+p\}$, such that if $j \in \mathcal{G}_k$, then $\|(B_j - \nabla^2 f(x_*))s_j\|/\|s_j\| < c\epsilon_k^{\frac{1}{n}}$ and

$$\frac{\text{ared}_j}{\text{pred}_j} \geq 0.75. \tag{2.26}$$

We show that this implies that for such steps

$$\frac{h_{j+1}}{\Delta_{j+1}} \leq \frac{1}{\tau_2}\frac{h_j}{\Delta_j}. \tag{2.27}$$

If $\|s_j\| \geq 0.8\Delta_j$, then since from Step 5 of Algorithm 2.1 $\Delta_{j+1} = \tau_2\Delta_j$ and since $\{h_i\}$ is decreasing, (2.27) follows. If on the other hand $\|s_j\| < 0.8\Delta_j$, then from Step 5 of Algorithm 2.1 we have that

$$\Delta_{j+1} = \Delta_j.$$

Also since the trust region is inactive, by condition (2.4) $B_j s_j = -\nabla f(x_j)$. Therefore (2.13) and (2.23) imply that if N is large enough, we have that

$$h_{j+1} \leq \frac{1}{\tau_2} h_j.$$

This implies that (2.27) is true for $j \in \mathcal{G}_k$, where $k \geq N$.

In addition, note that for any j, $h_{j+1} \leq h_j$ and $\Delta_{j+1} = \tau_1\Delta_j$ and so

$$\frac{h_{j+1}}{\Delta_{j+1}} \leq \frac{1}{\tau_1}\frac{h_j}{\Delta_j}. \tag{2.28}$$

Therefore (2.27) is true for $p - n$ values of $j \in \mathcal{G}_k$, and (2.28) holds for all j. Thus for all $k \geq N$, we have using (2.25) that

$$\frac{h_{k+p}}{\Delta_{k+p}} \leq \left(\frac{1}{\tau_1}\right)^n \left(\frac{1}{\tau_2}\right)^{p-n} \frac{h_k}{\Delta_k} \leq \left(\frac{1}{\tau_2}\right)^n \frac{h_k}{\Delta_k}. \tag{2.29}$$

Therefore, starting at $k = N$, it follows that

$$\frac{h_{N+lp}}{\Delta_{N+lp}} \to 0$$

as $l \to \infty$. Using (2.28) again, (2.24) is implied. □

Now that we know the trust region is eventually inactive at most steps, it is fairly easy to show that the rate of convergence of the sequence $\{x_k\}$ generated by Algorithm 2.1 is $n+1$-step q-superlinear.

THEOREM 2.7. *Consider Algorithm* 2.1 *satisfying* (2.3) *and* (2.4) *and suppose that Assumptions* (A1)–(A3) *hold. Then the sequence* $\{x_k\}$ *generated by Algorithm* 2.1 *is* $n+1$*-step q-superlinear; i.e.,*

$$\frac{e_{k+n+1}}{e_k} \to 0.$$

Proof. By Lemma 2.4 there exists N such that if $k \geq N$, then the set of steps $\{s_{k+1}, \ldots, s_{k+n+1}\}$ contains at least one step s_{k+j}, $1 \leq j \leq n+1$, for which

$$\|(B_{k+j} - \nabla^2 f(x_*))s_{k+j}\|/\|s_{k+j}\| < c\epsilon_k^{\frac{1}{n}}.$$

By (2.14) and (2.9)

$$\|s_{k+j}\| < \frac{2\beta_2}{\beta_1} e_{k+j} \leq \left(\frac{2}{\beta_1}\right)^{\frac{3}{2}} \beta_2 h_{k+j}.$$

Therefore by Lemma 2.6 if N is large enough and $k \geq N$, then $\|s_{k+j}\| < 0.8\Delta_{k+j}$. This implies $B_{k+j}s_{k+j} = -\nabla f(x_{k+j})$. Therefore by Theorem 2.1 $e_k \to 0$, and thus by inequality (2.23) of Lemma 2.5 if N is large enough and $k \geq N$, then

$$h_{k+j+1} = h(x_{k+j} + s_{k+j}) \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{\sqrt{2\beta_1}}h_{k+j}\right) h_{k+j}. \tag{2.30}$$

The first equality holds because (2.15) implies that the step is accepted.

Since the sequence $\{h_i\}$ is decreasing this implies that

$$h_{k+n+1} \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{\sqrt{2\beta_1}}h_k\right) h_k.$$

Therefore by (2.9) we have that

$$\begin{aligned} e_{k+n+1} &\leq \sqrt{\frac{2}{\beta_1}}h_{k+n+1} \\ &\leq \frac{\sqrt{2\beta_2}}{\beta_1}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{\sqrt{2\beta_1}}h_k\right) h_k \\ &\leq \frac{\sqrt{2\beta_2}}{\beta_1}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{2}\sqrt{\frac{\beta_2}{\beta_1}}e_k\right) \sqrt{\frac{\beta_2}{2}}e_k. \end{aligned}$$

Thus $n+1$-step q-superlinear convergence follows. □

As we noted before, this $n+1$ step q-superlinear convergence rate is the same as that proved for the SR1 with a line search by Khalfan, Byrd, and Schnabel [1993] but with a trust region method there is no assumption of positive definiteness of B_k required. We could not establish a result corresponding to the $2n$-step quadratic convergence in that paper; however, it would be possible to show that in the limit the subsequence of iterations on which superlinear convergence occurs includes almost all iterates.

It is natural to ask to what extent the matrices B_k are in fact asymptotically positive definite if Algorithm 2.1 is used. It turns out that slightly stronger conditions on the step computations are required in order to guarantee any positive semidefiniteness.

Example. Let $f(x) = x_{(1)}^2 + x_{(1)}^3 + x_{(2)}^2$ with gradient

$$\nabla f(x) = \left(2x_{(1)} + 3x_{(1)}^2,\ 2x_{(2)}\right)^T$$

so that there is a local minimizer at $x = 0$. Consider an iterate and a Hessian approximation of the form

$$x_k = \begin{bmatrix} \xi \\ 0 \end{bmatrix}, \text{ and } B_k = \begin{bmatrix} \beta & 0 \\ 0 & -1 \end{bmatrix},$$

where ξ and β are constants and $\beta > 0$. Assume Δ_k is relatively large so that the step $-B_k^{-1}\nabla f(x_k)$ satisfies (2.4) as well as (2.3) and is selected. Thus the step is

$$s = \left(-(2\xi + 3\xi^2)/\beta,\ 0\right)^T,$$

and the next iterate is

$$x_{k+1} = \left(\xi - (2\xi + 3\xi^2)/\beta,\ 0\right)^T.$$

The difference in gradients has the form

$$y = \left(y_{(1)},\ 0\right)^T.$$

By (1.2) the SR1 update gives a new Hessian approximation of the form

$$B_{k+1} = \begin{bmatrix} \beta' & 0 \\ 0 & -1 \end{bmatrix},$$

where β' is a one-dimensional secant approximation to $\frac{\partial^2 f}{\partial x_{(1)}^2}$, using $(x_k)_{(1)}$ and $(x_{k+1})_{(1)}$. Thus the method behaves like the one-dimensional secant method on $x_{(1)}$, while $x_{(2)} = 0$ and $(B_k)_{2,2} = -1$ for all iterates.

However, we can guarantee that the Hessian approximations are eventually positive definite most of the time if we further qualify the step selection strategy. Requiring s_k to solve (2.1), (2.2) exactly would do this, but it turns out that all we need to require, in addition to (2.3) and (2.4), is that a Newton step is taken only if B_k is positive semidefinite.

THEOREM 2.8. *Suppose that Algorithm* 2.1 *is run, that Assumptions* (A1)–(A3) *hold, and that the steps s_k are computed such that $\|s_k\| < 0.8\Delta_k$ only when B_k is positive semidefinite and $B_k s_k = -\nabla f(x_k)$. Then*

$$\lim_{k\to\infty} \frac{1}{k} |\{j \in [1,k] | B_j \text{ is positive semidefinite } \}| = 1. \tag{2.31}$$

Proof. Under these assumptions Lemma 2.6 holds and so

$$\frac{h_k}{\Delta_k} \to 0.$$

Consider any $\delta > 0$. Choose p such that $\frac{n}{p} < \delta$. By Lemma 2.4 we can find $K_1 > 0$ such that for any $k > K_1$ there exists a set $\mathcal{G}_k \subset \{k+1, \ldots, k+p\}$ such that $|\mathcal{G}_k| \geq p - n$ and such that (2.14) holds for all $j \in \mathcal{G}_k$. In addition, if K_1 is sufficiently large, we have that

$$\frac{h_j}{\Delta_j} < 0.8 \left[\beta_2 \left(\frac{2}{\beta_1}\right)^{\frac{3}{2}}\right]^{-1} \quad \text{for } j > K_1$$

by Lemma 2.6. Therefore, for $j \in \mathcal{G}_k$ with $k > K_1$ by (2.14) and (2.9)

$$\|s_j\| \leq \frac{2\beta_2}{\beta_1} e_j < \beta_2 \left(\frac{2}{\beta_1}\right)^{\frac{3}{2}} h_j < 0.8\Delta_j.$$

By hypothesis this only occurs if B_j is positive semidefinite. Therefore B_j is positive semidefinite when

$$j \in \bigcup_{i=0}^{\infty} \mathcal{G}_{K_1+ip}.$$

However, for any $k > K_1$ we can bound the cardinality of this set by

$$\left| \left[\bigcup_{i=0}^{\infty} \mathcal{G}_{K_1+ip} \right] \bigcap [1,k] \right| \geq \left\lfloor \frac{k - K_1}{p} \right\rfloor (p-n)$$

$$
\begin{aligned}
&> \frac{k-(K_1+p)}{p}(p-n) \\
&> \frac{p-n}{p}k-(K_1+p) \\
&= \left[\frac{p-n}{p}-\frac{K_1+p}{k}\right]k \\
&> \left[1-\delta-\frac{K_1+p}{k}\right]k.
\end{aligned}
$$

Thus for k sufficiently large,

$$\frac{1}{k}|\{j \in [1,k] | B_j \text{ is positive semidefinite }\}| > 1-2\delta.$$

Since δ is chosen arbitrarily, (2.31) follows. □

The restriction used in Theorem 2.8, that the full secant step can only be taken when the Hessian approximation is at least positive semidefinite, is imposed in virtually all implementations of trust region methods that we are aware of. Thus Theorem 2.8 shows the interesting fact that a practical SR1 algorithm that uses the framework of Algorithm 2.1 will generate positive definite Hessian approximations asymptotically at most iterations for a broad range of problems. The fact that the algorithm makes updates to the Hessian approximations at failed points is needed to establish this property.

3. Computational results. Algorithm 2.1 differs from a standard trust region algorithm in that the Hessian approximating matrix, B_k, is updated at all generated steps including the rejected steps. This was necessary for the convergence analysis, but it imposes the cost of an extra gradient evaluation at the rejected points upon the algorithm. However, it is possible that updating the Hessian approximation more often could make it more accurate and thereby reduce the total number of iterations to offset, or partially offset, this extra evaluation cost. In this section we present results of a numerical experiment conducted to investigate this possibility.

The trust region algorithm we used is from the UNCMIN unconstrained optimization software package (Schnabel, Koontz, and Weiss [1985]), modified to use the SR1 update and so that the exact trust region step is taken in all iterations. We compared two variants, one where updates are made only at successful points and one where updates are made at successful and failed points. We found that the option to update at failed points worked better if we skipped the update at failed points where the increase in f was too large. Specifically, the update in Step 6 of Algorithm 2.1 was not done if $f(x_k+s_k)-f(x_k) > 0.5[f(x_0)-f(x_k)]$. This change does not affect the convergence theory because it is easy to show from Lemma 2.2 that as x_k converges to x_*, $f(x_k+s_k)-f(x_k)$ converges to zero. For these experiments we used analytic gradients and the gradient stopping tolerance was 10^{-5}. The test problems were selected from Moré, Garbow, and Hillstrom [1981].

The results of this experiment are given in Table A. In this table limited updating refers to the trust region–SR1 algorithm when updates are made only along accepted steps, and updating at all points refers to the same algorithm when updates are made at all steps. For each test function the table contains the number of the function as given in the original source, the dimension of the problem (n), the number of *successful* iterations required to solve the problem (itrn.), the number of function evaluations required to solve the problem (f-eval), the number of gradient evaluations required to

TABLE A
Comparisons of limited updating and updating at all points.

Function	n	SR1–Limited updating				SR1–Updating at all points					sp
		itrn.	f-eval	g-eval	rgx	itrn.	f-eval	g-eval	updf	rgx	
MGH05	2	17	24	17	0.5E-05	16	24	20	4	0.8E-05	1
MGH07	3	31	41	31	0.6E-06	31	41	36	5	0.8E-06	1
MGH09	3	3	5	3	0.1E-07	3	5	3	0	0.1E-07	1
MGH12	3	35	45	35	0.1E-05	29	38	36	7	0.2E-05	1
MGH14	4	38	46	38	0.2E-05	48	62	57	9	0.1E-05	1
MGH16	4	21	29	21	0.2E-06	22	30	24	2	0.3E-08	1
MGH18	6	58	70	58	0.6E-06	50	61	57	7	0.5E-06	1
MGH20	9	73	92	73	0.7E-05	56	69	64	8	0.8E-05	1
MGH21	10	97	125	97	0.3E-05	53	69	63	10	0.3E-05	1
MGH22	8	29	33	29	0.6E-05	29	33	29	0	0.6E-05	1
MGH23	10	8	16	8	0.3E-06	9	17	13	4	0.1E-05	1
MGH24	10	27	36	27	0.5E-05	26	33	29	3	0.3E-05	1
MGH25	10	14	18	14	0.7E-05	14	18	14	0	0.7E-05	1
MGH26	10	25	31	25	0.2E-05	25	31	30	5	0.6E-06	1
MGH35	9	20	25	20	0.5E-05	20	25	21	1	0.5E-05	1
MGH05	2	42	53	42	0.1E-05	42	54	50	8	0.2E-06	10
MGH07	3	32	40	32	0.5E-06	34	48	43	9	0.2E-05	10
MGH09	3	21	28	21	0.8E-05	22	29	25	3	0.2E-07	10
MGH14	4	99	124	99	0.6E-06	86	107	102	16	0.9E-06	10
MGH16	4	36	47	36	0.5E-05	36	44	39	3	0.6E-06	10
MGH18	6	56	77	56	0.4E-05	51	65	62	11	0.1E-05	10
MGH20	9	73	92	73	0.7E-05	56	69	64	8	0.8E-05	10
MGH21	10	71	92	71	0.8E-07	71	91	87	16	0.6E-06	10
MGH22	8	46	52	46	0.9E-05	46	52	47	1	0.9E-05	10
MGH24	10	475	602	475	0.1E-04	344	434	430	86	0.1E-04	10
MGH25	10	25	30	25	0.4E-08	25	30	25	0	0.4E-08	10
MGH26	10	46	60	46	0.2E-05	46	61	53	7	0.2E-05	10
MGH07	3	24	34	24	0.6E-06	20	34	31	11	0.3E-07	100
MGH09	3	16	22	16	0.1E-06	16	22	18	2	0.9E-05	100
MGH14	4	103	130	103	0.7E-09	86	106	101	15	0.2E-06	100
MGH16	4	52	61	52	0.6E-07	49	58	53	4	0.9E-06	100
MGH18	6	94	108	94	0.1E-04	112	130	122	10	0.1E-04	100
MGH20	9	73	92	73	0.7E-05	56	69	64	8	0.8E-05	100
MGH21	10	411	531	411	0.5E-06	264	338	334	70	0.3E-05	100
MGH22	8	93	105	93	0.1E-05	59	68	63	4	0.5E-05	100
MGH26	10	39	55	39	0.5E-06	56	70	69	13	0.1E-06	100

TABLE B
Ratios of costs of SR1 with all-point updating to costs of SR1 with limited updating.

	Iterations	Function evaluations	Gradient evaluations
Arithmetic mean	0.83	0.83	0.98
Geometric mean	0.93	0.93	1.07

solve the problem (g-eval), and the relative gradient at the solution (rgx). The tenth column (updf) contains the number of times updates were made along failed steps, and the last column (sp) indicates whether the starting point used is x_0, $10x_0$, or $100x_0$, where x_0 is the standard starting point.

Table B summarizes the data in Table A and contains the ratios of the mean number of iterations, the mean number of function evaluations, and the mean number of gradient evaluations required to solve these problems by the algorithm when updates are made at all steps to those required by the algorithm when updates are made only at successful steps. Both arithmetic and geometric means are presented. These numbers indicate that on the set of test problems we used, updating at failed points improved the performance of the algorithm with respect to iterations and function evaluations and may have slightly increased the cost with respect to gradient evaluations depending on the statistic used. Combined with the theoretical advantages that are derived from updating at failed points, these computational results indicate to us that an implementation of a trust region–SR1 method for unconstrained optimization probably should include this feature.

Acknowledgments. The authors would like to thank the referees for their very careful reading and many helpful corrections.

REFERENCES

C. G. BROYDEN, J. E. DENNIS JR., AND J. J. MORÉ [1973], *On the local and superlinear convergence of quasi-Newton methods*, J. Inst. Math. Appl., 12, pp. 223–246.

A. R. CONN, N. I. M. GOULD, AND PH. TOINT [1988], *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50, pp. 399–430.

——— [1991], *Convergence of quasi-Newton matrices generated by the symmetric rank one update*, Math. Programming, 50, pp. 177–195.

——— [1992], *Numerical Experiments with the LANCELOT Package (Release A) for Large-scale Nonlinear Optimization*, Report 92/16, Department of Mathematics, University of Namur, Belgium.

H. FAYEZ KHALFAN [1989], *Topics in Unconstrained Optimization*, Ph.D. thesis, Department of Mathematics, University of Colorado at Boulder, Boulder, CO.

H. FAYEZ KHALFAN, R. H. BYRD, AND R. B. SCHNABEL [1993], *A theoretical and experimental study of the symmetric rank one update*, SIAM J. Optim., 3, pp. 1–24.

R.-P. GE AND M. J. D. POWELL [1983], *The convergence of variable metric matrices in unconstrained optimization*, Math. Programming, 27, pp. 123–143.

J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTROM [1981], *Testing unconstrained optimization software*, TOMS, 7, pp. 17–41.

J. J. MORÉ AND D. C. SORENSEN [1983], *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4, pp. 553–572.

J. M. ORTEGA AND W. C. RHEINBOLDT [1970], *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

M. J. D. POWELL [1970], *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. Rosen, O. Mangasarian, and K. Ritter, eds., Academic Press, New York, pp. 31–65.

——— [1975], *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, pp. 1–27.

R. B. SCHNABEL, J. E. KOONTZ, AND B. E. WEISS [1985], *A modular system of algorithms for unconstrained minimization*, TOMS, 11, pp. 419–440.

G. A. SHULTZ, R. B. SCHNABEL, AND R. H. BYRD [1985], *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22, pp. 47–67.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1040–1058, November 1996
© 1996 Society for Industrial and Applied Mathematics
009

A REFLECTIVE NEWTON METHOD FOR MINIMIZING A QUADRATIC FUNCTION SUBJECT TO BOUNDS ON SOME OF THE VARIABLES*

THOMAS F. COLEMAN† AND YUYING LI†

Abstract. We propose a new algorithm, a reflective Newton method, for the minimization of a quadratic function of many variables subject to upper and lower bounds on some of the variables. The method applies to a general (indefinite) quadratic function for which a local minimizer subject to bounds is required and is particularly suitable for the large-scale problem. Our new method exhibits strong convergence properties and global and second-order convergence and appears to have significant practical potential. Strictly feasible points are generated. We provide experimental results on moderately large and sparse problems based on both sparse Cholesky and preconditioned conjugate gradient linear solvers.

Key words. interior Newton method, interior-point method, quadratic programming

AMS subject classifications. 65K05, 90C20, 90C06

1. Introduction.

In this paper we propose a new algorithm for solving the box-constrained quadratic programming problem

$$\min_x \left\{ q(x) \stackrel{def}{=} c^T x + \frac{1}{2} x^T H x : \quad l \le x \le u \right\}. \tag{1.1}$$

The matrix H is symmetric and, in general, indefinite; $l \in \{\Re \cup \{-\infty\}\}^n$, $u \in \{\Re \cup \{\infty\}\}^n$, $l < u$. We denote the feasible region $\mathcal{F} = \{x : \ l \le x \le u\}$ and the strict interior, $\text{int}(\mathcal{F}) = \{x : \ l < x < u\}$. When H is indefinite we are interested in locating a local minimizer.

The purpose of this paper is to motivate and describe our new approach, the reflective Newton method, and to present the results of computational experiments designed to investigate practical viability. A companion report [9] proffers supporting theory and analysis.

Problem (1.1) arises as a subproblem when minimizing general nonlinear functions subject to bounds, and as a problem in its own right. The box-constrained quadratic programming problem represents an important class of optimization problems and has been the subject of considerable recent work (e.g., [1, 4, 11, 14, 13, 17, 19, 20, 22, 24, 26, 27, 30]). A special subclass deserves mention: the box-constrained least-squares problem

$$\min_x \{ \|Ax - b\|_2 : \quad l \le x \le u \}, \tag{1.2}$$

where A is a rectangular m-by-n matrix with $m > n$. Our proposed algorithm can of course be applied to this special case if we use $H = A^T A$ and $c = -A^T b$; moreover, it

* Received by the editors August 16, 1994; accepted for publication (in revised form) June 6, 1995. This research was partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25013.A000, by the Computational Mathematics Program of the National Science Foundation under grant DMS-8706133, and by the Advanced Computing Research Institute, a unit of the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

† Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14850 (coleman@cs.cornell.edu).

is possible to implement a conjugate gradient version of our method without explicitly forming $H = A^T A$.

We propose a new approach, a reflective Newton algorithm. The algorithm generates a sequence of strictly feasible iterates, $\{x_k\}$, which converges under standard assumptions to a local solution of (1.1), x_*, at a quadratic convergence rate. Coleman and Li [9] establish theoretical properties of the reflective Newton approach applied to the general nonlinear box-constrained problem—these results apply directly to the reflective Newton procedure proposed here for the quadratic minimization problem (1.1).

This new approach has potential advantages over existing methods. The main advantage appears to be the ability to achieve accurate solutions to large problems in relatively few iterations. The global and quadratic convergence properties established in [9] account, in some measure, for this good behaviour. Our numerical results, reported in §4, support the notion that for a given problem class, the number of required iterations increases only modestly with problem size. The numerical results also indicate a remarkable insensitivity to problem conditioning and the degree of near degeneracy. This is partially explained by the choice of scaling matrix inherent in our approach; we discuss this subsequently.

The sequence $\{x_k\}$ generated by the algorithm is strictly feasible; therefore, the algorithm can be regarded as an "interior-point" algorithm. However, this may be a misleading classification. The algorithm differs markedly from methods commonly referred to as "interior-point" algorithms. For example, the proposed algorithm does not use a barrier function to help ensure strict feasibility. The algorithm generates descent directions for $q(x)$ and then follows a piecewise linear path, reflecting off constraints as they are encountered. Most interior-point methods, on the other hand, generate descent directions (for some function) and then restrict the step, along this straight-line direction, to ensure strict feasibility. The proposed algorithm does not require the delicate choice or adjustment of a penalty or barrier parameter—no penalty or barrier function is used—nor does it require that a "central path" be followed. Indeed the proposed method is robust with respect to starting point.

The algorithm most similar to our current proposal is probably the recent method due to Coleman and Hulbert [5]. (There is also a strong connection to previous work by Coleman and Li [6, 7, 8] and Li [21] on various norm minimization problems.) Both are driven by the nonlinear system of equations representing first-order optimality conditions. Both methods require piecewise quadratic minimization. The methods differ in that our new algorithm is more general: positive definiteness of the symmetric matrix H is not required and it is not necessary to have finite upper and lower bounds on all the variables—the Coleman–Hulbert method requires both of these restrictive properties. Finally, the Coleman–Hulbert method is an exterior-point method, requiring strict decrease in a piecewise quadratic "dual" function, whereas the new method generates feasible iterates, requiring strict decrease in the original quadratic function q.

We conclude the introduction with a short review of optimality conditions for problem (1.1). Define $D = D(x)$ to be a diagonal matrix with the ith diagonal component equal to $|v_i(x)|^{\frac{1}{2}}$. Vector $v_i(x)$ is defined in Figure 1.1, where

$$g(x) \stackrel{def}{=} \nabla q(x) = Hx + c.$$

How to compute $v(x)$:

- If $g_i < 0$ and $u_i < \infty$ then $v_i = x_i - u_i$.
- If $g_i \geq 0$ and $l_i > -\infty$ then $v_i = x_i - l_i$.
- If $g_i < 0$ and $u_i = \infty$ then $v_i = -1$.
- If $g_i \geq 0$ and $l_i = -\infty$ then $v_i = 1$.

FIG. 1.1. *Definition of* $v(x)$.

The first-order optimality conditions can now be written. If a feasible point x_* is a local minimizer of (1.1) then

$$D_*^2 g_* = 0. \tag{1.3}$$

Let $Free_*$ denote the set of indices corresponding to "free" variables at point x_*:

$$Free_* = \{i : l_i < (x_*)_i < u_i\}.$$

Second-order necessary conditions can be written.[1] If a feasible point x_* is a local minimizer of (1.1) then $D_*^2 g_* = 0$ and $H^{Free_*} \geq 0$, where H^{Free_*} is the submatrix of H corresponding to the index set $Free_*$.

These conditions are necessary but not sufficient. To state practical sufficiency conditions we first need a definition of degeneracy.

DEFINITION 1.1. *A point* $x \in \Re^n$ *is* nondegenerate *if for each index* i

$$g_i = 0 \implies l_i < x_i < u_i.$$

With this definition we can state *second-order sufficiency conditions.* If a nondegenerate feasible point x_* satisfies $D_*^2 g_* = 0$ and $H^{Free_*} > 0$, then x_* is a local minimizer of (1.1).

2. Motivation. In this section we motivate and explain the basis of the reflective Newton method. Additional intuition can be gleaned from [9] where greater attention is paid to the theoretical underpinnings. We begin at the end.

2.1. The end game. As indicated by (1.3), a local solution to (1.1) is a zero of the nonlinear system

$$D^2(x)g(x) = 0, \tag{2.1}$$

where $g(x) \stackrel{def}{=} Hx+c$. The system (2.1) is nondifferentiable when either some $v_i = 0$ or $g_i = 0$. Let x_* be a nondegenerate feasible point satisfying the second-order sufficiency conditions. If a feasible point x is sufficiently close to x_* then a Newton step, $s^N(x)$, can be defined with respect to system (2.1):

$$s^N(x) \;=\; -(D^2 H + JD^g)^{-1} D^2 g, \tag{2.2}$$

where $D^g = D^g(x) \stackrel{def}{=} \mathrm{diag}(|g|)$. Each diagonal element of the diagonal Jacobian matrix J is defined as follows: If v_i equals either $x_i - u_i$ or $x_i - l_i$, then $J_{ii} = 1$; otherwise, $J_{ii} = 0$. In [9] it is established that the damped local Newton sequence,

$$x_{k+1} = x_k + \alpha_k s_k^N \tag{2.3}$$

[1] Notation: If a matrix A is a symmetric matrix then we write $A > 0$ to mean A is positive definite; $A \geq 0$ means A is positive semidefinite.

is a locally quadratically convergent sequence provided $\{x_k\}$ is strictly feasible and

$$|1 - \alpha_k| = O(\|x_k - x_*\|). \tag{2.4}$$

A crucial observation, proven in [9], is that the distance to the boundary from x_k, along s_k^N, satisfies (2.4). In particular, for each $j \notin Free_*$, $|1-(\beta_k^N)_j| = O(\|x_k-x_*\|)$, where $(\beta_k^N)_j = \frac{|(v_k(x))_j|}{|(s_k^N(x))_j|}$. The positive number $(\beta_k^N)_j$ is the distance from x_k to the approaching bound, one of u_j, l_j, in the direction s_k^N. Also, for $j \in Free_*$, $(\beta_k^N)_j \to \infty$ as $x_k \to x_*$. Therefore, it is possible to choose α_k to satisfy (2.4) and produce feasible iterates via (2.3). For example, let $(\beta_k^N)_{\min} = \min\{(\beta_k^N)_j\}$ and set $\alpha_k = (\beta_k^N)_{\min} - \epsilon_k$, where ϵ_k is a positive number satisfying $\epsilon_k \leq \chi_\alpha \|D_k g_k\|$ for any positive constant χ_α. It immediately follows that α_k satisfies (2.4) and that $\{x_k\}$, generated by (2.3), is strictly feasible. Local and quadratic convergence ensues [9].

We maintain strict feasibility $l < x_k < u$ at each iteration. Hence, a good way to compute s_k^N is to first form $\bar{M}_k = D_k H_k D_k + J_k D_k^g$, then solve

$$\bar{M}_k \bar{s}_k = -D_k g_k, \tag{2.5}$$

and finally set $s_k = D_k \bar{s}_k$. Notice that $\bar{M}_k$ is symmetric; moreover, as shown in [9], $\bar{M}_k$ is positive definite and bounded in a feasible neighbourhood of x_*. Furthermore, it can be argued that the conditioning of $\bar{M}_k$ in a feasible neighbourhood of x_* reflects the conditioning of the original problem (1.1) around x_*.

So we have all the ingredients necessary to construct a robust, interior, local, and quadratic Newton process for (1.1).

Algorithm 1

For $k = 1 : \infty$

1. Form $\bar{M}_k = D_k H_k D_k + J_k D_k^g$.
2. Solve $\bar{M}_k \bar{s}_k^N = -D_k g_k$. Set $s_k^N = D_k \bar{s}_k^N$.
3. Compute $\beta_{k_{min}}^N$; set $\alpha_k = \beta_{k_{min}}^N - \epsilon_k$, where $\epsilon_k \leq \chi_\alpha \|D_k g_k\|$.
4. Set $x_{k+1} = x_k + \alpha_k s_k^N$.

FIG. 2.1. *A local (strictly) feasible Newton method.*

Algorithm 1 (Figure 2.1) represents a practical interior Newton process. However, it is clear that this is a local procedure only; the challenge is to extend this approach to a global method.

2.2. The middle game. Our objective is to devise a global method that smoothly evolves into Algorithm 1, the fast local procedure discussed above. A common globalization technique is to generate a descent direction at the current point for a suitable "merit" function and then take a step along this direction. The most natural merit function for a strictly feasible algorithm is the objective function itself, $q(x) = c^T x + \frac{1}{2} x^T H x$. It is easy to see that the Newton step s_k^N, defined by (2.5) or (2.2), is a descent direction for q in a feasible neighbourhood of a nondegenerate second-order point x_*. The matrix $\bar{M}(x)$ is positive definite in a feasible neighbourhood of x_*. Moreover, in [9] we show that the stepsize $\alpha = 1$ yields sufficient decrease of $q(x)$ in a feasible neighbourhood of x_*. It follows that the calculation of the step length α described in Algorithm 1 also yields sufficient decrease of $q(x)$ in a feasible

neighbourhood of x_*. However, when $\|x - x_*\|$ is large, $\bar{M}(x)$ is not necessarily positive definite, and $s^N(x)$ is not necessarily a descent direction for q at x. Indeed, $\bar{M}(x)$ may be singular and therefore $s^N(x)$ is not necessarily well defined at points remote from x_*.

Our solution to this problem is founded on a highly successful technique used in unconstrained nonlinear minimization: minimize a quadratic approximation subject to an elliptical constraint. For example, a descent direction s_k can be determined by solving

$$\min_s \left\{ s^T g_k + \frac{1}{2} s^T B_k s : \ \|S_k^{-1} s\|_2 \leq \Delta_k \right\} \tag{2.6}$$

or, more generally,

$$\min_s \left\{ s^T g_k + \frac{1}{2} s^T B_k s : \ \|S_k^{-1} s\|_2 \leq \Delta_k, \ s \in \mathcal{S}_k \right\}, \tag{2.7}$$

where $\mathcal{S}_k$ is a subspace of $\mathcal{R}^n$, S_k is a positive diagonal scaling matrix, B_k is symmetric, and $\Delta_k > 0$. Appropriate choices of B_k, S_k, and $\mathcal{S}_k$ yield a descent direction for $q(x)$ at x_k. However, a most interesting question is how to choose these quantities to enable a smooth tie-in to Algorithm 1, the local procedure above. Appropriate choices may surprise. Choose $S = D$ and $B = M = [H + JD^g D^{-2}]$. (Recall that D is a diagonal matrix with the ith diagonal element defined to be $|v_i(x)^{\frac{1}{2}}|$, $i = 1 : n$.) Notice that with this selection (2.6) becomes

$$\min_{\bar{s}} \left\{ \bar{s}^T \bar{g}_k + \frac{1}{2} \bar{s}^T \bar{M}_k \bar{s} : \ \|\bar{s}\|_2 \leq \Delta_k \right\}, \tag{2.8}$$

with $s_k = D_k \bar{s}_k$. Therefore, provided the ball constraint $\|\bar{s}\|_2 \leq \Delta_k$ is inactive near the solution, problem (2.8) yields $s_k = s_k^N$ in a feasible neighbourhood of the solution. Moreover the solution to (2.7) is also $s_k = s_k^N$ near the solution provided $\mathcal{S}_k$ contains s_k^N and the elliptical constraint $\|D_k^{-1} s\|_2 \leq \Delta_k$ is inactive.

Global convergence properties of optimization algorithms usually require "sufficient decrease" guarantees [12]. While (2.6) and (2.7) can yield a descent direction s_k at any feasible point x_k, it may not be possible to sufficiently decrease $q(x)$ along s_k. The problem is that the feasibility requirement may restrict α_k to be very small when $x_{k+1} = x_k + \alpha_k s_k$. Sufficient decrease in $q(x)$ may not be possible. Note that, as indicated above, this inhibiting behaviour cannot occur in a feasible neighbourhood of x_*, but it certainly can occur when x_k is far from x_* but close to the boundary.

Our solution to this problem has two parts. The first part involves a search along a piecewise linear path, a reflective path; the second part restricts the descent directions to have certain desirable properties. We discuss the reflective path idea first.

Given a feasible point x and an initial descent direction s we define the reflective path $p(\alpha)$ as follows. First, determine the array BR of breakpoint distances:

$$BR(i) = \max\{(l_i - x_i) \ / \ s_i, \ \ (u_i - x_i) \ / \ s_i)\}. \tag{2.9}$$

Component $BR(i)$ records the distance from x to the breakpoint corresponding to variable x_i in the positive direction s. The array BR can now be used in the definition of the piecewise linear (reflective) path $p(\alpha)$ given by Algorithm 2 in Figure 2.2 and (2.10): for $\beta^{i-1} \leq \alpha < \beta^i$,

$$p(\alpha) = b^{i-1} + (\alpha - \beta^{i-1}) p^i. \tag{2.10}$$

Algorithm 2 [Let $\beta^0 = 0$, $p^1 = s$, set $b^0 = x_k$.]

For $i = 1, \ldots$

1. Let β^i be the distance to the nearest breakpoint along p^i:
$$\beta^i = \min(BR).$$
2. Define ith breakpoint: $b^i = b^{i-1} + (\beta^i - \beta^{i-1})p^i$.
3. Reflect to get new dir'n and update BR:
   (a) $p^{i+1} = p^i$
   (b) For each j such that $(b^i)_j = u_j$ (or $(b^i)_j = l_j$)
   - $BR(j) = BR(j) + |\frac{(u_j - l_j)}{s_j}|$.
   - $(p^{i+1})_j = -(p^i)_j$.

FIG. 2.2. *Determine the linear reflective path p.*

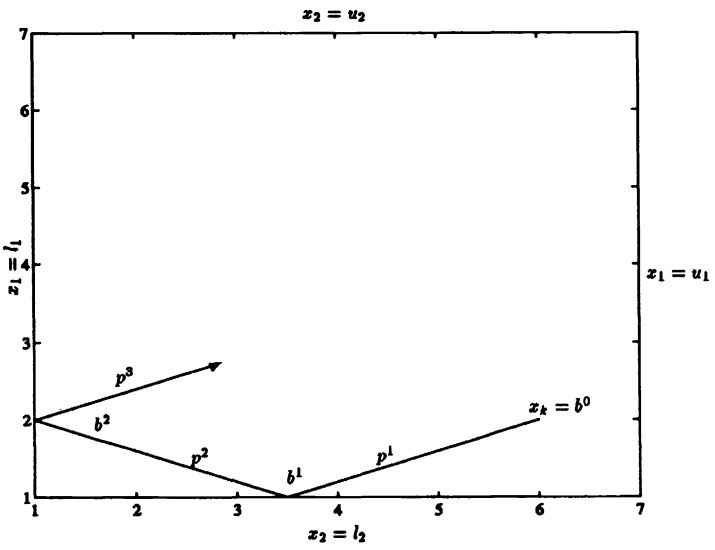


FIG. 2.3. *A reflective path.*

An example of a reflective path is given in Figure 2.3.

If we let $x_{k+1} = x_k + p_k(\alpha_k)$ then our sufficient decrease conditions along s_k are as follows: given $0 < \sigma_l < \sigma_u < 1$ and a descent direction s_k, α_k satisfies our approximate line search conditions if

$$q(x_{k+1}) < q(x_k) + \sigma_l \left(\alpha_k g_k^T s_k + \frac{1}{2}\alpha_k^2 \min(s_k^T H s_k, 0) \right) \tag{2.11}$$

and

$$q(x_{k+1}) > q(x_k) + \sigma_u \left(\alpha_k g_k^T s_k + \frac{1}{2}\alpha_k^2 \min(s_k^T H s_k, 0) \right), \tag{2.12}$$

where $g_k = \nabla q(x_k)$. These conditions are analogous to those suggested by Goldfarb [18] for use in the unconstrained nonlinear setting. As established in Theorem 2 of

[9], if boundedness of $q(x)$ in $\mathcal{F}$ is assumed, then an interval (α_l^k, α_u^k), $\alpha_l^k < \alpha_u^k$, can always be found such that if $\alpha \in (\alpha_l^k, \alpha_u^k)$ then $q(x_k + p_k(\alpha))$ satisfies (2.11) and (2.12). Condition (2.11) implies an upper bound on α; condition (2.12) gives a lower bound.

A basic reflective path algorithm for problem (1.1) can now be stated, as Algorithm 3 (Figure 2.4). To allow for flexibility, especially with regard to Newton steps, we do not always require that both (2.11) and (2.12) be satisfied. Instead, we demand that either both of these conditions are satisfied or (2.11) is satisfied and α_k is bounded away from zero, e.g., $\alpha_k > \rho > 0$. The latter conditions are used to allow for the liberal use of Newton steps and do not weaken the global convergence results.

Note that Algorithm 3 generates *strictly* feasible points; i.e., since $x_1 \in \text{int}(\mathcal{F})$, it follows that $x_k \in \text{int}(\mathcal{F})$.

Algorithm 3 [ρ is a positive scalar.]

Choose $x_1 \in \text{int}(\mathcal{F})$.

For $k = 1, 2, \ldots$

1. Determine an initial descent dir'n s_k for q at x_k . Note that the piecewise linear path p_k is defined by x_k, s_k.
2. Perform an approximate piecewise line minimization of $q(x_k + p_k(\alpha))$, with respect to α, to determine α_k such that
   (a) α_k does not correspond to a breakpoint,
   (b) condition (2.11) is satisfied,
   (c) either
      i. α_k satisfies condition (2.12), *or*
      ii. $\alpha_k > \rho > 0$.
3. $x_{k+1} = x_k + p_k(\alpha_k)$.

FIG. 2.4. *A reflective path algorithm satisfying line search conditions.*

Satisfaction of the line search conditions in Algorithm 3 is not sufficient to ensure a first-order convergence result. The convergence analysis given in [9] requires two additional properties of the search direction sequence $\{s_k\}$: "constraint compatibility" and "consistency."

DEFINITION 2.1. *A sequence of vectors* $\{s_k\}$ *is* constraint compatible *if the sequence* $\{D_k^{-2} s_k\}$ *is bounded.*

DEFINITION 2.2. *A sequence of vectors* $\{s_k\}$ *satisfies the* consistency condition *if* $\{s_k^T g_k\} \to 0$ *implies* $\{D_k g_k\} \to 0$.

Consistency is a standard notion which insists that first-order descent, represented by the term $g_k^T s_k$, be consistent with first-order convergence. This is a generalization of the condition that the angle of the gradient and the descent direction are bounded away from 90° in unconstrained minimization [12, p. 123].

Next we illustrate that the constraint compatibility of $\{s_k\}$ enables a sufficiently long step along s_k. Consider the following simple result.

LEMMA 2.3. *If* $\{s_k\}$ *is a constraint-compatible sequence then* $\{BR_k(j) : BR_k(j) = \frac{|(v_k)_j|}{|(s_k)_j|}\}$ *is bounded away from zero.*

Constraint compatibility avoids running directly into a bound by ensuring that the stepsizes to breakpoints, corresponding to "correct sign conditions," remain strictly bounded away from zero. Specifically, if $\{s_k\}$ is constraint compatible then the positive distance to constraint j along s_k, $BR_k(j) = \max\{\frac{l_j - (x_k)_j}{(s_k)_j}, \frac{u_j - (x_k)_j}{(s_k)_j}\}$, is strictly

bounded away from zero for any j with the correct sign condition. The correct sign condition implies a consistency between v_j and $\max\{\frac{l_j-(x_k)_j}{(s_k)_j}, \frac{u_j-(x_k)_j}{(s_k)_j}\}$. The correct sign condition holds when $(s_k)_j(g_k)_j < 0$, and so $BR_k(j) = \frac{|(v_k)_j|}{|(s_k)_j|}$.

When the "sign condition" is violated, i.e., when $(s_k)_j(g_k)_j > 0$, then a bound may be hit after only a short step along s_k. This is when the reflective line search idea steps forward: the reflective line search guarantees that the new direction passing this breakpoint will maintain descent if the bound is encountered soon enough (since $(s_k)_j(g_k)_j > 0$).

Several examples of descent directions satisfying both constraint compatibility and consistency are given in [9]. It is particularly noteworthy that the solution of the trust region problem (2.6) yields a direction s_k satisfying both conditions.

3. A practical algorithm. In this section we discuss two specifics of our approach that turn Algorithm 3 into a viable approach for large-scale problems. The first aspect is the line search; the second aspect is a particular way to determine the directions s_k via (2.6) by choosing a low-dimensional subspace $\mathcal{S}_k$ appropriately.

3.1. The line search. The most natural way to implement the piecewise line search is in a left-to-right fashion. With some cleverness an exact line search can be accomplished, moving left-to-right, requiring $O(n)$ work per breakpoint crossed. Unfortunately, even this linear order of work is too expensive for large-scale problems with many activities at the solution—the number of breakpoints crossed is proportional to the number of activities at the solution.

A viable solution to this problem is to use an approximate bisection algorithm, first attempting a unit step, $\alpha = 1$, to see if condition (2.11) is satisfied. (It is not necessary to test condition (2.12) since clearly a unit step is bounded away from zero!)

An important observation is that the point $x + p(\alpha)$ can easily be evaluated, no need to compute BR or apply Algorithm 2. To see this assume, for simplicity, that for each i all finite lower bounds equal zero and all finite upper bounds equal unity. Suppose we wish to evaluate $x + p(\alpha)$. Define $y = x + \alpha s$ and define R to be the reflective transformation, $R(y) = x + p(\alpha)$. The computation of R is described in Figure 3.1.

To evaluate $z_i = R(y)_i$

Case 1: $l_i = 0$, $u_i = 1$. Set $w_i = |y_i| \bmod 2$, set $z_i = \min(w_i, 2 - w_i)$.
Case 2: $l_i = 0$, $u_i = \infty$. Set $z_i = |y_i|$.
Case 3: $l_i = -\infty$, $u_i = 1$. If $y_i \leq 1$, set $z_i = y_i$; else, set $z_i = 2 - y_i$.
Case 4: $l_i = -\infty$, $u_i = \infty$. Set $z_i = y_i$.

FIG. 3.1. *The reflective transformation R.*

A bisection (piecewise) line search algorithm which yields a point x_{k+1} satisfying (2.11) and (2.12) can be implemented as follows. Assume x is the current feasible point and s is the current descent direction at x. As usual, $p(\alpha)$ denotes the piecewise linear path defined in (2.10). The simple bisection algorithm is given in Figure 3.2.

3.2. Subspace selection. We propose to implement step 1 of Algorithm 3, i.e., determine the descent direction s_k, by solving a problem of the form (2.7):

$$\min_s \left\{ s^T g_k + \frac{1}{2} s^T M_k s : \ \|D_k^{-1} s\|_2 \leq \Delta_k, \ s \in \mathcal{S}_k \right\}. \tag{3.1}$$

Algorithm 4

- If $R(x+s)$ satisfies (2.11), set $\hat{\alpha}=1$.
- Otherwise
  - Set $\alpha_L=0$, $\alpha_R=1$, $y=x+\frac{1}{2}s$.
  - Repeat until $R(y)$ satisfies (2.11) and (2.12)
    - If $R(y)$ violates (2.11), set $\alpha_R=y$;
      else if $R(y)$ violates (2.12), set $\alpha_L=y$, end
    - Set $\hat{\alpha}=(\alpha_L+\alpha_R)/2$, $y=x+\hat{\alpha}s$.
- If $\hat{\alpha}$ corresponds to a breakpoint, set $\alpha=\hat{\alpha}-\epsilon$, where $\epsilon\leq\chi_\alpha\|Dg\|$; otherwise, set $\alpha=\hat{\alpha}$.

FIG. 3.2. *The piecewise linear bisection algorithm.*

A subspace $\mathcal{S}_k$ must be chosen. If we choose $\mathcal{S}_k=\Re^n$ then strong convergence results follow [9]; however, this choice is impractical for large problems.

To meet global convergence requirements, the inclusion of the scaled gradient $D_k^2 g$ in $\mathcal{S}_k$ (or a direction with a nonzero projection onto the scaled gradient) is useful. Moreover, asymptotic second-order convergence requires that a good approximation to the Newton step s_k^N be included in $\mathcal{S}_k$ when $\bar{M}_k$ is determined to be positive definite. Finally, when $\bar{M}_k$ is determined to be non–positive definite, it is desirable to include a direction of negative curvature in $\mathcal{S}_k$ if available. That is, include $D_k w_k$ in $\mathcal{S}_k$ when $w_k^T\bar{M}_k w_k<0$. These subspace ideas are similar to those explored in [3] for unconstrained minimization problems. Coleman and Li [9] show that the consistency and constraint-compatibility conditions can be satisfied using the two-dimensional subspace framework.

Thus our practical algorithm has the structure given in Figure 3.3.

Algorithm 5 [ρ is a positive scalar.]

Choose $x_1\in\text{int}(\mathcal{F})$.

For $k=1,2,\ldots$

1. Determine a two-dimensional subspace $\mathcal{S}_k$: if a nonzero vector w_k is found such that $w_k^T\bar{M}_k w_k\leq 0$, set $\mathcal{S}_k=<D_k^2 sgn(g_k), D_k w_k>$; otherwise, set $\mathcal{S}_k=<D_k^2 g_k, D_k\tilde{s}_k^N>$, where $\tilde{s}_k^N$ approximately solves (2.5).
2. Determine an initial descent dir'n s_k for q at x_k: solve (2.6) to determine s_k. Note that the piecewise linear path p_k is defined by x_k, s_k.
3. Apply Algorithm 4, the approximate bisection minimization of $q(x_k+p_k(\alpha))$, with respect to α, to determine α_k such that
   (a) α_k does not correspond to a breakpoint,
   (b) condition (2.11) is satisfied,
   (c) either
      i. α_k satisfies condition (2.12), *or*
      ii. $\alpha_k>\rho>0$.
4. $x_{k+1}=x_k+p_k(\alpha_k)$.

FIG. 3.3. *A practical reflective path algorithm satisfying line search conditions.*

Algorithm 5 will locate a second-order point, under reasonable assumptions, when a constraint-compatible direction sequence $\{w_k\}$ of sufficient negative curvature, rel-

ative to $\lambda_{min}(\bar{M}_k)$, is guaranteed [9]. The computational results we present in this paper correspond to implementations that do not guarantee that sufficient negative curvature is found. However, our implementations are globally first-order convergent and if there is convergence to a second-order point then, under second-order sufficiency assumptions, a second-order convergence rate can be achieved [9].

We consider two variations of Algorithm 5 in our experimental testing. The two variations represent different ways to compute the subspace $\mathcal{S}_k$ in step 1 of Algorithm 5; both variations have applicability to large-scale problems.

Variation Cholesky. A sparse Cholesky factorization of a permutation of $\bar{M}_k$ is attempted. If this factorization completes, then $\bar{M}_k$ is positive definite and $\tilde{s}_k^N = -\bar{M}_k^{-1}\bar{g}_k$ is computed. Otherwise, a nonzero vector w_k is found such that $w_k^T \bar{M}_k w_k \leq 0$. In our experiments we use the symmetric minimum degree ordering to define a permutation of $\bar{M}_k$ to reduce fill in the Cholesky factor, e.g., [15]. Note that the ordering step, and determination of potential fill, needs to be done only once for a given instance of problem (1.1).

Variation conjugate gradients (PCG). We apply a preconditioned conjugate gradient (PCG) process to the system $\bar{M}_k \tilde{s}_k^N = \bar{g}_k$ until either a vector $\tilde{s}_k^N$ is found such that $\|\bar{M}_k \tilde{s}_k^N - \bar{g}_k\| \leq \epsilon_k$ for a given tolerance $\epsilon_k > 0$ or a nonzero vector w_k is generated such that $w_k^T \bar{M}_k w_k \leq 0$ [29].

Both variations exhibit a first-order global convergence result; moreover, assuming convergence to a point satisfying second-order sufficiency conditions, variation Cholesky will generate a quadratically convergent sequence, whereas variation PCG will yield a superlinearly convergent sequence provided ϵ_k goes to zero sufficiently fast [2].

4. Numerical experiments. We have implemented our algorithm in MATLAB 4.0, a version of MATLAB which allows for *sparse* matrix data structures [16]. In this section we present some preliminary numerical results.

For each problem we record the number of required iterations for each of our two variations of Algorithm 5, Cholesky and PCG. In the PCG approach we use a simple diagonal preconditioner where the diagonal elements correspond to the 2-norms of the columns of the corresponding matrix $\bar{M}_k$. Our purpose here is not to compare the overall relative efficiencies of these two variations per se but rather to explore the use of each approach within the context of the reflective Newton method, Algorithm 5. Is the number of major iterations required by the reflective Newton method sensitive to the use of Cholesky versus PCG? How does the number of major iterations grow with problem size and is this growth dependent on the choice of PCG versus Cholesky? How does the reflective Newton approach depend on problem characteristics, and is this dependence related to the choice of PCG versus Cholesky? Of course ultimately the relative efficiencies of the Cholesky version versus the PCG depend also on the work required to "solve" each linear system. This, in turn, depends on the usual factors: sparsity and distribution of eigenvalues. Except for a few remarks at the end of this section, we do not explore the latter question in this paper.

All experiments were performed on Sun Sparc workstations in the (sparse) MATLAB environment [23].

Starting and stopping. In all the experiments reported in this paper the starting value of x, i.e., x_1, is as follows. For component j where both upper and lower bounds are finite, choose the midpoint, $(x_1)_j = \frac{l_j + u_j}{2}$. If both upper and lower bound corresponding to component j are infinite in size, choose $(x_1)_j = 0$. If l_j is finite and $u_j = \infty$, choose $(x_1)_j = l_j + 1$; if $l_j = -\infty$ and u_j is finite, choose $(x_1)_j = u_j - 1$.

(Note: The reflective Newton approach is not particularly sensitive to starting value. For example, we repeated many of the experiments reported here using a random (strictly) feasible starting point; very little difference in behaviour was detected.)

Choosing a robust stopping rule in optimization is never easy. Our primary stopping rule is based on the relative difference in function value. This is reasonable partly because strict feasibility is always maintained and partly because usually the real objective in practical optimization is to achieve a point of relatively low function value. Specifically, our primary stopping rule is

$$q(x_k) - q(x_{k+1}) \leq tol * (1 + |q(x_k)|). \tag{4.1}$$

We choose $tol = 100 * \mu$ where μ is unit roundoff (machine epsilon); i.e., in MATLAB on a Sun Sparc workstation, $\mu = 2.2204 * 10^{-16}$. We do have secondary stopping criteria as well, designed to determine when progress is deemed too slow. This secondary rule tends to kick in when solving degenerate or ill-conditioned problems and a very flat region around the solution has been entered.

Of course the stopping criterion mentioned above does not guarantee a minimizer. In our numerical experiments reported in this paper, independent verification of optimality was performed whenever feasible. The optimal point was known, a priori, for the positive definite problems. In this case optimality was verified by comparing the computed function value at the computed solution to the known value (or, in some cases, the computed function value at the known minimizer). Optimality is much more difficult to verify for indefinite problems. For these problems, first-order optimality conditions were verified, and in some cases, when computationally feasible, second-order optimality was confirmed.

Parameter settings. There are a few preset parameters in the algorithm. Here are the settings we used in our experiments:

- σ_l: used in the line search, see (2.11): we use $\sigma_l = .1$.
- σ_u: used in the line search, see (2.12): we use $\sigma_u = .9$.
- ρ: a lower bound on the stepsize, see Algorithm 5: we use $\rho = .1$.
- χ_α: if the line search produces a unit step which turns out to be a breakpoint, this point is perturbed by an amount bounded by $\chi_\alpha \|D_k g_k\|$; see [9]: we use $\chi_\alpha = 1$.
- ϵ_k: used with the PCG variation; we set $\epsilon_k = .1$ for all k.

4.1. Positive definite problems. We have generated a number of quadratic test problems with certain properties. In the first set of results we concentrate on the case where H is symmetric positive definite. In the results reported below we use sparse matrices H with sparsity patterns representing a three-dimensional grid using a seven-point difference scheme. The Moré–Toraldo [24] QP-generator was adapted to generate problems with a given sparsity pattern (see also [5]). We will not review the generator characteristics here: our generator is a straightforward adaptation of the Moré–Toraldo scheme to the sparse setting using several sparse MATLAB functions (e.g., "sprandsym," "sprand").

In Tables 4.1–4.3, the dimension of the test problems is $n = 1000$ in each case. The parameter "pctbnd" indicates the percentage of variables tight at the solution, approximately evenly divided between upper and lower bounds. Parameter "deg" reflects the degree to which the solution is (nearly) degenerate; the larger the value of deg, the greater the amount of (near) degeneracy. Specifically, at the solution the gradient g satisfies $|g_i| \leq 10^{-\text{deg}}$ for some of the indices i corresponding to components of x which are tight at the solution, i.e., $x_i = u_i$ or $x_i = l_i$. Additional discussion

of this concept is contained in [5]. Parameter "cond" reflects the conditioning of the matrix H; the condition number of H is approximately 10^{cond}.

TABLE 4.1
Positive definite problems, pctbnd $= .1$, $n = 1000$.

		Cholesky		PCG	
deg	cond	max	avg	max	avg
3	3	15	14	16	14.7
6	3	16	15.6	17	16.3
9	3	15	15	16	16
3	6	14	12.7	15	14
6	6	16	15.3	17	16
9	6	15	15	17	16.7
3	9	13	12.7	14	13.7
6	9	16	15.3	17	16
9	9	16	15.7	17	16.3

TABLE 4.2
Positive definite problems, pctbnd $= .5$, $n = 1000$.

		Cholesky		PCG	
deg	cond	max	avg	max	avg
3	3	15	15	17	16
6	3	17	17	19	18.3
9	3	17	16.7	18	18
3	6	16	15.3	17	16
6	6	18	17.3	19	17.3
9	6	17	17	19	18.7
3	9	15	14.3	17	16
6	9	17	17	18	18
9	9	17	16.3	18	17.7

TABLE 4.3
Positive definite problems, pctbnd $= .9$, $n = 1000$.

		Cholesky		PCG	
deg	cond	max	avg	max	avg
3	3	17	16.7	18	17.7
6	3	18	17.3	19	18
9	3	17	16.3	19	18.7
3	6	16	15.7	18	16.7
6	6	18	17.3	19	18.3
9	6	18	17.3	18	16.7
3	9	16	15.3	16	15.7
6	9	17	17	18	17
9	9	17	16.7	18	17.7

The upper and lower bound vectors u and l were generated as follows. Approximately 75% of the components of l were chosen to be finite and assigned the value of zero; the index assignment was made in a random fashion. Similarly, approximately 75% of the components of u were chosen to be finite and assigned the value of unity. Again, the index assignment was made in a random fashion, independent of the assignment of l.

Each row of Tables 4.1–4.3 reflects the results of three independent runs with the same parameter settings. The columns labeled "max" indicate the maximum number

TABLE 4.4
Positive definite problems: Timing breakdown.

	Cholesky			PCG		
n	it	totM	totls	it	totM	totls
512	18	11.8	1.7	15	4.5	1.8
1000	15	38.5	3.9	16	8.4	3.8
1728	17	133.5	6.1	16	16.4	6.7
2744	16	533	11.4	17	33.3	12.1
8000	15	5369	29.4	17	95.2	33.6

of iterations required over the set of three independent runs to achieve the stopping criteria. The column labelled "avg" records the average number of iterations required to reach the stopping criteria over the three problems.

Full accuracy, in terms of function value, was achieved for every problem run. That is, the computed value of the objective function $q(x)$ at the computed solution matched the computed value of $q(x)$ at the true (known) solution to 15 decimal places.

Observations on Tables 4.1–4.3. First, we observe the remarkable consistency of both versions of the reflective Newton method in these problems. In terms of iterations required to achieve the stopping criteria and accuracy attained in the function value, there is apparently very little sensitivity to degeneracy, conditioning, or number of variables tight at the solution. Of course we do not claim that accuracy in x is independent of condition/degeneracy—it surely is not. However, it is usually acceptable in optimization to locate a point with nearly optimal function value, and we have been quite successful in that (on this test collection).

Second, the absolute number of iterations required to obtain a very accurate solution (in terms of the function value q) is modest in every case for both reflective Newton variations, i.e., less than 20. In general the PCG version required 1–2 additional iterations compared with the Cholesky version.

It is important to know where an algorithm spends its time. To this end we generated larger problems, with the same structure, and we have broken down the timing information. In Table 4.4 we consider a representative positive definite problem with "average characteristics," i.e., deg = 6, cond = 6, pctbnd = .5, and we vary the problem dimension n. (The sparsity structure remains the same.) The second column, labelled "it," records the number of iterations required to achieve the stopping criteria; "totM" records the total number of mega-flops used in the matrix operations corresponding to each linear system $\bar{M}_k s_k = \bar{g}_k$; "totls" records the number of mega-flops used in the approximate line search algorithm. More than 95% of the total flop count on these problems is represented by the sum of the totM and totls columns. The remaining work in the algorithm, such as the two-dimensional trust region solution, is negligible in comparison.

Observations on Table 4.4. First, there is no significant growth in number of iterations as the problem dimension n increases for either version. In the Cholesky variation, the sparse matrix factorization work totM increases relative to the line search cost totls as n increases; however, this trend is not evident in the PCG version. For large n the cost of the line search PCG is comparable to the linear solve whereas in Cholesky the line search cost is dominated by the matrix factorization cost. Therefore, in the Cholesky variation, speedup of the sparse Cholesky factorization aspect of the algorithm (e.g., use of parallelism, exploitation of specific particular structure) will have significant impact on the overall computing time. Conversely, improving

the approximate line search (in terms of cost) in the PCG version is as crucial as decreasing the cost of the matrix operations in the linear solve.

In addition to these randomly generated, but structured, positive definite problems, we have experimented with three specific test cases. Two of these problems are from the literature (e.g., [11, 25]) and the third example is new. In Tables 4.5 and 4.6 we report on the "obstacle" problem. In the first case there are lower bounds only; in the second case there are lower and upper bounds. In defining the specific example used we have chosen the same parameter settings and specific functions used in [25]. Table 4.7 reports on the elastic-plastic torsion problem. Again we used the same parameters as reported in [24] to define the problem; the parameter "c," as defined in [24], was assigned $c = 5$ in our experiments. The columns labelled "norm" record the logarithm (base 10) of the reciprocal of the final 2-norms of the vector D^2g, a first-order measure or optimality; see (1.3). For example, if $\|D^2g\| = 10^{-7}$ then the corresponding norm table entry equals 7.

TABLE 4.5
Obstacle problem: Lower bounds only.

		Cholesky		PCG	
m	n	its	norm	its	norm
30	900	14	15	17	8
40	1600	14	15	14	7
50	2500	15	15	21	8
60	3600	16	14	18	7
100	10,000	15	16	17	6

TABLE 4.6
Obstacle problem: Lower and upper bounds.

		Cholesky		PCG	
m	n	its	norm	its	norm
30	900	12	10	12	7
40	1600	12	16	12	7
50	2500	13	16	13	7
60	3600	13	16	15	9
100	10,000	14	9	14	8

TABLE 4.7
Elastic-plastic torsion problem.

		Cholesky		PCG	
m	n	its	norm	its	norm
30	900	10	15	11	6
40	1600	11	13	11	6
50	2500	11	13	16	9
60	3600	11	14	10	5
100	10,000	10	14	12	7

In Table 4.8 we report on a linear spline approximation problem. This type of problem arises, for example, in a particle method approach to turbulent combustion simulation [28]. The problem results in a large sparse least-squares problem subject to nonnegativity constraints on the variables. To set up a sample problem we assume an m-by-m-by-m three-dimensional grid. Within each cell is a set of particles randomly located. (We use approximately 10 particles per cell in our experiments.) Each

TABLE 4.8
Linear spline approximation.

		Cholesky		PCG	
m	n	its	norm	its	norm
30	900	15	13	14	6
40	1600	15	15	15	6
50	2500	15	15	17	7
60	3600	15	15	17	7
100	10,000	16	15	16	6

particle p has a known function value $\phi(p)$. Associate with each grid intersection point a linear basis function and determine the best set of coefficients, x, for the basis functions in the least-squares sense subject to nonnegativity constraints on x. The function ϕ we used in our experiments is defined as follows. Given a point in 3-space, $p = (p_1, p_2, p_3)$, define

$$\phi(p) = .3\sin(9.2p_1)\sin(9.3p_2)\sin(9.4p_3).$$

Observations on Tables 4.5–4.8. The most noteworthy observation is the apparent insensitivity of the reflective Newton method to problem size for each of these problems. The number of iterations does not grow significantly for a given problem class as the dimension of the problem increases. For example, for the linear spline problem, 15 iterations are required by the Cholesky variation when $n = 900$; 16 iterations are required when $n = 10,000$. Similarly, the PCG variation requires 14 iterations when $n = 900$ and 16 iterations when $n = 10,000$. In general, PCG required a few more iterations to satisfy the stopping criteria than did Cholesky. Moreover, the accuracy achieved by PCG, in terms of the final norm of D^2g, is considerably worse than Cholesky. The total amount of work is another matter; see Table 4.4 and subsequent comments.

4.2. Indefinite problems. We have adapted the Moré–Toraldo QP generation scheme, in combination with sparse matrix functions in MATLAB 4.0, to generate large sparse indefinite matrices, each with a given sparsity pattern and preassigned set of approximate eigenvalues. In the indefinite case we chose finite upper and lower bound vectors, $l = 0$, $u = 1$. (We use all finite bounds in the generation of indefinite problems to avoid the generation of unbounded problems.)

In each of the problems in Tables 4.9–4.11 roughly 10% of the eigenvalues of H are negative. (In each case the optimality conditions were verified to hold at the final point.)

Observations on Tables 4.9–4.11. Iteration counts indicate that the reflective Newton method is not quite as consistent or efficient on indefinite problems compared with the performance on positive definite problems. Still, the overall efficiency seems very good—the average number of iterations required for any problem category is always less than 23 for Cholesky and less than 26 for PCG.

The difference between PCG and Cholesky, in terms of number of iterations to satisfy the optimality conditions, is greater on the set of indefinite problems than on the positive definite set. It is not clear why this is the case.

In Table 4.12 we indicate where the algorithm spends its time on indefinite problems by considering a representative example and increasing the dimension.

Remark on Table 4.12. We see little growth in required iterations for either variation as n increases. Clearly the totM column dominates the totls column as

TABLE 4.9
Indefinite problems, pctbnd = .1, $n = 1000$.

		Cholesky		PCG	
deg	cond	max	avg	max	avg
3	3	18	16.7	21	20
6	3	19	17	22	21
9	3	23	19.3	22	21.3
3	6	14	13.7	19	18.3
6	6	32	22.7	25	22.7
9	6	26	21.7	24	22.7
3	9	15	14	24	21
6	9	16	15.7	24	22
9	9	16	15.7	25	21.7

TABLE 4.10
Indefinite problems, pctbnd = .5, $n = 1000$.

		Cholesky		PCG	
deg	cond	max	avg	max	avg
3	3	17	15.7	24	20
6	3	19	18	24	22.3
9	3	18	16.7	25	25
3	6	15	13.3	21	19
6	6	18	17.3	28	23.3
9	6	19	17.7	25	23.6
3	9	14	11.3	28	23
6	9	16	15.7	19	18.3
9	9	25	18.3	23	21.7

n increases for the Cholesky version; in the PCG variation the totM and totls are comparable. Recall that totM represents the matrix flop count for the linear solve while totls represents the number of total mega-flops required by the line search procedure. Again, for large n the cost of the line search PCG is comparable to the linear solve, whereas in Cholesky the line search cost is dominated by the matrix factorization cost. Therefore, to obtain further improvements in efficiency of the Cholesky variation it is best to focus on the matrix factorization aspect of the overall procedure. Equal attention must be given to both aspects, line search and linear solve, in the PCG variation.

Overall remarks on computational experiments. As mentioned above, the purpose of the experimental study is not to compare the PCG variation with the Cholesky variation in terms of overall efficiency. Rather, the purpose is to study the numerical behaviour of two possible implementations of the reflective Newton strategy. Our numerical results indicate that either implementation, PCG or Cholesky, is credible within this framework. The best choice between these two variations is problem dependent. In most of our experiments Cholesky required slightly fewer iterations but often significantly more time—witness Tables 4.4 and 4.12. On the other hand, there certainly are situations where the Cholesky variation is more economical. For example, in Table 4.13 we present the result of our experiments with a quadratic problem based on the evaluation of the Hessian matrix in [10]. In this case Cholesky is the clear winner in terms of operation count.

In the PCG variation, the fundamental subtask in both the line search and the linear solve operations is multiplication between the Hessian matrix and several vectors. The number of Hessian–vector products for our implementation of the PCG variation

TABLE 4.11
Indefinite problems. pctbnd = .9, $n = 1000$.

		Cholesky		PCG	
deg	cond	max	avg	max	avg
3	3	16	13.3	22	21
6	3	18	16	26	21.3
9	3	16	11	33	24
3	6	13	12	20	17.7
6	6	14	13	19	17
9	6	16	14.3	30	25.7
3	9	12	11.3	19	18
6	9	15	13.7	17	16.7
9	9	15	12.7	19	16.7

TABLE 4.12
Indefinite problems: Timing breakdown.

	Cholesky			PCG		
n	it	totM	totls	it	totM	totls
512	17	10.8	2.1	18	2.5	2.3
1000	19	38.2	4.9	25	7.4	6.8
1728	22	115.5	9.1	19	9.4	8.2
2744	24	396.6	16.4	24	18.3	16.7
8000	32	5030	69	31	78.4	67.8

averaged about $10 * it$ plus the total number of conjugate gradient iterations. Our implementation of the Cholesky variation averaged $10 * it$ Hessian–vector products.

5. Concluding remarks. The reflective Newton idea represents a new interior approach to box-constrained minimization problems. A convergence theory establishing global and second-order convergence results is presented in [9]; in the present paper we have provided a motivation of this approach and given results of computational experiments for sparse quadratic (indefinite) objective functions. The numerical results strongly support the notion that this approach is attractive for solving large sparse box-constrained quadratic programs.

Two variations of the reflective Newton method for box-constrained quadratic programs have been proposed in this paper. The Cholesky variation attempts a sparse Cholesky factorization in each iteration. The matrix to be factored has the same structure as the original Hessian matrix H and ultimately turns positive definite in the neighbourhood of a strong local minimizer. If the factorization fails then the matrix is not positive definite and a direction of negative curvature is usually available. The reflective Newton method uses this direction to help locate the next point. The PCG variation attempts to solve a local "Newton" system, each major iteration k, via PCG iterations. If negative curvature is found in the process, which may happen far from the solution (but not in a neighbourhood of the solution), then the PCG process is aborted and the reflective Newton method uses the negative direction to help locate the next point.

Regardless of which linear solver is used, the reflective Newton method then proceeds by solving a two-dimensional trust region problem to locate an initial feasible descent direction. A simple one-dimensional reflective path search is then performed to locate an improved point. We show how to efficiently implement an approximate search along this piecewise linear path.

Note that this approach does not involve an artificial "merit" function such as a

TABLE 4.13
Problem var.

	Cholesky		PCG	
n	it	m-flops	it	m-flops
1000	16	6.1	16	14.1
5000	17	32.9	14	62.7
10,000	17	65.9	17	225.3

barrier function—no delicate choice of barrier or penalty parameter is needed—nor does a "central path" need to be followed. These are potential advantages of this proposed new method. Moreover, our computational experiments to date indicate that this approach has considerable promise as a robust and efficient way to solve large box-constrained quadratic programs.

REFERENCES

[1] Å. BJORCK, *A direct method for sparse least squares problems with lower and upper bounds*, Numer. Math., 54 (1988), pp. 19–32.

[2] M. A. BRANCH, T. F. COLEMAN, AND Y. LI, *A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems*, Tech. report TR95-1525, Department of Computer Science, Cornell University, Ithaca, NY, 1995.

[3] R. H. BYRD, R. B. SCHNABEL, AND G. SHULTZ, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[4] T. F. COLEMAN AND L. A. HULBERT, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Math. Programming, 45 (1989), pp. 373–406.

[5] ———, *A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds*, SIAM J. Optim., 3 (1993), pp. 298–321.

[6] T. F. COLEMAN AND Y. LI, *A quadratically-convergent algorithm for the linear programming problem with lower and upper bounds*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM, Philadelphia, PA, 1990, pp. 49–57.

[7] ———, *A global and quadratically-convergent method for linear l_∞ problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1166–1186.

[8] ———, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, 56 (1992), pp. 189–222.

[9] ———, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds*, Math. Programming, 67 (1994), pp. 189–224.

[10] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[11] R. S. DEMBO AND U. TULOWITZKI, *On the Minimization of Quadratic Functions Subject to Box Constraints*, Tech. report B 71, School of Organization and Management, Yale University, New Haven, CT, 1983.

[12] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[13] F. FACCHINEI AND S. LUCIDI, *A class of penalty functions for optimization problems with bound constraints*, Optimization, 26 (1992), pp. 239–259.

[14] R. FLETCHER AND M. P. JACKSON, *Minimization of a quadratic function of many variables subject only to lower and upper bounds*, J. Inst. Math. Appl., 14 (1974), pp. 159–174.

[15] A. GEORGE AND J. W.-H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewoods Cliffs, NJ, 1981.

[16] J. GILBERT, C. MOLER, AND R. SCHREIBER, *Sparse matrices in* MATLAB*: Design and implementation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 333–356.

[17] P. GILL AND W. MURRAY, *Minimization Subject to Bounds on the Variables*, Tech. report NAC 71, National Physical Laboratory, England, 1976.

[18] D. GOLDFARB, *Curvilinear path steplength algorithms for minimization algorithms which use directions of negative curvature*, Math. Programming, 18 (1980), pp. 31–40.

[19] C.-G. HAN, P. M. PARDALOS, AND Y. YE, *Computational aspects of an interior point algo-*

rithm for quadratic programming problems with box constraints, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM, Philadelphia, PA, 1990, pp. 92–112.

[20] J. J. JÚDICE AND F. M. PIRES, *Direct methods for convex quadratic programs subject to box constraints*, Investigacao Peracional, 9 (1989), pp. 23–56.

[21] Y. LI, *A globally convergent method for l_p problems*, SIAM J. Optim., 3 (1993), pp. 609–629.

[22] P. LOTSTEDT, *Solving the minimal least squares problem subject to bounds on the variables*, BIT, 24 (1984), pp. 206–224.

[23] C. B. MOLER, J. LITTLE, S. BANGERT, AND S. KLEIMAN, *ProMatlab User's Guide*, MathWorks, Sherborn, MA, 1987.

[24] J. J. MORÉ AND G. TORALDO, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[25] ———, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[26] D. P. O'LEARY, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[27] U. ÖREBORN, *A direct method for sparse nonnegative least squares problems*, Ph.D. thesis, Department of Mathematics, Linköping University, Linköping, Sweden, 1986.

[28] S. POPE, *Application of the velocity-dissipation PDF model to inhomogeneous turbulent flows*, Phys. Fluids A, 3 (1991), pp. 1947–1957.

[29] T. STEIHAUG, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[30] E. K. YANG AND J. W. TOLLE, *A class of methods for solving large convex quadratic programs subject to box constraints*, Math. Programming, 51 (1991), pp. 223–228.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1059–1086, November 1996
© 1996 Society for Industrial and Applied Mathematics
010

CONVERGENCE PROPERTIES OF MINIMIZATION ALGORITHMS FOR CONVEX CONSTRAINTS USING A STRUCTURED TRUST REGION*

A. R. CONN†, NICK GOULD‡, A. SARTENAER§, AND PH. L. TOINT§

Abstract. In this paper, we present a class of trust region algorithms for minimization problems within convex feasible regions in which the structure of the problem is explicitly used in the definition of the trust region. This development is intended to reflect the possibility that some parts of the problem may be more accurately modelled than others, a common occurrence in large-scale nonlinear applications. After describing the structured trust region mechanism, we prove global convergence for all algorithms in our class.

Key words. trust region methods, structured problems, large-scale optimization, partial separability, convex constraints

AMS subject classifications. 90C30, 65K05

1. Introduction.

Trust region algorithms have enjoyed a long and successful history as tools for the solution of nonlinear, nonconvex, optimization problems. They have been studied and applied to unconstrained problems (see [7], [17], [25], [28]–[31], [34], [35], [38]) and to problems involving various classes of constraints, including simple bounds [6], [10], [11], [27], [32], convex constraints [2], [3], [14], [41], and nonconvex ones [5], [8], [16], [36], [44]. This long-lasting interest is probably justified by the attractive combination of a solid convergence theory, a noted algorithmic robustness, the existence of numerically efficient implementations, and an intuitively appealing motivation. The main idea behind trust region algorithms is that, if a nonlinear function (objective and/or constraints) is expensive to compute or difficult to handle explicitly, it should be replaced by a suitable *model*. This model is deemed to be trustworthy within a certain *trust region* around the current point. The trust region is defined by its shape and its *radius*. The minimization involving the difficult nonlinear function(s) is then replaced by a sequence of minimizations of the simpler model(s) within appropriate trust regions. The trust region radii are adjusted to reflect the agreement between the model and true functions as the process proceeds.

It is remarkable that, up to now, all algorithms that we are aware of use a *single* trust region radius to measure the degree of trustworthiness of the models employed, even if several different functions are involved. This choice is somewhat surprising if one admits that some of the modelled functions could be substantially "better behaved" than others in the same problem, as this implies that the region in which their

* Received by the editors August 28, 1992; accepted for publication (in revised form) August 18, 1995. This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under contract F49620-91-C-0079. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

† IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598 (arconn@watson.ibm.com).

‡ Rutherford Appleton Laboratory, Chilton, Oxfordshire, England (nimg@directory.rl.ac.uk). Current reports available by anonymous ftp from the directory "pub/reports" on camelot.cc.rl.ac.uk (internet 130.246.8.61).

§ Department of Mathematics, Facultés Universitaires Notre Dame de la Paix, 61 rue de Bruxelles, B-5000 Namur, Belgium (as@math.fundp.ac.be and pht@math.fundp.ac.be). Current reports available by anonymous ftp from the directory "pub/reports" on thales.math.fundp.ac.be (internet 138.48.4.14).

models can be trusted might also be substantially larger. In this context, the unstructured trust region choice might be viewed as a conservative strategy ensuring that *all* models may be trusted in what amounts to a safe minimal region. While this strategy might be reasonable for small problems, where each involved function depends on all the problem's variables, it is clearly questionable for large-scale applications, where each of the problem's functions typically depends only on a small number of variables. For instance, one might consider the minimization of an unconstrained objective function consisting of the sum of many quadratic and a few highly nonlinear terms, the latter involving a small subset of the variables. If a classical unstructured trust region algorithm, with a quadratic model, is used, the quadratic terms are perfectly modelled, but the steps that one can make at each iteration are (unnecessarily) limited by the highly nonlinear behaviour of a small subset of the variables.

It is the purpose of this paper to present and analyze a class of algorithms that uses the problem's *structure* in the *definition* of the trust region, allowing large steps in directions in which the model has proved to be adequate while restricting the movement in directions where the model seems unreliable. To be more precise, we will consider the problem of minimizing a *partially separable* objective function subject to convex constraints; we will then use the decomposition of the objective function into element functions as the basis for our structured trust region definition. The choice of the partially separable structure, a concept introduced in [21], is motivated by the very general geometric nature of this structure and by the increasing recognition of its practical use (see [4], [9], [12], [13], [18]–[20], [22], [26], [39], [42], [43], among others). More significantly, partial separability provides a decomposition of the considered nonlinear function into a linear combination of smaller *element functions*, each of which may then be modelled separately (see [40]). It is then quite natural to assign one trust region radius per element function and to decide on its increase or decrease separately. Because different element functions typically involve different sets of variables, each *element trust region* only restricts the components of the step corresponding to its *elemental variables*.

An obvious approach is to use the norm-scaling matrices allowed in the theory for unstructured trust region methods ([10], for instance) to account for differences in model adequacy among elements when constructing the trust region. This would be satisfactory if the existing theory did not require that the scaling matrices be of uniformly bounded condition number. Unfortunately, it is easy to conceive of instances where this is a severe handicap. For example, it would prevent the trust region radius of a well-modelled (perhaps linear or quadratic) element from increasing to infinity while at the same time ensuring that that of a badly behaved nonlinear element function remains of modest size. Moreover, this strategy may well cause numerical difficulties when used to solve the trust region problem. In fact, as we will shortly see, additional algorithmic safeguards are important when simultaneously handling trust regions of vastly different sizes. Thus, we do not consider such an approach further in this paper.

Section 2 of the paper presents the problem in more detail and the new class of algorithms using the principle of structured trust regions. Global convergence for all algorithms in the class is proved in §3. We briefly discuss the identification of active constraints in §4. We examine in §5 some extensions of the results of the previous sections. We finally give some comments and perspectives in §6.

2. Structured trust region for partially separable problems.

2.1. A structured model of the objective and the corresponding structured trust region.

2.1.1. The problem. The problem we consider is that of minimizing a smooth objective function subject to convex constraints. That is, we wish to solve the problem

$$\underset{x \in X}{\text{minimize}}\, f(x), \tag{2.1}$$

where X is a closed convex subset of $\mathbf{R}^n$. We denote the Euclidean inner product on $\mathbf{R}^n$ by $\langle \cdot, \cdot \rangle$, and the associated ℓ_2-norm by $\| \cdot \|$. Given Y a closed convex subset of $\mathbf{R}^n$, we define the operator $P_Y(\cdot)$ to be the orthogonal projection onto Y. We now list our additional assumptions on (2.1).

AS.1. X has a nonempty interior.

AS.2. f is bounded below on X.

AS.3. f is partially separable, which means that

$$f(x) = \sum_{i=1}^{p} f_i(x) \tag{2.2}$$

and that, for each $i \in \{1, \ldots, p\}$, there exists a subspace $\mathcal{N}_i \neq \{0\}$ such that, for all $w \in \mathcal{N}_i$ and all $x \in X$,

$$f_i(x + w) = f_i(x). \tag{2.3}$$

AS.4. For each $i \in \{1, \ldots, p\}$, f_i is continuously differentiable in an open set containing X and its gradient is uniformly bounded on X.

Note that we admit the case where X is unbounded or even identical to $\mathbf{R}^n$ itself, in which case we obtain an unconstrained problem. In relation to the partial separability of the objective function, we also consider the *range subspace* (see [23]) associated with each element function f_i, which is defined as

$$\mathcal{R}_i \overset{\text{def}}{=} \mathcal{N}_i^{\perp}. \tag{2.4}$$

We are mostly interested in the case where the dimension of each $\mathcal{R}_i$ is small compared to n. A commonly occurring case is when each element function f_i depends only on a small subset of the problem's variables; $\mathcal{R}_i$ is then the subspace spanned by the vectors of the canonical basis corresponding to the variables that occur in f_i (the elemental variables). The range of the projection operator $P_{\mathcal{R}_i}(\cdot)$ is therefore of low dimensionality. The reader is referred to [12] for a more detailed introduction to partially separable functions.

We note that f is invariant for any translation in the subspace $(\sum_{i=1}^{p} \mathcal{R}_i)^{\perp}$. We may therefore restrict our attention to the case where

$$\sum_{i=1}^{p} \mathcal{R}_i = \mathbf{R}^n \tag{2.5}$$

without loss of generality.

2.1.2. The element models. The algorithm we have in mind is iterative and generates feasible iterates (in the sense that all iterates belong to X). At iteration k we will associate a model $m_{i,k}$ with each element function f_i. This model, defined

on $\mathcal{R}_i$ in a neighbourhood of the projection of the kth iterate x_k on this subspace, is meant to approximate f_i for all x in the *element trust region*

$$B_{i,k} \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\| \leq \Delta_{i,k}\}, \tag{2.6}$$

where $\Delta_{i,k} > 0$ is the ith trust region radius at iteration k and the norm $\|\cdot\|$ is chosen to be the usual Euclidean norm in order to simplify the exposition. In what follows, we will slightly abuse notation by writing $m_{i,k}(x)$ for an $x \in \mathbf{R}^n$, instead of the more complete $m_{i,k}(P_{\mathcal{R}_i}(x))$. We will furthermore assume that each model $m_{i,k}$ ($i \in \{1, \ldots, p\}, k = 0, 1, 2, \ldots$) is differentiable and has Lipschitz continuous first derivatives on an open set containing $B_{i,k}$ and that

$$m_{i,k}(x_k) = f_i(x_k) \quad (i \in \{1, \ldots, p\}, k = 0, 1, 2, \ldots). \tag{2.7}$$

Moreover, we assume that $g_{i,k} \stackrel{\text{def}}{=} \nabla m_{i,k}(x_k) \in \mathcal{R}_i$ approximates $\nabla f_i(x_k) \in \mathcal{R}_i$ in the sense that, for all $i \in \{1, \ldots, p\}$ and all k,

$$\|e_{i,k}\| \leq \kappa_1 \Delta_{\min,k}, \tag{2.8}$$

where $e_{i,k} \stackrel{\text{def}}{=} g_{i,k} - \nabla f_i(x_k)$, where $\kappa_1 > 0$ is a constant and where $\Delta_{\min,k}$ is defined by

$$\Delta_{\min,k} \stackrel{\text{def}}{=} \min_{i \in \{1,\ldots,p\}} \Delta_{i,k}. \tag{2.9}$$

Condition (2.8) is quite weak, as it merely requires that the first-order information be reasonably accurate whenever some trust region radius is small (i.e., the corresponding model fits badly). Indeed, one expects the coherency of this first-order behaviour to be of crucial importance in such cases. Further arguments supporting a choice similar to (2.8) for problems with convex constraints are presented in [14].

Amongst the most commonly used element models, linear or quadratic approximations are preeminent. One can, for instance, consider the quadratic model given by the first three terms of the element function Taylor series around the current iterate. Another popular choice is a quadratic model where the second derivative matrix is recurred using quasi-Newton formulae.

2.1.3. The overall model and trust region. With all the element models at hand, we are now in position to define the overall model at iteration k, denoted m_k, whose purpose is to approximate the overall objective function f in a neighbourhood of the current iterate x_k. From (2.2), it is natural to use the overall model

$$m_k(x) \stackrel{\text{def}}{=} \sum_{i=1}^{p} m_{i,k}(x) \tag{2.10}$$

for all x in the overall trust region defined by

$$B_k = \bigcap_{i \in \{1,\ldots,p\}} B_{i,k}. \tag{2.11}$$

Indeed B_k is the intersection of all element trust regions, that is, the region in which all element models may be trusted, irrespective of the additional limitation possibly imposed by the feasible set X.

Of course, the actual shape of the trust region B_k is determined by the choice of the Euclidean norm; it corresponds to the intersection of cylinders whose axes are aligned with the subspaces $\mathcal{N}_i$ and whose radii reflect the quality of the element models: large in subspaces where the element models predict the element function correctly and smaller in subspaces where the prediction is poorer. In practice, one might wish to choose other norms, such as the ℓ_∞-norm. In this case, and assuming that the subspaces $\mathcal{R}_i$ are spanned by subsets of the canonical basis vectors, the shape of the trust region is that of a box, the length of whose sides again reflects the quality of the element models. The extension of the theory to more general norms is considered in §5.4.

2.1.4. Curvature. We now follow [14] and [41] and define the *generalized Rayleigh quotient* of f at x along $s \neq 0$ by

$$\omega(f,x,s) \stackrel{\text{def}}{=} \frac{2}{\|s\|^2}[f(x+s) - f(x) - \langle \nabla f(x), s\rangle]. \tag{2.12}$$

Obviously, this definition is valid only if s is such that $x+s$ belongs to the domain of the definition of f. Note that, by convention,

$$\omega(f,x,s) = 0 \text{ whenever } s = 0. \tag{2.13}$$

If we assume that f is twice continuously differentiable, the mean value theorem (see [24]) implies that

$$\omega(f,x,s) = 2\int_0^1 \int_0^1 t \frac{\langle s, \nabla^2 f(x+tvs)s\rangle}{\|s\|^2}\, dv\, dt. \tag{2.14}$$

Furthermore, if f is quadratic, then one easily verifies that $\omega(f,x,s)$ is independent of x and is equal to the Rayleigh quotient of the matrix $\nabla^2 f$ in the direction s. We note that, because of AS.4, $\omega(f_i,x,s)$ is bounded by some constant $L_i \geq 0$ (see [24]). Hence we obtain that

$$|\omega(f_i,x,s)| \leq \max\left\{1, \max_{i\in\{1,\ldots,p\}} L_i\right\} \stackrel{\text{def}}{=} L \tag{2.15}$$

for all $x, x+s \in X$ and all $i \in \{1,\ldots,p\}$. The quantity that we need in our algorithm statement and analysis is a monotonically increasing upper bound on the magnitude of the generalized Rayleigh quotient $\omega(m_{i,k}, x_k, s_{i,k})$ defined by

$$\beta_k \stackrel{\text{def}}{=} 1 + \max_{\substack{q\in\{0,\ldots,k\}\\ i\in\{1,\ldots,p\}}} |\omega(m_{i,q}, x_q, s_{i,q})| \geq 1, \tag{2.16}$$

where $s_{i,k} \stackrel{\text{def}}{=} P_{\mathcal{R}_i}(s_k)$ for s_k the actual trial step computed by the algorithm, as defined below. The quantity $\omega(m_{i,k}, x_k, s_{i,k})$ measures the curvature of the model $m_{i,k}$ in the direction of the trial step s_k. If quadratic models $m_{i,k}$ are considered, an upper bound on β_k is given by the largest singular value of all Hessian matrices, plus one. We will assume that our choice of models is such that this curvature does not increase too fast, which could lead to premature convergence of the algorithm to a noncritical point (see [41]). More precisely, we make the following assumption, as in [14], [10], [35], and [41].

AS.5.

$$\sum_{i=0}^{\infty} \frac{1}{\beta_k} = +\infty. \tag{2.17}$$

This condition is weaker than the common assumption that the model's second-derivative matrices are uniformly bounded [32], which holds, for instance, for the classical Newton method, where quadratic models using analytical second derivatives are used on a compact domain. It is also weaker than the condition

$$|\omega(m_{i,k}, x_k, s_{i,k})| \leq c_0 k \tag{2.18}$$

for some constant $c_0 > 0$, which holds in the case where quadratic element models are used and updated using either the Broyden–Fletcher, Goldfarb, Shanno (BFGS) or the safeguarded symmetric rank one quasi-Newton formulae.

2.1.5. Criticality. Before we can describe our algorithm in detail, we also need a *criticality criterion* for our problem. A critical point of our problem is a feasible point x where the negative gradient of the objective function $-\nabla f(x)$ belongs to the *normal cone* of X at $x \in X$, which is defined by

$$\mathcal{N}(x) \stackrel{\text{def}}{=} \{y \in \mathbf{R}^n \mid \langle y, u - x\rangle \leq 0 \ \forall u \in X\}. \tag{2.19}$$

The associated *tangent cone* of X at $x \in X$ is the polar of $\mathcal{N}(x)$, that is,

$$\mathcal{T}(x) \stackrel{\text{def}}{=} \mathcal{N}(x)^0 = \text{closure}\{\lambda(u - x) \mid \lambda \geq 0 \ \text{ and } \ u \in X\}. \tag{2.20}$$

Thus every measure of criticality has to depend on the (differentiable) objective f and on the geometry of the feasible set at the current point. We will use the symbol $\alpha(x, f, X)$ to denote such a criticality measure.

AS.6. The criticality measure $\alpha(x, h, X)$ is nonnegative for all $x \in X$ and all functions h differentiable in an open neighbourhood of x. Moreover, $\alpha(x, h, X) = 0$ if and only if x is critical for the problem

$$\underset{x \in X}{\text{minimize}} \ h(x). \tag{2.21}$$

But, within the algorithm, only approximate gradient vectors might be available, namely, the vectors g_k and $g_{i,k}$, the gradients of the models. It is therefore natural to use

$$\alpha_k \stackrel{\text{def}}{=} \alpha(x_k, m_k, X), \tag{2.22}$$

the criticality measure for the problem

$$\underset{x \in X}{\text{minimize}} \ m_k(x), \tag{2.23}$$

as an approximate criticality measure for (2.1). Note that $\alpha_k > 0$ implies that $g_k \neq 0$.

In unconstrained optimization, one typically chooses

$$\alpha_k = \|g_k\|, \tag{2.24}$$

the obvious criticality measure (see [31] or [34]). When bound constraints are present, the choice

$$\alpha_k = \|P_X(x_k - g_k) - x_k\| \tag{2.25}$$

is made in [10]. For the infinite-dimensional case, the definition

$$\alpha_k = \|P_X(x_k - g_k) - x_k\|^2 \tag{2.26}$$

is used in [41]. For the case where convex constraints are considered,

$$\alpha_k = \frac{\|P_X(x_k - t_k^C g_k) - x_k\|}{t_k^C} \tag{2.27}$$

is chosen in [32], where $t_k^C > 0$ is the line-coordinate of the so-called generalized Cauchy point to be discussed below. In a similar context,

$$\alpha_k = \left| \min_{\substack{x_k + d \in X \\ \|d\| \leq 1}} \langle g_k, d \rangle \right| \tag{2.28}$$

is used in [14].

2.2. Ensuring sufficient model decrease.

2.2.1. An overview of the classical sufficient decrease condition. A key to trust region algorithms is to choose a step s_k at iteration k that is guaranteed to provide a sufficient decrease on the overall objective function model m_k. In other words, a step such that

$$\delta m_k \stackrel{\text{def}}{=} m_k(x_k) - m_k(x_k + s_k) \tag{2.29}$$

is sufficiently positive, given the value of a suitable criticality measure α_k satisfying AS.6. This concept of sufficient decrease is usually made more formal by introducing the notion of the *(generalized) Cauchy point.* This remarkable point, denoted x_k^C, is typically computed by trust region algorithms as a point on (or close to) the projected gradient path $P_X(x_k - tg_k)$ $(t \geq 0)$ that is also within the trust region and sufficiently reduces the overall model in the sense that

$$m_k(x_k) - m_k(x_k^C) \geq \bar{\kappa}_2 \frac{\alpha_k^2}{\beta_k}, \tag{2.30}$$

where $\bar{\kappa}_2 > 0$ is a constant and α_k is a criticality measure satisfying AS.6. However, such a point may not exist when the trust region radius Δ_k is small compared with α_k^2/β_k. In this case, the generalized Cauchy point is chosen as (or close to) the intersection of the projected gradient path with the boundary of the trust region, yielding an inequality of the form

$$m(x_k) - m(x_k^C) \geq \bar{\kappa}_2 \alpha_k \Delta_k. \tag{2.31}$$

A point on the projected gradient path satisfying (2.30) may also fail to exist because the projected gradient path itself ends on the boundary of X, well inside the trust region. In that case, this endpoint (or another feasible point close to it) is typically chosen as the generalized Cauchy point, and it is then typically shown that

$$m(x_k) - m(x_k^C) \geq \bar{\kappa}_2 \alpha_k. \tag{2.32}$$

One then ensures the sufficient decrease by requiring that the chosen step s_k produces at least a fixed fraction of the overall model reduction achieved by the generalized Cauchy point, which is to say that

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_k, 1\right\}, \tag{2.33}$$

where $\kappa_2 \in (0, \bar{\kappa}_2]$.

Many variants on the above scheme exist in the literature for the unstructured trust region case. All of these variants ensure that a suitable step is found after a finite number of trials. The best known is for unconstrained problems when the ℓ_2-norm is used to define the trust region shape. In that case, the projected gradient path is simply given by all negative multiples of the gradient g_k and the Cauchy point is simply the point that minimizes the model m_k in the intersection of the steepest descent direction and the trust region (see, for instance, [34] and [37]). When other norms are used, for example the ℓ_∞-norm, one can then choose either to minimize the model in the intersection of this steepest descent direction and the trust region, as before (see [10]), or to "bend" the projected gradient path onto the boundary of the trust region and to choose the generalized Cauchy point as a point which satisfies classical Goldstein-type linesearch conditions along that path while staying within the trust region (see [33] and [41]). Both of these latter strategies are used in the LANCELOT software [13]. When additional convex constraints are present, the projected gradient path is additionally bent to follow the boundary of the feasible domain. Thus the philosophy is the same in that (2.33) is guaranteed in the above cases. Indeed satisfaction of this condition has been derived for each of the choices (2.24)–(2.28) for α_k in the papers where they were respectively introduced.

2.2.2. Sufficient decrease for the structured model and trust region. We will use a similar approach in our structured model and trust region framework to determine what is a sufficient decrease of the overall model m_k within the region B_k, whose shape is chosen to reflect the structure of the problem. Special care is needed because this region might be very asymmetric in the sense that it may allow very large steps in some directions and only very short ones in others. As a consequence, we have to adapt the notion of trust region radius to our context and adequately reformulate condition (2.33).

From a practical point of view, one might use a two-staged approach. In this, one first aims to find a step producing a sufficient model decrease in a smaller, but more symmetric, region. Following this, one then allows the step to increase within the trust region while maintaining control over the model decrease.

To be specific, let

$$B_{\min,k} \stackrel{\text{def}}{=} B_k \cap \{x \in \mathbf{R}^n \mid \|x - x_k\| \leq \Delta_{\min,k}\} \tag{2.34}$$

be the trust region whose radius is determined by the possibly most nonlinear part of the model. Applying the results discussed in the previous section after condition (2.33), one may deduce that it is possible to find, in a finite number of trials, a step $s_{\min,k}$ such that $x_k + s_{\min,k} \in B_{\min,k} \cap X$ and

$$m_k(x_k) - m_k(x_k + s_{\min,k}) \geq \bar{\kappa}_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_{\min,k}, 1\right\}, \tag{2.35}$$

for some suitably chosen criticality measure α_k satisfying AS.6 and some constant $\bar{\kappa}_2 > 0$.

However, the restriction that the length of $s_{\min,k}$ be bounded by $\Delta_{\min,k}$ makes the whole exercise of shaping B_k to reflect the problem's structure entirely irrelevant. One might therefore be prepared to accept a larger step provided it remains feasible, within the trust region B_k, and produces a further significant model decrease. More specifically, we allow our algorithm to choose any step s_k such that $x_k + s_k \in B_k \cap X$ and which guarantees that

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \max[\Delta_{\min,k}, \|s_k\|], 1\right\} \tag{2.36}$$

for some $\kappa_2 \in (0, \bar{\kappa}_2]$.

Note that, since (2.36) holds for $s_k = s_{\min,k}$, this condition can therefore be achieved in practice after a finite number of trials. Observe also that (2.36) is fundamentally different from an angle test of the form

$$|\langle g_k, s_k \rangle| \geq \zeta \|g_k\| \|s_k\| \quad (\zeta \in (0,1)) \tag{2.37}$$

as (2.36) does not prevent s_k from being orthogonal to the steepest descent direction, so long as a sufficient model reduction is obtained. This is useful because such a step may occur when moving away from a saddle point of the objective function. Finally note that, as expected, (2.36) reduces to (2.33) in the case where only one trust region is considered.

2.3. A class of structured trust region algorithms. We now describe the class of algorithms that we consider for solving (2.1). Besides κ_1 used in (2.8) and κ_2 used in (2.36), it depends on the constants

$$0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3, \tag{2.38}$$

$$0 < \eta_1 \leq \eta_2 < \eta_3 < 1, \tag{2.39}$$

and

$$0 < \mu_1 < \mu_2 < 1. \tag{2.40}$$

In addition to the above conditions, we also require a compatibility condition between the η_i's and the μ_i's. Specifically, we request that

$$\eta_2 - \eta_1 \geq \mu_1 + \mu_2. \tag{2.41}$$

Typical values for these constants are $\kappa_1 = 0.1$, $\kappa_2 = 0.01$, $\gamma_1 = 0.1$, $\gamma_2 = 0.5$, $\gamma_3 = 2$, $\eta_1 = 0.01$, $\eta_2 = 0.25$, $\eta_3 = 0.75$, $\mu_1 = 0.05$, and $\mu_2 = 0.1$.

ALGORITHM.

step 0: initialization.

The starting point $x_0 \in X$ is given, together with the element function values $\{f_i(x_0)\}_{i=1}^p$ and the initial trust region radii $\{\Delta_{i,0}\}_{i=1}^p$. Set $k = 0$.

step 1: model choice.

For $i \in \{1, \ldots, p\}$, choose the model $m_{i,k}$ of the element function f_i in the trust region $B_{i,k}$ centered at x_k (as defined in (2.6)), satisfying (2.7) and (2.8).

step 2: determination of the step.
Choose a step s_k such that the sufficient decrease condition (2.36) holds and

$$x_k + s_k \in B_k \cap X. \tag{2.42}$$

step 3: measure overall model fit.
If

$$\delta f_k \stackrel{\text{def}}{=} f(x_k) - f(x_k + s_k) \geq \eta_1 \delta m_k \tag{2.43}$$

then

$$x_{k+1} = x_k + s_k, \tag{2.44}$$

else

$$x_{k+1} = x_k. \tag{2.45}$$

step 4: update the element trust region radii.
Denote the achieved changes in the element functions and their models by

$$\delta f_{i,k} \stackrel{\text{def}}{=} f_i(x_k) - f_i(x_k + s_k), \quad i \in \{1, \ldots, p\}, \tag{2.46}$$

and

$$\delta m_{i,k} \stackrel{\text{def}}{=} m_{i,k}(x_k) - m_{i,k}(x_k + s_k), \quad i \in \{1, \ldots, p\}, \tag{2.47}$$

respectively. Then define the set of *negligible* elements at iteration k as

$$N_k \stackrel{\text{def}}{=} \left\{ i \in \{1, \ldots, p\} \;\middle|\; |\delta m_{i,k}| \leq \frac{\mu_1}{p} \delta m_k \right\} \tag{2.48}$$

and the set of *meaningful* elements as its complement, that is,

$$M_k = \{1, \ldots, p\} \setminus N_k. \tag{2.49}$$

Then, for each $i \in \{1, \ldots, p\}$, perform the following.
Case 1: $i \in M_k$.

- If

$$\delta f_{i,k} \geq \delta m_{i,k} - \frac{1 - \eta_3}{p} \delta m_k \tag{2.50}$$

and (2.43) both hold, then choose

$$\Delta_{i,k+1} \in [\Delta_{i,k}, \gamma_3 \Delta_{i,k}]. \tag{2.51}$$

- If (2.50) holds but (2.43) fails then choose

$$\Delta_{i,k+1} = \Delta_{i,k}. \tag{2.52}$$

- If (2.50) fails but

$$\delta f_{i,k} \geq \delta m_{i,k} - \frac{1 - \eta_2}{p} \delta m_k \tag{2.53}$$

holds, then choose

$$\Delta_{i,k+1} \in [\gamma_2 \Delta_{i,k}, \Delta_{i,k}]. \tag{2.54}$$

- If (2.53) fails, then choose

$$\Delta_{i,k+1} \in [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}]. \tag{2.55}$$

Case 2: $i \in N_k$.

- If

$$|\delta f_{i,k}| \leq \frac{\mu_2}{p} \delta m_k, \tag{2.56}$$

and (2.43) both hold, then choose

$$\Delta_{i,k+1} \in [\Delta_{i,k}, \gamma_3 \Delta_{i,k}]. \tag{2.57}$$

- If (2.56) holds but (2.43) fails, then choose

$$\Delta_{i,k+1} = \Delta_{i,k}. \tag{2.58}$$

- If (2.56) fails, then choose

$$\Delta_{i,k+1} \in [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}]. \tag{2.59}$$

Increment k by one and return to step 1.

End of Algorithm

As is traditional in trust region algorithms, we will call an iteration *successful* if the test (2.43) is satisfied, that is, when the achieved objective reduction δf_k is large enough compared to the reduction δm_k predicted by the overall model. If (2.43) fails, the iteration is said to be *unsuccessful*. In what follows, we will denote by $\mathcal{S}$ the set of all successful iterations.

We now comment on various aspects of the algorithm.

1. The algorithm is constructed in such a way that a successful step is always possible for sufficiently small trust region radii if the current iterate x_k is not critical. This result is formally proved in Corollary 3.6.
2. The choice of the element models $m_{i,k}$ is left rather open in the above description. It clearly needs to be made precise for any practical implementation of the algorithm. One common choice would be to set

$$m_{i,k}(x_k + s) = f_i(x_k) + \langle g_{i,k}, s \rangle + \frac{1}{2} \langle s, H_{i,k} s \rangle, \tag{2.60}$$

where $H_{i,k}$ is a symmetric approximation to $\nabla^2 f_i(x_k)$ whose nullspace contains the subspace $\mathcal{N}_i$. In particular, Newton's method corresponds to the choice $g_{i,k} = \nabla f_i(x_k)$ and $H_{i,k} = \nabla^2 f_i(x_k)$, which is guaranteed to satisfy this latter condition. Another possible choice is $m_{i,k}(x_k + s) = f_i(x_k + s)$, which may be attractive for the simpler element functions. In this case, the model's fit to the true function is always good for the ith element and the algorithm guarantees that the $\Delta_{i,k}$ form a nondecreasing sequence.
3. If the model change for an element is negligible, that is, small compared to the overall predicted change, we do not need to restrict its element trust region size unless the true element change is relatively large compared with the same overall predicted change. We can therefore afford to ignore negligible items until they stop being relatively negligible, something which is inevitable when convergence occurs. Hence our distinction between "negligible" elements (in N_k) and "meaningful" ones (in M_k).
   Condition (2.41) can be viewed in this context as a guarantee that a new iterate will be accepted in (2.43) whenever the model reduction obtained for all meaningful elements is also acceptable (i.e., (2.53) holds for all $i \in M_k$), irrespective of the contribution of the negligible ones. This interpretation is clarified in Lemma 2.2.

4. The apparent intricacy of (2.50) and (2.53) is caused by two complications which arise in the context of multiple elements. The first is that, although (2.36) ensures that δm_k is always positive, we may not assume in general that the same is true for $\delta m_{i,k}$. The second is that possible cancellation between elements makes it necessary to consider the accuracy of model fit for an element to be relative to the *overall* model fit. Indeed, requiring small relative errors for models with very large values may result in large absolute errors. If δm_k is small, these large errors will then cause δm_k to be a poor prediction of δf_k and the iteration might be unsuccessful. This explains why the perhaps more intuitive tests

$$\delta f_{i,k} \geq \delta m_{i,k} - (1-\eta_j)|\delta m_{i,k}| \quad (j=2,3) \tag{2.61}$$

cannot be used instead of (2.53) ($j=2$) and (2.50) ($j=3$).
Observe also that conditions (2.50) and (2.53) reduce to the familiar

$$\delta f_k \geq \eta_j \delta m_k \quad (j=2,3) \tag{2.62}$$

when $p=1$.

5. Note again the consistency between the trust region radii updates in step 4 and the case where $p=1$. In this latter case, the set N_k is always empty and (2.50) then implies (2.43), because of (2.39). Equation (2.52) is thus never invoked.
6. No stopping criterion has been explicitly included in our algorithm description. This is adequate for the theoretical analysis that we consider in the present paper, where we are interested in the asymptotic behaviour of the method, but it should be completed for any practical use. The choice of a particular stopping criterion will depend on the type of models being used.
7. The mechanism that we specified for updating the trust region radii does not exclude the additional requirement that the radii be uniformly bounded, if that is judged suitable for the type of models used. In practice, keeping the radii bounded is essential to prevent numerical overflow.
8. One possible implementation of step 2 first computes a feasible step s_k^C that minimizes $m_k(P_X(x_k - tg_k))$ within a trust region of radius $\Delta_{\min,k}$. Note that s_k^C satisfies (2.35) and (2.36) by construction. This step may then subsequently be increased by progressing further along the arc $P_X(x_k - tg_k) - x_k$ so long as the overall model m_k continues to decrease and (2.36) holds. Additional decrease in m_k may then be obtained (for instance by applying conjugate-gradient steps) provided condition (2.36) is maintained.

Before starting our global convergence analysis, we first state, for future reference, some properties that result from the mechanism of the algorithm.

LEMMA 2.1. *Assume that* AS.3 *holds. At each iteration k of the algorithm,*

1. *M_k contains at least one element. Furthermore,*

$$\left(1 - \frac{p-1}{p}\mu_1\right)\delta m_k \leq \sum_{i \in M_k} \delta m_{i,k} \leq \left(1 + \frac{p-1}{p}\mu_1\right)\delta m_k. \tag{2.63}$$

2.

$$\gamma_1 \Delta_{i,k} \leq \Delta_{i,k+1} \leq \gamma_3 \Delta_{i,k} \tag{2.64}$$

for all $i \in \{1, \ldots, p\}$.

Proof. The first result immediately follows from the definition of N_k and the inequality $\mu_1 < 1$. One then deduces that N_k contains at most $p-1$ elements. Hence,

$$\delta m_k = \sum_{i \in M_k} \delta m_{i,k} + \sum_{i \in N_k} \delta m_{i,k} \leq \sum_{i \in M_k} \delta m_{i,k} + \mu_1 \frac{|N_k|}{p} \delta m_k, \tag{2.65}$$

from which the first part of (2.63) may be deduced. The second inequality in this result is obtained from

$$\sum_{i \in M_k} \delta m_{i,k} = \delta m_k - \sum_{i \in N_k} \delta m_{i,k} \leq \delta m_k + \sum_{i \in N_k} |\delta m_{i,k}|, \tag{2.66}$$

the relation (2.48), and $|N_k| \leq p-1$. The bound (2.64) results from (2.51), (2.54), (2.55), (2.57), and (2.59). ☐

We also investigate the coherency between the measure of fit for individual elements and that for the overall model.

LEMMA 2.2. *Assume* AS.3 *holds and that, at iteration k of the algorithm,* (2.53) *holds for all $i \in M_k$ and* (2.56) *holds for all $i \in N_k$. Then iteration k is successful; i.e., $k \in \mathcal{S}$.*

Proof. Because (2.53) holds for $i \in M_k$, one has that

$$\sum_{i \in M_k} \delta f_{i,k} \geq \sum_{i \in M_k} \delta m_{i,k} - (1-\eta_2) \frac{|M_k|}{p} \delta m_k \geq \left(\eta_2 - \frac{p-1}{p} \mu_1 \right) \delta m_k \tag{2.67}$$

for all such i, where we used the inequality $|M_k| \leq p$ and Lemma 2.1 to deduce the second inequality. On the other hand, since (2.56) holds for $i \in N_k$, one obtains for these i that

$$\sum_{i \in N_k} |\delta f_{i,k}| \leq \frac{p-1}{p} \mu_2 \delta m_k, \tag{2.68}$$

where we used item 1 of Lemma 2.1 to bound $|N_k|$. Now,

$$\delta f_k = \sum_{i \in M_k} \delta f_{i,k} + \sum_{i \in N_k} \delta f_{i,k} \geq \sum_{i \in M_k} \delta f_{i,k} - \sum_{i \in N_k} |\delta f_{i,k}|. \tag{2.69}$$

Combining this last inequality with (2.67) and (2.68) gives that

$$\delta f_k \geq \left(\eta_2 - \frac{p-1}{p} \mu_1 - \frac{p-1}{p} \mu_2 \right) \delta m_k, \tag{2.70}$$

which then yields (2.43) because of (2.41). ☐

We observe from this proof that the weaker condition

$$\eta_2 - \eta_1 \geq \frac{p-1}{p} (\mu_1 + \mu_2) \tag{2.71}$$

could be imposed instead of (2.41). However (2.71), and hence the setting of the algorithm's constants, would then be problem dependent, which one might consider to be undesirable.

Of course, (2.53) holds whenever (2.50) holds because of (2.39). Lemma 2.2 therefore shows that (2.43) is coherent with the measure of the fit between the element models and element functions.

3. Global convergence. We now study the convergence properties of the class of algorithms that we introduced in the preceding section. Our analysis follows the pattern of similar proofs with an unstructured trust region (see [14] or [41]). The central idea in the proof is that the algorithm will continue to make progress as long as a critical point is not reached. We first start by bounding the error between the true element functions and their models. We next derive a lower bound on the size of the smallest trust region radius at a noncritical point. This lower bound ensures that the trust region constraint will not prevent further progress toward a critical point. Only with this bound can we then prove that limit points of the sequence of iterates produced by the algorithm are indeed critical for the models used. We close the section by deriving some simple consequences of these results on the criticality of the limit points for the true objective function.

We first start by bounding the error made between the model of any element function and the element function itself at $x_k + s_k$.

LEMMA 3.1. *Assume that* AS.4 *holds and consider a sequence* $\{x_k\}$ *of iterates generated by the algorithm. Then there exists a positive constant* $c_1 \geq 1$ *such that*

$$|f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| \leq c_1 \beta_k \Delta_{i,k}^2 \tag{3.1}$$

for all $i \in \{1, \ldots, p\}$ *and all* k.

Proof. We first observe that, for each $i \in \{1, \ldots, p\}$ and for all k, the definition (2.12), (2.7), and the Cauchy–Schwarz inequality imply that

(3.2)

$$\begin{aligned} |f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| &= |f_i(x_k + s_{i,k}) - m_{i,k}(x_k + s_{i,k})| \\ &\leq |\langle \nabla f_i(x_k) - g_{i,k}, s_{i,k} \rangle| \\ &\quad + \frac{1}{2}\|s_{i,k}\|^2 |\omega(f_i, x_k, s_{i,k}) - \omega(m_{i,k}, x_k, s_{i,k})| \\ &\leq \|e_{i,k}\| \, \|s_{i,k}\| \\ &\quad + \frac{1}{2}\|s_{i,k}\|^2 (|\omega(f_i, x_k, s_{i,k})| + |\omega(m_{i,k}, x_k, s_{i,k})|). \end{aligned}$$

But $\|s_{i,k}\| \leq \Delta_{i,k}$ because of (2.6), and hence we obtain from (2.8), (2.15), and (2.16) that

$$|f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| \leq \kappa_1 \Delta_{\min,k} \Delta_{i,k} + \frac{1}{2}(L + \beta_k)\Delta_{i,k}^2. \tag{3.3}$$

Using (2.9), this then yields (3.1) with

$$c_1 \stackrel{\text{def}}{=} \kappa_1 + \frac{1}{2}(L + 1) \geq 1, \tag{3.4}$$

where the last inequality results from (2.15). ∎

We now derive an upper bound on the change predicted for an element at a noncritical point as a function of the size of the step in the corresponding range subspace.

LEMMA 3.2. *Assume that* AS.1, AS.3, *and* AS.4 *hold. Consider iteration* k *of the algorithm and assume that, for some* $i \in \{1, \ldots, p\}$,

$$\beta_k \Delta_{i,k} \leq 1. \tag{3.5}$$

Then one has that

$$|\delta m_{i,k}| \leq c_2 \|s_{i,k}\| \tag{3.6}$$

for some constant $c_2 > 0$ *independent of* i *and* k.

Proof. We first note that (2.9), (2.16), and (3.5) imply that

$$\Delta_{\min,k} \leq 1. \tag{3.7}$$

Using (2.12) and (2.16), we also obtain that

(3.8)
$$|\delta m_{i,k}| \leq |\langle g_{i,k}, s_{i,k}\rangle| + \frac{1}{2}\beta_k \|s_{i,k}\|^2 \leq |\langle \nabla f_i(x_k), s_{i,k}\rangle| + |\langle e_{i,k}, s_{i,k}\rangle| + \frac{1}{2}\beta_k \|s_{i,k}\|^2.$$

Remembering now (2.8), (2.6), (3.5), and (3.7), we can deduce that

$$\begin{aligned} |\delta m_{i,k}| &\leq \max_{x\in\mathcal{R}_i\cap X} (\|\nabla f_i(x)\|)\, \|s_{i,k}\| + \kappa_1 \Delta_{\min,k} \|s_{i,k}\| + \frac{1}{2}\beta_k \|s_{i,k}\|^2 \\ &\leq [\max_{x\in\mathcal{R}_i\cap X} (\|\nabla f_i(x)\|) + \kappa_1 + \frac{1}{2}]\, \|s_{i,k}\|. \end{aligned} \tag{3.9}$$

Inequality (3.9) then gives (3.6) with

$$c_2 \stackrel{\text{def}}{=} \max_{\substack{x\in\mathcal{R}_i\cap X \\ i\in\{1,\ldots,p\}}} (\|\nabla f_i(x)\|) + \kappa_1 + \frac{1}{2}. \qquad \square \tag{3.10}$$

We next prove the important fact that, so long as a critical point has not been determined, the trust region radii stay sufficiently bounded away from zero, therefore allowing further progress to be made.

LEMMA 3.3. *Assume that* AS.1–AS.4 *hold. Consider a sequence* $\{x_k\}$ *of iterates generated by the algorithm and assume that there exists a constant* $\epsilon > 0$ *such that*

$$\alpha_k \geq \epsilon \tag{3.11}$$

for all k. *Then there is a constant* $c_3 > 0$ *such that*

$$\Delta_{\min,k} \geq \frac{c_3}{\beta_k} \tag{3.12}$$

for all k.

Proof. Assume, without loss of generality, that

$$\epsilon < \min\{1, \beta_0 \Delta_{\min,0}\}. \tag{3.13}$$

To derive a contradiction, assume that there exists a k such that

$$\beta_k \Delta_{\min,k} < \gamma_1 \min\left\{\epsilon, \frac{\mu_1 c_4^2 (1-\eta_3)}{c_1 c_2 p^2}, \frac{c_4(\mu_2-\mu_1)}{c_1 p}\right\} \stackrel{\text{def}}{=} c_3, \tag{3.14}$$

where $c_4 \stackrel{\text{def}}{=} \kappa_2 \gamma_1 \epsilon$. Now define r to be the smallest iteration number such that (3.14) holds. (Note that $r \geq 1$ because of (3.13) and the inequality $\gamma_1 < 1$.) Also fix i such

that $\Delta_{\min,r} = \Delta_{i,r}$. The monotonic nature of the sequence $\{\beta_k\}$ and the bound (2.64) then ensure that

$$\beta_{r-1}\Delta_{\min,r-1} \leq \beta_{r-1}\Delta_{i,r-1} \leq \beta_r \frac{\Delta_{i,r}}{\gamma_1} \leq \frac{c_3}{\gamma_1} \leq \epsilon < 1, \tag{3.15}$$

where we used (3.14) and the inequality (3.13). We note that the definitions of i and r give that

$$\beta_r \Delta_{i,r} = \beta_r \Delta_{\min,r} < \beta_{r-1}\Delta_{\min,r-1}, \tag{3.16}$$

which in turn implies that $\Delta_{\min,r-1} > \Delta_{i,r}$ because of the monotonic nature of the sequence $\{\beta_k\}$. Using this inequality with (2.36), (3.11), and (3.15), we obtain that

$$\begin{aligned} \delta m_{r-1} &\geq \kappa_2 \epsilon \min\left\{\frac{\epsilon}{\beta_{r-1}}, \max[\Delta_{\min,r-1}, \|s_{r-1}\|], 1\right\} \\ &\geq \kappa_2 \epsilon \min\left\{\frac{\epsilon}{\beta_{r-1}}, \Delta_{\min,r-1}, 1\right\} \\ &\geq \kappa_2 \epsilon \Delta_{\min,r-1} \\ &\geq \kappa_2 \epsilon \Delta_{i,r}, \end{aligned} \tag{3.17}$$

which ensures, because of (2.64), that

$$\delta m_{r-1} \geq c_4 \Delta_{i,r-1}. \tag{3.18}$$

But (3.15) guarantees that $\beta_{r-1}\Delta_{i,r-1} \leq 1$. We may thus apply Lemma 3.2 and deduce that

$$|\delta m_{i,r-1}| \leq c_2 \|s_{i,r-1}\| \leq c_2 \Delta_{i,r-1} \leq \frac{c_2}{c_4}\delta m_{r-1}, \tag{3.19}$$

where we also used (2.6) and (3.18).

Assume first that $i \in M_{r-1}$, which guarantees that $\delta m_{i,r-1} \neq 0$. Then, using (2.48) and (3.18),

$$|\delta m_{i,r-1}| > \frac{\mu_1}{p}\delta m_{r-1} \geq \frac{\mu_1 c_4}{p}\Delta_{i,r-1}. \tag{3.20}$$

Because of (2.7), (3.1), and (3.20), we therefore obtain that

$$\left|\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} - 1\right| = \frac{|f_i(x_{r-1}+s_{r-1}) - m_{i,r-1}(x_{r-1}+s_{r-1})|}{|\delta m_{i,r-1}|} \leq \frac{c_1 p}{\mu_1 c_4}\beta_{r-1}\Delta_{i,r-1}. \tag{3.21}$$

But (3.14) and (3.15) together give that

$$\beta_{r-1}\Delta_{i,r-1} \leq (1-\eta_3)\frac{\mu_1 c_4^2}{c_1 c_2 p^2}, \tag{3.22}$$

which, with (3.21), implies that

$$\left|\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} - 1\right| \leq \frac{(1-\eta_3)c_4}{c_2 p}. \tag{3.23}$$

Consider first the case where $\delta m_{i,r-1} > 0$. We may then apply (3.19) and deduce that

$$\delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1} = \delta m_{i,r-1}\left(1 - \frac{(1-\eta_3)}{p}\frac{\delta m_{r-1}}{|\delta m_{i,r-1}|}\right) \leq \delta m_{i,r-1}\left(1 - \frac{(1-\eta_3)c_4}{c_2 p}\right). \tag{3.24}$$

Using (3.23), we now deduce that

$$\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} \geq 1 - \frac{(1-\eta_3)c_4}{c_2 p}, \tag{3.25}$$

and therefore, because of (3.24), that

$$\delta f_{i,r-1} \geq \delta m_{i,r-1}\left(1 - \frac{(1-\eta_3)c_4}{c_2 p}\right) \geq \delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1}, \tag{3.26}$$

which implies that (2.50) holds for element i at iteration $r-1$. Now turn to the case where $\delta m_{i,r-1} < 0$. Because of (3.19), we deduce that

$$\delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1} = \delta m_{i,r-1}\left(1 + \frac{(1-\eta_3)}{p}\frac{\delta m_{r-1}}{|\delta m_{i,r-1}|}\right) \leq \delta m_{i,r-1}\left(1 + \frac{(1-\eta_3)c_4}{c_2 p}\right). \tag{3.27}$$

As above, we use (3.23) to obtain that

$$\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} \leq 1 + \frac{(1-\eta_3)c_4}{c_2 p}, \tag{3.28}$$

and therefore, because of (3.27), that

$$\delta f_{i,r-1} \geq \delta m_{i,r-1}\left(1 + \frac{(1-\eta_3)c_4}{c_2 p}\right) \geq \delta m_{i,r-1} - \frac{1-\eta_3}{p}\delta m_{r-1}, \tag{3.29}$$

which again implies that (2.50) holds for element i at iteration $r-1$.

Assume now that $i \in N_{r-1}$. Then, because of (2.7), (2.48), and (3.1), we have that

$$\begin{aligned} |\delta f_{i,r-1}| &\leq |\delta m_{i,r-1}| + |f_i(x_{r-1}+s_{r-1}) - m_{i,r-1}(x_{r-1}+s_{r-1})| \\ &\leq \frac{\mu_1}{p}\delta m_{r-1} + c_1\beta_{r-1}\Delta_{i,r-1}^2. \end{aligned} \tag{3.30}$$

Now, multiplying (3.18) by $\Delta_{i,r-1}$, we obtain that

$$\Delta_{i,r-1}^2 \leq \frac{\Delta_{i,r-1}}{c_4}\delta m_{r-1}. \tag{3.31}$$

Combining (3.30) and (3.31), we deduce that

$$|\delta f_{i,r-1}| \leq \left(\frac{\mu_1}{p} + \frac{c_1}{c_4}\beta_{r-1}\Delta_{i,r-1}\right)\delta m_{r-1}. \tag{3.32}$$

Observing now that (3.14) and (3.15) imply that

$$\beta_{r-1}\Delta_{i,r-1} \leq \frac{c_4(\mu_2 - \mu_1)}{c_1 p}, \tag{3.33}$$

we obtain from (3.32) that

$$|\delta f_{i,r-1}| \leq \frac{\mu_2}{p}\delta m_{r-1}. \tag{3.34}$$

But this inequality implies that (2.56) holds for element i at iteration $r-1$. Thus either (2.50) or (2.56) holds for element i at iteration $r-1$ and the mechanism of the algorithm then implies that $\Delta_{i,r} \geq \Delta_{i,r-1}$. But we may deduce from this inequality that

$$\beta_{r-1}\Delta_{\min,r-1} \leq \beta_{r-1}\Delta_{i,r-1} \leq \beta_r \Delta_{i,r}, \tag{3.35}$$

which contradicts the assumption that r is the smallest iteration number such that (3.14) holds. The inequality (3.14) therefore never holds and we obtain that (3.12) is satisfied for all k. □

We now turn to one of the main results in this section, which proves a weak form of global convergence. The technique is inspired by [35].

THEOREM 3.4. *Assume that* AS.1–AS.6 *hold. Consider a sequence* $\{x_k\}$ *of iterates generated by the algorithm. Then*

$$\liminf_{k\to\infty} \alpha_k = 0. \tag{3.36}$$

Proof. Assume, for the purpose of obtaining a contradiction, that there exists an $\epsilon \in (0,1)$ such that (3.11) holds for all $k \geq 0$. Then

$$\begin{aligned}
\textstyle\sum_{k\in\mathcal{S}} \delta f_k &\geq \eta_1 \textstyle\sum_{k\in\mathcal{S}} \delta m_k \\
&\geq \eta_1\kappa_2\epsilon \textstyle\sum_{k\in\mathcal{S}} \min\left\{\dfrac{\epsilon}{\beta_k}, \max[\Delta_{\min,k}, \|s_k\|], 1\right\} \\
&\geq \eta_1\kappa_2\epsilon \textstyle\sum_{k\in\mathcal{S}} \min\left\{\dfrac{\epsilon}{\beta_k}, \Delta_{\min,k}\right\} \\
&\geq \eta_1\kappa_2\epsilon \min\{\epsilon, c_3\} \displaystyle\sum_{k\in\mathcal{S}} \frac{1}{\beta_k},
\end{aligned} \tag{3.37}$$

where we used successively (2.43), (2.36), (3.11), and Lemma 3.3. We note that (3.37) and AS.2 then imply that

$$\sum_{k\in\mathcal{S}} \frac{1}{\beta_k} < +\infty. \tag{3.38}$$

Now let r be an integer such that

$$\gamma_3\gamma_2^{(r-1)/p} < 1 \tag{3.39}$$

and define

$$\mathcal{S}(k) \stackrel{\text{def}}{=} |\mathcal{S} \cap \{0,\ldots,k-1\}|, \tag{3.40}$$

the number of successful iterations up to iteration $k-1$ ($k \geq 1$). Then define

$$\mathcal{F}_1 \stackrel{\text{def}}{=} \{k \geq 1 \mid k \leq r\mathcal{S}(k)\} \text{ and } \mathcal{F}_2 \stackrel{\text{def}}{=} \{k \geq 1 \mid k > r\mathcal{S}(k)\}. \tag{3.41}$$

We now wish to show that both sums

$$\sum_{k\in\mathcal{F}_1} \frac{1}{\beta_k} \text{ and } \sum_{k\in\mathcal{F}_2} \frac{1}{\beta_k} \tag{3.42}$$

are finite. Consider the first. If it has only finitely many terms, its convergence is obvious. Otherwise, we may assume that $\mathcal{F}_1$ has an infinite number of elements,

and we then construct two subsequences. The first consists of the indices of $\mathcal{F}_1$ in ascending order and the second, $\mathcal{F}_3$ say, consists of the set of indices in $\mathcal{S}$ (in ascending order) with each index repeated r times. Hence the jth element of $\mathcal{F}_3$ is no greater than the jth element of $\mathcal{F}_1$. This gives that

$$\sum_{k\in\mathcal{F}_1}\frac{1}{\beta_k}\leq\sum_{k\in\mathcal{F}_3}\frac{1}{\beta_k}=r\sum_{k\in\mathcal{S}}\frac{1}{\beta_k}\leq+\infty \tag{3.43}$$

because of the nondecreasing nature of the sequence $\{\beta_k\}$ and (3.38). Now turn to the second sum in (3.42). Lemma 2.2 and the mechanism of the algorithm imply that, at each unsuccessful iteration, at least one element trust region radius satisfies (2.55) or (2.59) and none of them is allowed to increase. Hence

$$\prod_{i=1}^{p}\Delta_{i,k}\leq\gamma_3^{p\mathcal{S}(k)}\gamma_2^{k-\mathcal{S}(k)}\prod_{i=1}^{p}\Delta_{i,0}, \tag{3.44}$$

which immediately implies that

$$\Delta_{\min,k}\leq\gamma_3^{\mathcal{S}(k)}\gamma_2^{(k-\mathcal{S}(k))/p}\Delta_{\max,0}, \tag{3.45}$$

where $\Delta_{\max,0}\stackrel{\text{def}}{=}\max_{i\in\{1,\ldots,p\}}\Delta_{i,0}$. We deduce from this inequality that, for $k\in\mathcal{F}_2$,

$$\begin{aligned}\frac{c_3}{\beta_k}\leq\Delta_{\min,k}&\leq\gamma_3^{\mathcal{S}(k)}\gamma_2^{(k-\mathcal{S}(k))/p}\Delta_{\max,0}\leq\gamma_3^{k/r}\gamma_2^{(k-k/r)/p}\Delta_{\max,0}\\&\leq\left[\gamma_3\gamma_2^{(r-1)/p}\right]^{k/r}\Delta_{\max,0},\end{aligned} \tag{3.46}$$

where we have also used Lemma 3.3 and the definition of $\mathcal{F}_2$ in (3.41). Using (3.39), this gives that

$$\sum_{k\in\mathcal{F}_2}\frac{1}{\beta_k}\leq\frac{\Delta_{\max,0}}{c_3}\sum_{k\in\mathcal{F}_2}\left[\gamma_3\gamma_2^{(r-1)/p}\right]^{k/r}<+\infty, \tag{3.47}$$

and the second sum is convergent. Therefore the sum

$$\sum_{k=1}^{\infty}\frac{1}{\beta_k}=\sum_{k\in\mathcal{F}_1}\frac{1}{\beta_k}+\sum_{k\in\mathcal{F}_2}\frac{1}{\beta_k} \tag{3.48}$$

is finite, which contradicts AS.5. Hence condition (3.11) is impossible and (3.36) follows. □

Notice that the relation between α_k, the criticality measure for problem (2.23), and $\alpha(x_k,f,X)$, the criticality measure for problem (2.1), has been left rather unspecified up to this point. It is indeed remarkable that we can prove Theorem 3.4 assuming so little on α. To derive convergence properties for the original problem from Theorem 3.4, we have to be slightly more specific and request that, if both function and model have the same first-order information, then the criticality measures on the original problem and on the model problem agree.

AS.7. Let h_1 and h_2 be two continuously differentiable functions in the intersection of X with a neighbourhood of the feasible point x such that $h_1(x)=h_2(x)$. Then, the difference $\alpha(x,h_1,X)-\alpha(x,h_2,X)$ tends to zero when $\nabla h_1(x)-\nabla h_2(x)$ tends to zero.

In other words, we require the criticality measure to be continuous (near zero) in the *gradient* of its second argument. Again, this is true for the choices (2.24)–(2.25) and (2.28).

With this additional assumption, we are now ready to examine the criticality of the limit points of the sequence of iterates generated by the algorithm for the original problem (2.1).

COROLLARY 3.5. *Assume that* AS.1–AS.7 *hold. Consider a sequence* $\{x_k\}$ *of iterates generated by the algorithm and assume that*

$$\lim_{k\to\infty} \|e_{i,k}\| = 0 \tag{3.49}$$

for all $i \in \{1,\ldots,p\}$. *Then this sequence has at least one critical limit point* x_*.

Proof. From AS.7 and (3.49), we obtain that

$$\lim_{k\to\infty} [\alpha(x_k, f, X) - \alpha_k] = 0, \tag{3.50}$$

which, with (3.36), guarantees

$$\liminf_{k\to\infty} \alpha(x_k, f, X) = 0. \tag{3.51}$$

The desired conclusion then follows by taking a subsequence of $\{x_k\}$ if necessary. □

Condition (3.49) is important, otherwise the situation might arise that an iterate is critical for the current overall model (because its gradient is inexact) while not being critical for the original problem. There are various ways in which (3.49) can be achieved in a practical algorithm, the simplest being to make the size of $e_{i,k}$ also depend on α_k itself, ensuring that the first goes to zero if the latter does.

COROLLARY 3.6. *Assume that* AS.1–AS.7 *hold. If* $\mathcal{S}$, *the set of successful iterations generated by the algorithm, is finite, then all iterates* x_k *are equal to some* x_* *for* k *large enough, and* x_* *is critical.*

Proof. Assume indeed that $\mathcal{S}$ is finite. It is then clear from (2.45) that x_k is unchanged for k large enough, and therefore that $x_* = x_{j+1}$, where j is the largest index in $\mathcal{S}$. Note now that Lemma 2.2 implies that, if $k \notin \mathcal{S}$, then (2.53) or (2.56) must be violated for at least one element. Hence we obtain that $\Delta_{\min,k}$ converges to zero. But (2.8) then implies that $e_{i,k}$ also converges to zero for all $i \in \{1,\ldots,p\}$ and g_k converges to $\nabla f(x_k)$. Thus AS.7 and Corollary 3.5 then guarantee the criticality of x_*. □

As in existing theories for the unstructured trust region case, it is possible to replace the limit inferior in (3.36) by a true limit, therefore ensuring (if the gradients are asymptotically exact) that *all* limit points are critical. As in these theories, a slight strengthening of our assumptions is however necessary.

AS.8. We assume that

$$\lim_{k\to\infty} \beta_k \delta f_k = 0. \tag{3.52}$$

This assumption is similar to that used in [14] and [41], where it is motivated in detail. We only mention here that (3.52) holds for Newton's method on bounded domains because β_k is bounded above in that case.

With this additional assumption, we are now able to replace the limit inferior by a true limit.

THEOREM 3.7. *Assume that* AS.1–AS.8 *hold. Consider the sequence* $\{x_k\}$ *of iterates generated by the algorithm and assume that there are infinitely many successful iterations. Then*

$$\lim_{k\in\mathcal{S}} \alpha_k = 0, \tag{3.53}$$

where $\mathcal{S}$ *is, as above, the set of successful iterations.*

Proof. We again proceed by contradiction. Assume therefore that there exists an $\epsilon_1 \in (0,1)$ and a subsequence $\{q_j\}$ of successful iterates such that, for all q_j in this subsequence,

$$\alpha_{q_j} \geq \epsilon_1. \tag{3.54}$$

Theorem 3.4 guarantees the existence of another subsequence $\{l_j\}$ such that

$$\alpha_k \geq \epsilon_2 \text{ for } q_j \leq k < l_j \text{ and } \alpha_{l_j} < \epsilon_2, \tag{3.55}$$

where we have chosen $\epsilon_2 \in (0, \epsilon_1)$. We may now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \stackrel{\text{def}}{=} \{k \mid k \in \mathcal{S} \text{ and } q_j \leq k < l_j\}, \tag{3.56}$$

where q_j and l_j belong, respectively, to the two subsequences defined above. Applying now (2.36) for $k \in \mathcal{K}$, we obtain from (2.43), (2.16), and $\epsilon_2 < 1$ that

$$\delta f_k \geq \eta_1\kappa_2\epsilon_2 \min\left\{\frac{\epsilon_2}{\beta_k}, \max[\Delta_{\min,k}, \|s_k\|], 1\right\} = \eta_1\kappa_2\epsilon_2 \min\left\{\frac{\epsilon_2}{\beta_k}, \max[\Delta_{\min,k}, \|s_k\|]\right\}. \tag{3.57}$$

But AS.8, along with (3.57), implies that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \beta_k\|s_k\| = 0 \text{ and } \lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \beta_k\Delta_{\min,k} = 0 \tag{3.58}$$

and, because of (2.16), that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \|s_k\| = 0 \text{ and } \lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \Delta_{\min,k} = 0. \tag{3.59}$$

Therefore, we can deduce from (3.57) and (3.58) that, for j sufficiently large,

$$\begin{aligned} \|x_{q_j} - x_{l_j}\| &\leq \textstyle\sum_{k=q_j}^{l_j-1} \|x_{k+1} - x_k\| \\ &= \textstyle\sum_{k=q_j}^{l_j-1} {}^{(\mathcal{K})}\|s_k\| \\ &\leq c_5 \textstyle\sum_{k=q_j}^{l_j-1} {}^{(\mathcal{K})}[f(x_k) - f(x_{k+1})] \\ &\leq c_5[f(x_{q_j}) - f(x_{l_j})], \end{aligned} \tag{3.60}$$

where the sums with superscript $(\mathcal{K})$ are restricted to the indices in $\mathcal{K}$, and

$$c_5 \stackrel{\text{def}}{=} \frac{1}{\eta_1\kappa_2\epsilon_2}. \tag{3.61}$$

But AS.2 and the decreasing nature of the sequence $\{f(x_k)\}$ imply that the last right-hand side of (3.60) converges to zero as j tends to infinity. Hence the continuity of ∇f and AS.7 give that

$$|\alpha(x_{q_j}, f, X) - \alpha(x_{l_j}, f, X)| \leq \frac{1}{6}(\epsilon_1 - \epsilon_2) \tag{3.62}$$

for j sufficiently large. On the other hand, the second part of (3.59) and (2.8) imply that g_{q_j} is arbitrarily close to $\nabla f(x_{q_j})$ when j is large enough, and AS.7 hence guarantees that

$$|\alpha_{q_j} - \alpha(x_{q_j}, f, X)| \leq \frac{1}{6}(\epsilon_1 - \epsilon_2) \tag{3.63}$$

for j sufficiently large. We note also that, because of (2.8),

$$\|g_{l_j} - \nabla f(x_{l_j})\| \leq \sum_{i=1}^{p} \|e_{i,l_j}\| \leq \kappa_1 p \Delta_{\min,l_j}. \tag{3.64}$$

But the mechanism of the algorithm guarantees that no $\Delta_{i,k}$ can increase between iterations $k_j + 1$ and l_j (assuming $k_j + 1 \neq l_j$), where k_j is the largest integer in $\mathcal{K}$ that is smaller than l_j. This yields that

$$\|g_{l_j} - \nabla f(x_{l_j})\| \leq \kappa_1 \gamma_3 p \Delta_{\min,k_j}. \tag{3.65}$$

We now deduce from the second part of (3.59) that the left-hand side of (3.65) tends to zero when j tends to infinity and therefore that, for j sufficiently large,

$$|\alpha_{l_j} - \alpha(x_{l_j}, f, X)| \leq \frac{1}{6}(\epsilon_1 - \epsilon_2) \tag{3.66}$$

because of AS.7. Combining (3.62), (3.63), and (3.66), we obtain, using (3.55), that

$$\alpha_{q_j} \leq \alpha_{l_j} + \frac{1}{2}(\epsilon_1 - \epsilon_2) \leq \frac{1}{2}(\epsilon_1 + \epsilon_2) < \epsilon_1, \tag{3.67}$$

which is impossible because of (3.54). Hence our initial assumption cannot hold and the theorem is proved. □

As above, we now consider the case where we impose that the element gradients are asymptotically exact.

COROLLARY 3.8. *Assume that* AS.1–AS.8 *hold. Consider the sequence* $\{x_k\}$ *of iterates generated by the algorithm and assume furthermore that* (3.49) *holds for all* $i \in \{1, \ldots, p\}$. *Then all limit points of this sequence are critical.*

Proof. If the set $\mathcal{S}$ is finite, the conclusion immediately follows from Corollary 3.6. If, on the other hand, $\mathcal{S}$ has an infinite number of elements, (3.49) implies that g_k is arbitrarily close to $\nabla f(x_k)$ and the combination of AS.7 and Theorem 3.7 ensures the criticality of any limit point of the sequence of successful iterates. □

Of course, (3.49) might be impossible to achieve in practice, and one might consider the case where we can only assert that

$$\limsup_{k \to \infty} \left[\max_{i \in \{1,\ldots,p\}} \|e_{i,k}\| \right] = \kappa_3 \tag{3.68}$$

for some small constant $\kappa_3 > 0$. This is the case, for instance, if gradients are approximated by finite differences.

COROLLARY 3.9. *Assume that* AS.1–AS.6 *and* AS.8 *hold. Consider the sequence* $\{x_k\}$ *of iterates generated by the algorithm. Assume furthermore that* (3.68) *holds and that, for some constant* $L_\alpha > 0$, *the criticality measure* α *satisfies*

$$|\alpha(x, h_1, X) - \alpha(x, h_2, X)| \leq L_\alpha \|\nabla h_1(x) - \nabla h_2(x)\| \tag{3.69}$$

for all $x \in X$ and all functions h_1 and h_2 continuously differentiable in a neighbourhood of x such that $h_1(x) = h_2(x)$. Then, for each limit point x_* of the sequence,

$$\alpha(x_*, f, X) \leq \kappa_3 p L_\alpha. \tag{3.70}$$

Proof. As in Corollary 3.8, the desired conclusion immediately follows from Corollary 3.6 if $\mathcal{S}$ is finite. Assume therefore that $\mathcal{S}$ has infinitely many elements. We then deduce that, for all $k \in \mathcal{S}$,

$$\begin{aligned} \alpha(x_k, f, X) &\leq \alpha_k + |\alpha(x_k, m_k, X) - \alpha(x_k, f, X)| \\ &\leq \alpha_k + L_\alpha \|g_k - \nabla f(x_k)\| \\ &\leq \alpha_k + L_\alpha p \max_{i \in \{1,\ldots,p\}} \|e_{i,k}\|. \end{aligned} \tag{3.71}$$

Taking the limit for k tending to infinity in $\mathcal{S}$ and using Theorem 3.7 and (3.68) then gives the desired conclusion. □

Finally observe that although (3.69) is stronger than AS.7, it is not a very strong condition. For instance, it is satisfied with $L_\alpha = 1$ for the choices (2.24) and also for (2.25) and (2.26) because of the nonexpansive character of the projection operator P_X (see [41], for example). The same property also holds for the choice (2.28), as discussed in [14].

4. Finite identification of the correct active set. When applied to constrained problems, trust region algorithms typically use the notion of projected gradient or projected gradient path to identify a subset of inequality constraints that are satisfied as equalities. Ultimately, the aim thereby is to identify the constraints satisfied as equalities at the solution well before the solution is reached. The methods then reduce to an unconstrained calculation in the manifold defined by the currently "active" constraints. As a consequence, it is possible to guarantee fast asymptotic rates of convergence when using accurate models, as is the case when analytical second-order information of the objective and constraint functions is available.

It is possible to show that structured trust regions do not upset the theory developed in the unstructured case: it can indeed be shown that the constraints active at a particular limit point of the sequence of iterates are identified after a finite number of iterations, provided the normals of the active constraints are linearly independent and strict complementarity holds and provided the step s_{k+1} satisfies the inequality

$$\|s_{k+1}\|_{(k+1)} \geq \gamma_0 \|s_k\|_{(k)} \tag{4.1}$$

for each $k \notin \mathcal{S}$ and for some constant $\gamma_0 \in (0, \gamma_1]$. This latter condition is meant to avoid a situation where the successful iterates converge to a critical point while a subsequence of unsuccessful iterates converges to another point with a different active set. It does not constitute a severe restriction in the step selection procedure and is automatically verified if s_k is determined by a succession of steps of increasing norm such that they remain feasible, within the trust region B_k, and ensure (2.36). This is the case, for instance, if truncated conjugate gradients are used for computing the step in the solution of an unconstrained problem (see [37] or [38]).

The theory considers the active constraint identification problem from a quite general point of view. The main observation is that a number of the existing theories for active constraint identification are based on the definition of a special criticality measure that satisfies AS.6 while not satisfying AS.7 (see [2] or [3], for instance). Let us denote this measure at iteration k by $\bar{\alpha}_k$. The steps leading to constraint identification are then as follows.

1. The first step is to prove that a sufficient decrease condition of the type (2.33) also holds with $\bar{\alpha}_k$ instead of α_k.
2. One then proceeds to prove that

$$\liminf_{k\to\infty} \bar{\alpha}_k = 0 \tag{4.2}$$

much in the same way as for (3.36).
3. The measure $\bar{\alpha}_k$ is also constructed to ensure that it is asymptotically bounded away from zero for all points such that their active set is not identical to that of a (close) critical point. (This, in particular, prevents AS.7 from holding.)
4. Some contradiction is then deduced from these last two properties.

However, since this development is rather technical and lengthy, we do not include it in the present paper, but refer the interested reader to [15] for details of the results and additional assumptions. This reference also contains the theory concerning the convergence of the iterates to a single limit point, adapted from [14].

Our experience with the solution of practical problems however indicates that the identification of active constraints is seldom observed in practice before the very last iterations of the algorithm, which makes the results discussed in this section mainly of theoretical interest.

5. Extensions. We examine in this section some extensions and variants of the results presented above.

5.1. A hybrid technique. One of the possible drawbacks of the algorithm of §2.3 is that steps might be constrained to be unnecessarily small in directions corresponding to highly nonlinear element functions. Indeed, the negative effect of inaccurate models for these elements might be compensated by a successful step in directions corresponding to less nonlinear elements. This compromise between the different parts of the objective is, of course, inherent to the classical method using an unstructured trust region.

We might try to obtain the best of both classical and structured approaches by using a hydrid technique. In this technique, a *global* trust region radius Δ_k is recurred for the objective function considered as a single element (using the algorithm analyzed above, which is then equivalent to the classical one), along with the individual radii $\Delta_{i,k}$. We then define the individual "hybrid" radii by

$$\Delta^h_{i,k} \stackrel{\text{def}}{=} \max\{\Delta_k, \Delta_{i,k}\} \tag{5.1}$$

for each $i \in \{1, \ldots, p\}$ and redefine $B_{i,k}$ as

$$B_{i,k} \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\| \leq \Delta^h_{i,k}\}. \tag{5.2}$$

We can then apply our algorithm with these new quantities, to the effect that well-modelled elements have their associated trust regions possibly extended without having to contract those corresponding to badly modelled ones, as long as the global agreement is satisfactory.

It is not difficult to verify that the theory presented above still holds for this hybrid modification. The key points are to observe that the revised definition of our trust region implies that

$$\delta m_k \geq \kappa_2 \alpha_k \min\left\{\frac{\alpha_k}{\beta_k}, \Delta_k, 1\right\}, \tag{5.3}$$

which is the classical sufficient decrease condition (2.33), that the inequalities (2.64) are still valid with $\Delta_{i,k}$ replaced by $\Delta_{i,k}^h$, and also that an analogous result to Lemma 3.3 also holds for the global trust region radius, as is already well known from the unstructured trust region case (see [14], for instance).

5.2. An alternative definition of success. An immediate consequence of inequality (2.63) in Lemma 2.1 is that it would be possible to replace the condition (2.43) for an iteration to be successful by

$$\delta f_k \geq \eta_1 \sum_{i \in M_k} \delta m_{i,k}(x_k) \tag{5.4}$$

without altering the developments presented above. Indeed, (2.63) shows the equivalence between (2.43) and (5.4). We have chosen to use (2.43) above, because it seems natural to consider the same collection of elements on both sides of the inequality.

5.3. Weaker sufficient decrease conditions. It is remarkable to note that Lemma 3.3 and Theorem 3.4 can be proved in a weaker context. Indeed, we could require the weaker sufficient decrease condition

$$\delta m_k \geq \kappa_2 \alpha_k \min \left\{ \frac{\alpha_k}{\beta_k}, \Delta_{\min,k}, 1 \right\} \tag{5.5}$$

instead of (2.36) and still prove Lemma 3.3 and Theorem 3.4. However, we have not been able to prove Theorem 3.7, nor active constraint identification, with these assumptions because (5.5) involves only the length of the step in a possibly small subspace of $\mathbf{R}^n$.

5.4. Using uniformly equivalent norms. Another possible generalization of the theory developed above allows the use of different norms for each element and for each iteration. Let us denote these norms by the symbol $\| \cdot \|_{(i,k)}$. The element trust region definition (2.6) then becomes

$$B_{i,k} \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\|_{(i,k)} \leq \Delta_{i,k}\}, \tag{5.6}$$

while the gradient approximation condition (2.8) may be written as

$$\|e_{i,k}\|_{[i,k]} \leq \kappa_1 \Delta_{\min,k}, \tag{5.7}$$

where the norm $\| \cdot \|_{[i,k]}$ is any norm that satisfies

$$|\langle x, y \rangle| \leq \|x\|_{(i,k)} \|y\|_{[i,k]} \tag{5.8}$$

for all $x, y \in \mathbf{R}^n$. In particular, one can choose the *dual norm* of $\| \cdot \|_{(i,k)}$ defined by

$$\|y\|_{[i,k]} \stackrel{\text{def}}{=} \sup_{x \neq 0} \frac{|\langle x, y \rangle|}{\|x\|_{(i,k)}}. \tag{5.9}$$

With iteration k, we may also associate an overall norm $\| \cdot \|_{(k)}$ defined on the whole of $\mathbf{R}^n$, whose purpose is to reflect the relative weighting of the different elemental norms $\| \cdot \|_{(i,k)}$ in a global measure.

If we assume that all the considered norms are *uniformly equivalent*, that is if there exists a constant $\sigma \geq 1$ such that, for all x,

$$\frac{1}{\sigma} \|x\|_b \leq \|x\|_a \leq \sigma \|x\|_b, \tag{5.10}$$

where $\|\cdot\|_a$, $\|\cdot\|_b$ is any pair of the above-defined norms, then the theory developed in all the preceding sections is still valid without any substantial modification. Again the details of the proofs in this more general setting are provided in [15]. Note that this extension covers the possible introduction of iteration-dependent scaling in a practical implementation of our algorithm, which can be highly desirable for some difficult problems.

6. Conclusions. We have shown in this paper that the trust region concept, one of the most powerful tools for building efficient and robust algorithms for optimization, can be extended in a very natural way to reflect the structure of the underlying problem. The algorithm proposed is indeed a direct generalization of the more usual case where only an unstructured uniform trust region is considered. Similar global convergence properties can be proved for the new algorithm, including the case where dynamic scaling is performed on the variables and the situation where the gradients are only known approximately.

We must wait to see whether this modification of a trust region algorithm will prove efficient in practice and justify the slight additional complexity of the method. Note that the results of preliminary numerical experiments (based on a modification of LANCELOT using the implementation described after the algorithm) have been encouraging. Tests on unconstrained problems from the CUTE collection [1] have shown that the new method, although very comparable to LANCELOT in many cases, sometimes produces substantial improvements. However, we anticipate the real power of the concept to appear when minimizing augmented Lagrangians or other penalty-like functions, because scaling is much more critical there than in many of the classical unconstrained test examples. The authors are planning to include the new technique described in this paper within the next release of LANCELOT.

One of the nice features of the partially separable functions considered in the present theory is that the objective is a *linear* combination of its elements. While group partial separability, as used in [12] or [13], has computational advantages in terms of economy of derivative calculation, this structure involves a *nonlinear* relationship between the elements and the overall function. This seems to make exploiting the link between local and global models much harder. While we would be interested in deriving structured trust region methods for group partially separable functions, the methods would undoubtedly be more complicated and less amenable to analysis. Thus, we are content, in the present paper, to consider the simpler, but nonetheless very general, partially separable structure.

Finally, there might be other ways to introduce structure in trust region methods than to consider (group) partially separable objective functions. In particular, trust region methods for nonlinearly constrained problems seem attractive candidates for an alternative approach that would separate the trust region(s) on the objective from those on the constraints.

Acknowledgments. The authors are indebted to Johara Shahabuddin for twice pointing out an unsuitable definition of the sufficient decrease condition (2.36) in §2.2.2.

REFERENCES

[1] I. BONGARTZ, A. R. CONN, N.I.M. GOULD, AND PH. L. TOINT, CUTE: *Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[2] J. V. BURKE AND J. J. MORÉ, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[3] J. V. BURKE, J. J. MORÉ, AND G. TORALDO, *Convergence properties of trust region methods for linear and convex constraints*, Math. Programming, Ser. A, 47 (1990), pp. 305–336.

[4] R. H. BYRD, E. ESKOW, R. B. SCHNABEL, AND S. L. SMITH, *Parallel Global Optimization: Numerical Methods, Dynamic Scheduling Methods, and Application to Molecular Configuration*, Tech. Report CU-CS-553-91, Department of Computer Science, University of Colorado at Boulder, Boulder, CO, 1991.

[5] R. H. BYRD, R. B. SCHNABEL, AND G. A. SCHULTZ, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[6] P. H. CALAMAI AND J. J. MORÉ, *Projected gradient methods for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[7] R. G. CARTER, *On the global convergence of trust region methods using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.

[8] M. R. CELIS, J. E. DENNIS, AND R. A. TAPIA, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1985, pp. 71–82.

[9] A. R. CONN, N. I. M. GOULD, M. LESCRENIER, AND PH. L. TOINT, *Performance of a multifrontal scheme for partially separable optimization*, in Advances in Optimization and Numerical Analysis, Proceedings of the 6th Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico, Vol. 275, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1994, pp. 79–96.

[10] A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460. See also 26 (1989), pp. 764–767.

[11] ———, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comput., 50 (1988), pp. 399–430.

[12] ———, *An introduction to the structure of large scale nonlinear optimization problems and the* LANCELOT *project*, in Computing Methods in Applied Sciences and Engineering, R. Glowinski and A. Lichnewsky, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 42–51.

[13] ———, LANCELOT: *A Fortran package for large-scale nonlinear optimization (Release A)*, in Springer Series in Computational Mathematics, No. 17, Springer-Verlag, Heidelberg, Berlin, New York, 1992.

[14] A. R. CONN, NICK GOULD, A. SARTENAER, AND PH. L. TOINT, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[15] ———, *Convergence Properties of Minimization Algorithm for Convex Constraints Using a Structured Trust Region (revised)*, Technical Report 94/5, Department of Mathematics, FUNDP, Namur, Belgium, 1994.

[16] M. EL-ALEM, *A global convergence theory for the Dennis–Celis–Tapia trust-region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[17] R. FLETCHER, *Practical Methods of Optimization: Unconstrained Optimization*, J. Wiley and Sons, New York, 1980.

[18] D. M. GAY, *Is exploiting partial separability useful?* Talk at the 6th Mexico United States Workshop, Oaxaca, Mexico, January 1992.

[19] D. GOLDFARB AND S. WANG, *Partial-update Newton methods for unary, factorable, and partially separable optimization*, SIAM J. Optim., 3 (1993), pp. 383–397.

[20] A. GRIEWANK, *The global convergence of partitioned BFGS on problems with convex decompositions and Lipschitzian gradients*, Math. Programming, Ser. A, 50 (1991).

[21] A. GRIEWANK AND PH. L. TOINT, *On the unconstrained optimization of partially separable functions*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, London, New York, 1982, pp. 301–312.

[22] A. GRIEWANK AND PH. L. TOINT, *Numerical experiments with partially separable optimization problems*, in Numerical Analysis: Proceedings Dundee 1983, Lecture Notes in Mathematics, Vol. 1066, Springer-Verlag, Berlin, 1984, pp. 203–220.

[23] ———, *On the existence of convex decomposition of partially separable functions*, Math. Programming, 28 (1984), pp. 25–49.

[24] W. A. GRUVER AND E. SACHS, *Algorithmic Methods in Optimal Control*, Pitman, Boston, 1980.

[25] M. D. HEBDEN, *An Algorithm for Minimization Using Exact Second Derivatives*, Technical Report T.P. 515, AERE Harwell Laboratory, Harwell, UK, 1973.

[26] M. LESCRENIER, *Partially separable optimization and parallel computing*, in Parallel Optimization on Novel Computer Architectures, R. R. Meyer and S. Zenios, eds., A. C. Baltzer Scientific Publishing Co., Switzerland, 1989.

[27] ———, *Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold*, SIAM J. Numer. Anal., 28 (1991), pp. 476–495.

[28] K. LEVENBERG, *A method for the solution of certain problems in least squares*, Quarterly J. Appl. Math., 2 (1944), pp. 164–168.

[29] D. MARQUARDT, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM J. Appl. Math., 11 (1963), pp. 431–441.

[30] J. J. MORÉ, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Proceedings Dundee 1977, Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1978.

[31] ———, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[32] ———, *Trust regions and projected gradients*, in System Modelling and Optimization, Lecture Notes in Control and Information Sciences, Vol. 113, Springer-Verlag, Berlin, 1988, pp. 1–13.

[33] J. J. MORÉ AND G. TORALDO, *On the solution of large scale quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[34] M. J. D. POWELL, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970.

[35] ———, *On the global convergence of trust region algorithms for unconstrained optimization*, Math. Programming, 29 (1984), pp. 297–303.

[36] M. J. D. POWELL AND Y. YUAN, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1990), pp. 189–213.

[37] T. STEIHAUG, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[38] PH. L. TOINT, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, London, 1981, pp. 57–88,

[39] ———, *Global convergence of the partitioned BFGS algorithm for convex partially separable optimization*, Math. Programming, 36 (1986), pp. 290–307.

[40] ———, *On large scale nonlinear least squares calculations*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 416–435.

[41] ———, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[42] PH. L. TOINT AND D. TUYTTENS, *On large scale nonlinear network optimization*, Math. Programming, Ser. B, 48 (1990), pp. 125–159.

[43] ———, *LSNNO: A Fortran subroutine for solving large scale nonlinear network optimization problems*, ACM Trans. Math. Software, 18 (1992), pp. 308–328.

[44] A. VARDI, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1087–1105, November 1996
© 1996 Society for Industrial and Applied Mathematics
011

CHARACTERIZATIONS OF STRONG REGULARITY FOR VARIATIONAL INEQUALITIES OVER POLYHEDRAL CONVEX SETS*

A. L. DONTCHEV† AND R. T. ROCKAFELLAR‡

Abstract. Linear and nonlinear variational inequality problems over a polyhedral convex set are analyzed parametrically. Robinson's notion of strong regularity, as a criterion for the solution set to be a singleton depending Lipschitz continuously on the parameters, is characterized in terms of a new "critical face" condition and in other ways. The consequences for complementarity problems are worked out as a special case. Application is also made to standard nonlinear programming problems with parameters that include the canonical perturbations. In that framework a new characterization of strong regularity is obtained for the variational inequality associated with the Karush–Kuhn–Tucker conditions.

Key words. variational inequality, Aubin property, strong regularity, Lipschitz stability in optimization

AMS subject classifications. 49K40, 90C31, 49J40

1. Introduction.

For a map $f : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^n$ and a nonempty, polyhedral, convex set $C \subset \mathbb{R}^n$, we study the variational inequality problem in which a point x is sought such that

$$x \in C \text{ and } \langle z + f(w,x),\, x' - x \rangle \geq 0 \text{ for all } x' \in C.$$

(Here $\langle \cdot, \cdot \rangle$ refers to the scalar product of two vectors.) This problem is viewed as depending on $w \in \mathbb{R}^d$ and $z \in \mathbb{R}^n$ as parameter vectors (with z representing the "canonical perturbations"); we put them together as $p = (z, w)$. In terms of the normal cone $N_C(x)$ to C at x in convex analysis, which is given by

$$N_C(x) = \begin{cases} \{v \in \mathbb{R}^n \mid \langle v,\, x' - x \rangle \leq 0 \text{ for all } x' \in C\} & \text{if } x \in C, \\ \emptyset & \text{if } x \notin C, \end{cases}$$

the targeted variational inequality can be expressed conveniently as

$$z + f(w,x) + N_C(x) \ \ni \ 0. \tag{1}$$

For each $p = (z,w) \in \mathbb{R}^n \times \mathbb{R}^d$ let $S(p)$ be the (possibly empty) set of solutions x of (1). We concern ourselves with the local behavior of the map S around a fixed element $p_0 = (z_0, w_0)$ and a point $x_0 \in S(p_0)$. Specifically we are interested in the circumstances under which S is locally single valued and Lipschitz continuous around (p_0, x_0), in the sense that there exist neighborhoods U of x_0 and V of p_0 such that the map $p \mapsto S(p) \cap U$ is single valued and Lipschitz continuous relative to $p \in V$. In addressing this we assume here that

(A) *f is differentiable with respect to x with Jacobian matrix $\nabla_x f(w,x)$ depending continuously on (w,x) in a neighborhood of (w_0, x_0);*

* Received by the editors April 4, 1995; accepted for publication (in revised form) August 30, 1995. This work was supported by National Science Foundation grant DMS 9404431 for the first author and DMS 9500957 for the second.

† Mathematical Reviews, Ann Arbor, MI 48107 (ald@math.ams.org).

‡ Department of Mathematics, University of Washington, Seattle, WA 98195 (rtr@math.washington.edu).

(B) *f is Lipschitz continuous in w uniformly in x around (w_0, x_0); that is, there exist neighborhoods U of x_0 and V of w_0 and a number $l > 0$ such that $\|f(w_1, x) - f(w_2, x)\| \leq l\|w_1 - w_2\|$ for all $x \in U$ and $w_1, w_2 \in V$.*

It has long been known, thanks to Robinson, that the analysis of S is closely tied to the linear variational inequality

$$q + Ax + N_C(x) \ni 0 \tag{2}$$

with canonical parameter vector q around q_0 in the case of

$$A = \nabla_x f(w_0, x_0), \qquad q_0 = z_0 + f(w_0, x_0) - \nabla_x f(w_0, x_0)x_0, \tag{3}$$

which serves to "linearize" (1) at (p_0, x_0). Let $L(q)$ denote the set of solutions x to (2). Then $x_0 \in L(q_0)$ by (3). In a landmark paper [23], Robinson proved that the solution map S for (1) is locally single valued and Lipschitz continuous around (p_0, x_0) when the solution map L for (2) is locally single valued and Lipschitz continuous around (q_0, x_0). Robinson called this property of L under (3) the *strong regularity* of the variational inequality (1) at (p_0, x_0). (His framework in [23] was somewhat broader than the one adopted here: C did not have to be polyhedral, and w could range over a parameter space other than $\mathbb{R}^d$; for subsequent extensions in such a mode see Robinson [25] and Dontchev and Hager [6].)

The strong regularity of (1) at (p_0, x_0) is identical by definition to the strong regularity of (2) at (q_0, x_0) under the choice of elements in (3). Our goal here is to characterize this strong regularity by a certain *critical face condition* on A and the closed faces of the critical cone K_0 consisting of the vectors in the tangent cone $T_C(x_0)$ to C at x_0 that are orthogonal to the normal vector

$$v_0 = -Ax_0 - q_0 \in N_C(x_0). \tag{4}$$

We further provide characterizations through a localized Lipschitz condition on L at (q_0, x_0) which we call the Aubin property and also through the lower semicontinuity of L around (q_0, x_0).

The result that the lower semicontinuity of L around (q_0, x_0) thereby entails the local single valuedness and Lipschitz continuity of L can be compared with the well-known fact that a monotone map has to be single-valued and continuous wherever it is lower semicontinuous. We would have L monotone if the matrix A in (2) were monotone (i.e., positive semidefinite, not necessarily symmetric), but such monotonicity is not assumed. We do not know whether the lower semicontinuity of S around (p_0, x_0) likewise ensures the local single-valuedness and Lipschitz continuity of S around (p_0, x_0), but we verify that the Aubin property of S at (p_0, x_0) does yield it.

Throughout we denote by $\mathbb{B}_r(x)$ the closed ball centered at x with radius r and by $\mathbb{B}$ the closed unit ball. For a (potentially set-valued) map Γ from $\mathbb{R}^m$ to $\mathbb{R}^n$ we denote by $\operatorname{gph}\Gamma$ the graph of Γ, i.e., the set $\{(u, x) \mid u \in \mathbb{R}^m,\ x \in \Gamma(u)\}$. Recall that Γ is called lower semicontinuous *at* the pair $(u_0, x_0) \in \operatorname{gph}\Gamma$ if for every sequence $u_i \to u_0$ there exists a sequence $x_i \to x_0$ such that $x_i \in \Gamma(u_i)$ for all $i = 1, 2, \ldots$ sufficiently high. If Γ is lower semicontinuous at every point $(u, x) \in \operatorname{gph}\Gamma$ with u belonging to an open set U, it is said to be lower semicontinuous on U. We say that Γ is lower semicontinuous *around* $(u_0, x_0) \in \operatorname{gph}\Gamma$ if there exists a neighborhood W of (u_0, x_0) such that Γ is lower semicontinuous at every $(u, x) \in \operatorname{gph}\Gamma \cap W$. We also employ the following concept of Aubin [1].

DEFINITION 1. *A set-valued map* Γ *from* $\mathbb{R}^m$ *to the subsets of* $\mathbb{R}^n$ *has the* Aubin property *at* $(u_0, x_0) \in \operatorname{gph}\Gamma$ *with a constant* M *if there exist neighborhoods* U *of* u_0 *and* V *of* x_0 *such that*

$$\Gamma(u_1) \cap V \subset \Gamma(u_2) + M\|u_1 - u_2\|\mathbb{B} \text{ for all } u_1, u_2 \in U.$$

Aubin himself referred to this as "pseudo-Lipschitz continuity." In actuality it is a fundamental property more important in general than Lipschitz continuity as usually interpreted with the Hausdorff metric: it readily characterizes the latter when Γ is locally bounded (see Rockafellar [27]), but it makes better sense in most cases when Γ is not locally bounded or in particular has unbounded images $\Gamma(u)$—all of which jars with the connotation of "pseudo" as "false." We prefer therefore to call this concept the Aubin property, giving credit where credit is due. This property of Γ is equivalent to Γ^{-1} having a "linear rate of openness" (hence providing a link to open map theorems) as well as to Γ^{-1} being metrically regular (a basic condition employed in the stability analysis of optimization problems); see Borwein and Zhuang [3] and Penot [21].

When C is the nonnegative orthant $\mathbb{R}^n_+$, the variational inequality (1) corresponds to the *complementarity* problem while (2) gives the *linear complementarity* problem, which seeks an $x \in \mathbb{R}^n$ such that

$$Ax + q \geq 0, \quad x \geq 0, \quad \langle x, Ax + q\rangle = 0. \tag{5}$$

The initial motivation for our efforts came from this case and the results that had been obtained for its solution map L_0, assigning to each q the set of all x that satisfy (5), if any.

Samelson, Thrall and Wesler [29] showed that L_0 is single valued everywhere on $\mathbb{R}^n$ if and only if A is a P-matrix; that is, every principal minor of A has positive sign. Alternative descriptions of P-matrices have been provided in [4] and [19]. Mangasarian and Shiau [16] proved that when L_0 is single valued everywhere it is automatically Lipschitz continuous everywhere as well. Gowda [8] proved that if $L_0(q) = \{0\}$ for some positive q, then the Lipschitz continuity of L_0 everywhere in the sense of the Hausdorff metric guarantees that L_0 is single valued everywhere; see also Pang [20]. Murthy, Parthasarathy, and Sabatini [18] dropped the requirement that $L_0(q) = \{0\}$ for some positive q, obtaining that L_0 is Lipschitz continuous everywhere if and only if it is single valued everywhere. Gowda and Sznajder [9] noted that this result can be extended to the map L by using some recently discovered properties of normal maps. Such results got us interested in investigating also the question of local single valuedness versus local Lipschitz continuity and how these properties could better be understood in relation to each other, and this led to the developments presented here. A product of this study, back on the global level, turns out to be that the lower semicontinuity of L everywhere on $\mathbb{R}^n$ is already enough to guarantee the single valuedness of L everywhere.

We begin in §2 by considering the linear variational inequality (2) and establishing the equivalences that have been mentioned for its solution map L. In obtaining our critical face condition, a key step is the application to L of Mordukhovich's coderivative criterion [17] for the Aubin property to hold. In §3 we return to the nonlinear variational inequality (1), putting the preceding results together and furnishing along the way an independent proof of Robinson's theorem, not based on a fixed point argument but utilizing instead our identification of strong regularity with the Aubin

property, and noting further that strong regularity is not just sufficient but necessary for the local single valuedness and Lipschitz continuity of S when the canonical perturbations z are present along with the general parameter element w.

As applications in §§4 and 5 we characterize strong regularity in the complementarity problem and for the variational inequality representing the first-order optimality conditions in a nonlinear programming problem. In the latter case we demonstrate that the combination of linear independence of the active constraint gradients with the strong second-order sufficient condition for local optimality is necessary as well as sufficient (in the presence of the canonical perturbations) for the Karush–Kuhn–Tucker (KKT) map to be not just locally single valued and Lipschitz continuous but such that its primal components are locally optimal solutions.

2. Strong regularity and the linear problem. As noted, the strong regularity of the nonlinear variational inequality (1) at (p_0, x_0) is the same as the strong regularity of the linear variational inequality (2) at (q_0, x_0) under the choice of elements in (3). Our plan of characterizing the strong regularity property in various ways can therefore be executed entirely in the linear context. Eventually the case of interest will be the one given by (3), but for now (q_0, x_0) can be any pair belonging to the graph of the solution map L for (2).

A key part in our investigation will be played by a lemma which reduces the linear variational inequality (2) over a polyhedral convex set C to a variational inequality over a polyhedral convex *cone* K. To formulate it, we introduce for each $x \in C$ and normal vector $v \in N_C(x)$ the cone

$$K(x,v) = \{x' \in T_C(x) \mid x' \perp v\}, \tag{6}$$

where $T_C(x)$ denotes the tangent cone to C at x. Here $T_C(x)$ is polyhedral convex because C is polyhedral convex, and this ensures that $K(x,v)$ is polyhedral convex too. Our interest will center especially on the *critical cone* associated with (2) for (q_0, x_0), which is

$$K_0 = K(x_0, v_0) \quad \text{for} \quad v_0 = -Ax_0 - q_0. \tag{7}$$

REDUCTION LEMMA. *For any $(x,v) \in G = \operatorname{gph} N_C$ there is a neighborhood U of $(0,0)$ in $\mathbb{R}^n \times \mathbb{R}^n$ such that for $(x',v') \in U$ one has*

$$v + v' \in N_C(x + x') \iff v' \in N_{K(x,v)}(x').$$

In particular, the tangent cone $T_G(x,v)$ to G at (x,v) is $\operatorname{gph} N_{K(x,v)}$.

Proof. This is a particular case of Lemma 3.5 in Robinson [24]; see also Theorem 5.6 in Rockafellar [28]. □

In the following theorem we show that the Aubin property of the map L is equivalent to the strong regularity of the variational inequality (2). The key steps in the proof are the Reduction Lemma and a combination of some recently obtained characterizations of normal maps.

THEOREM 1. *The following are equivalent:*

(i) *L is lower semicontinuous around (q_0, x_0);*

(ii) *L has the Aubin property at (q_0, x_0);*

(iii) *L is locally single valued and Lipschitz continuous around (q_0, x_0);*

(iv) *the linear variational inequality (2) is strongly regular at (q_0, x_0).*

Proof. Obviously, (iv) ⇔ (iii) ⇒ (ii) ⇒ (i). It will suffice therefore to show that (i) ⇒ (iii). For the critical cone K_0 in (7) consider the variational inequality

$$q' + Ax' + N_{K_0}(x') \ni 0 \tag{8}$$

and denote its solution map by L': for each q', $L'(q')$ is the set of all x' satisfying (8). The Reduction Lemma tells us that as long as (x', v') is near enough to $(0,0)$, we have $v_0 + v' \in N_C(x_0 + x')$ if and only if $v' \in N_{K_0}(x')$. Thus, we have $(q_0 + q') + A(x_0 + x') + N_C(x_0 + x') \ni 0$ if and only if $q' + Ax' + N_{K_0}(x') \ni 0$. In the shifted notation $x = x_0 + x'$ and $q = q_0 + q'$, therefore, we have $x \in L(q)$ if and only if $x' \in L'(q')$. In particular $0 \in L'(0)$, and the lower semicontinuity of L around (q_0, x_0) in (i) reduces to that of L' around $(0,0)$. But L' is positively homogeneous by virtue of K_0 being a cone, so the lower semicontinuity of L' around $(0,0)$ implies the lower semicontinuity and nonempty valuedness of L' on all of $\mathbb{R}^n$. Our task comes down to proving that this implies L' is locally single valued and Lipschitz continuous around $(0,0)$ (and hence by positive homogeneity has these properties globally).

Let Π_{K_0} be the projection map onto K_0. We have

$$u - \Pi_{K_0}(u) \in N_{K_0}\big(\Pi_{K_0}(u)\big) \text{ for all } u.$$

Let h be the normal map associated with (8), namely

$$h(u) = \big[u - \Pi_{K_0}(u)\big] + A\Pi_{K_0}(u).$$

As a step toward applying known results of the theory of normal maps, we prove next that h is an open map: it maps open sets into open sets.

For this purpose fix any open set $O \subset \mathbb{R}^n$ and any point $h(u)$ with $u \in O$; it will be expedient to take $q' = -h(u)$. Consider any sequence $q'_i \to q'$ as $i \to \infty$. By demonstrating the existence of a sequence $u_i \to u$ with $q'_i = -h(u_i)$, we will confirm that eventually $-q'_i \in h(O)$ and therefore that $h(O)$ is open. From the definitions of L' and h along with the choice of q' we have for $x' = \Pi_{K_0}(u)$ that $-q' = [u - x'] + Ax'$ with $u - x' \in N_{K_0}(x')$, hence $x' \in L'(q')$ and $x' - q' - Ax' = u$. The nonempty valuedness and lower semicontinuity of L' on $\mathbb{R}^n$ imply the existence of points $x'_i \in L'(q'_i)$ (for i sufficiently large) with $x'_i \to x'$. Since $x'_i \in L'(q'_i)$ we have $-q'_i - Ax'_i \in N_{K_0}(x'_i)$ and consequently for the points $u_i = x'_i - q'_i - Ax'_i$ that $\Pi_{K_0}(u_i) = x'_i$. But $x'_i - q'_i - Ax'_i \to x' - q - Ax'$, so we have $u_i \to u$ as demanded.

The rest of the proof is based on combining two known facts. The first is that a piecewise affine map (here h fits this category because Π_{K_0} is piecewise linear) is open if and only if it is coherently oriented; see Eaves and Rothblum [7, Lem. 6.12] and Scholtes [30, Prop. 2.3.7]. The second fact is that the normal map corresponding to a linear variational inequality over a polyhedral convex set is coherently oriented if and only if it is one-to-one; see Robinson [26, Thm. 4.3], and also Ralph [22]. From these facts we deduce that h^{-1} is single valued and Lipschitz continuous everywhere. The equivalence

$$x' \in L'(q') \iff x' = \Pi_{K_0}(h^{-1}(q'))$$

implies then that L' is single valued and Lipschitz continuous everywhere. Thus we have arrived at (iii), the goal we had set. □

Remark 1. Corresponding to Theorem 1 on the global level is the fact that the following are equivalent:

(i) L is lower semicontinuous on $\mathbb{R}^n$;

(ii) $L(q)$ is a singleton set for every $q \in \mathbb{R}^n$.

This is easily derivable from known literature with a little help from the argument we have used in proving Theorem 1. Assuming (i), denote by h_C the normal map associated with (2). Tracing the argument in the proof of Theorem 1 but with h replaced

by h_C and L' replaced by L and relying on the references cited there, we obtain that h_C is open everywhere and consequently that L is single valued everywhere. Conversely, under (ii) the map h_C is a homeomorphism, hence it is Lipschitz continuous everywhere. Then L is Lipschitz continuous and in particular lower semicontinuous everywhere. □

We proceed now toward our critical face condition. Recall that the closed faces F of any polyhedral convex cone K are the polyhedral convex cones of the form

$$F = \{x \in K \mid x \perp v\} \text{ for some } v \in K^*, \tag{9}$$

where K^* denotes the polar of K. The largest of these faces is K itself, while the smallest is $K \cap (-K)$, this being the maximal subspace of $\mathbb{R}^n$ included within K. Recall too that

$$v' \in N_K(x') \iff x' \in K, \quad v' \in K^*, \quad x' \perp v'. \tag{10}$$

DEFINITION 2 *The* critical face condition *will be said to hold at* (q_0, x_0) *if for all choices of closed faces* F_1 *and* F_2 *of the critical cone* K_0 *with* $F_1 \supset F_2$,

$$u \in F_1 - F_2, \quad A^\top u \in (F_1 - F_2)^* \implies u = 0$$

(where $A^\top$ *denotes the transpose of* A*).*

Remark 2. When the critical cone K_0 happens to be a subspace, it has a unique closed face (namely itself). The critical face condition reduces then to a nonsingularity condition for A relative to this subspace:

$$u \in K_0, \quad A^\top u \perp K_0 \implies u = 0.$$

THEOREM 2. *The solution map* L *for the linear variational inequality* (2) *has the Aubin property at* (q_0, x_0)*, and therefore the other equivalent properties of Theorem* 1 *as well, if and only if the critical face condition holds at* (q_0, x_0).

Proof. According to the powerful criterion developed by Mordukhovich [17], we know that a necessary and sufficient condition for the Aubin property to hold for L at (q_0, x_0) is

$$A^\top u + D^* N_C(x_0|v_0)(u) \ni 0 \implies u = 0,$$

where the map $D^* N_C(x_0|v_0)$ is the coderivative of the map N_C at the point (x_0, v_0) of $G = \operatorname{gph} N_C$. By definition, the graph of this coderivative map consists of all the pairs $(-u, r)$ such that $(r, u) \in \widetilde{N}_G(x_0, v_0)$, where $\widetilde{N}_G(x_0, v_0)$ is the generalized cone of normals to the (nonconvex) set G that is described below. In these terms the Mordukhovich criterion takes the form

$$(A^\top u, u) \in \widetilde{N}_G(x_0, v_0) \implies u = 0. \tag{11}$$

Everything hinges therefore on determining $\widetilde{N}_G(x_0, v_0)$.

In general, $\widetilde{N}_G(x_0, v_0)$ is defined as the "lim sup" of polar cones $T_G(x, v)^*$ as $(x, v) \to (x_0, v_0)$ in G, but because G is the union of finitely many polyhedral sets in $\mathbb{R}^{2n}$ (due to C being polyhedral), only finitely many cones can be manifested as $T_G(x, v)$ at points $(x, v) \in G$ near (x_0, v_0). Thus, we have for any sufficiently small neighborhood U of (x_0, v_0) that

$$\widetilde{N}_G(x_0, v_0) = \bigcup_{(x,v) \in U \cap G} T_G(x, v)^*. \tag{12}$$

Next we utilize the Reduction Lemma: we have $T_G(x,v) = \operatorname{gph} N_{K(x,v)}$, and therefore by (10) as applied to $K = K(x,v)$, that

$$T_G(x,v) = \big\{(x',v') \,\big|\, x' \in K(x,v),\ v' \in K(x,v)^*,\ x' \perp v'\big\}.$$

It follows that

$$\begin{aligned} T_G(x,v)^* &= \big\{(r,u) \,\big|\, \langle (r,u),(x',v')\rangle \le 0 \text{ for all } (x',v') \in T_G(x,v)\big\} \\ &= \big\{(r,u) \,\big|\, \langle r,x'\rangle + \langle u,v'\rangle \le 0 \text{ for all} \\ &\qquad\qquad x' \in K(x,v),\ v' \in K(x,v)^* \text{ with } x' \perp v'\big\}. \end{aligned}$$

It is evident from this (first in considering $v' = 0$, then in considering $x' = 0$) that actually

$$T_G(x,v)^* = K(x,v)^* \times K(x,v). \tag{13}$$

Hence $\widetilde{N}_G(x_0,v_0)$ is the union of all product sets $K^* \times K$ associated with cones K such that $K = K(x,v)$ for some $(x,v) \in G$ near enough to (x_0,v_0).

We claim now that the cones K arising in this manner are precisely the cones of the form $F_1 - F_2$ where F_1 and F_2 are closed faces of $K_0 = K(x_0,v_0)$ satisfying $F_1 \supset F_2$. This will be enough to prove the theorem by way of (11), (12), and (13).

For any vector $v \in \mathbb{R}^n$, let $[v] = \{\tau v \,|\, \tau \in \mathbb{R}\}$. Of course, this is a subspace of dimension 1 if $v \neq 0$, but just $\{0\}$ if $v = 0$. Accordingly, $[v]^\perp$ is a hyperplane through the origin if $v \neq 0$, but $[v]^\perp = \mathbb{R}^n$ if $v = 0$.

Because C is polyhedral, we know that for $x \in C$ sufficiently near to x_0 we are sure to have

$$\begin{aligned} T_C(x) &= T_C(x_0) + [x - x_0] \supset T_C(x_0), \\ N_C(x) &= N_C(x_0) \cap [x - x_0]^\perp \subset N_C(x_0). \end{aligned}$$

Furthermore, the vectors $x - x_0$ for $x \in C$ sufficiently near to x_0 are the vectors $x' \in T_C(x_0)$ having sufficiently small norm. On the other hand, the cones of form $T_C(x_0) \cap [v]^\perp$ for $v \in N_C(x_0)$ are the closed faces of $T_C(x_0)$, while the "lim sup" of $T_C(x_0) \cap [v]^\perp$ as $v \to v_0$ with $v \in N_C(x_0)$ is included within $T_C(x_0) \cap [v_0]^\perp$. Since $T_C(x_0)$ has only finitely many closed faces, we must have $T_C(x_0) \cap [v]^\perp \subset T_C(x_0) \cap [v_0]^\perp$ for $v \in N_C(x_0)$ sufficiently close to v_0. Since the critical cone $K_0 = T_C(x_0) \cap [v_0]^\perp$ is itself a closed face of $T_C(x_0)$, any closed face of $T_C(x_0)$ within K_0 is also a closed face of K_0.

In light of this, the cones $K = K(v,x)$ at points $(x,v) \in G$ arbitrarily near to (x_0,v_0) are the cones having the form

$$K = \big(T_C(x_0) + [x']\big) \cap [v]^\perp \text{ for some } x' \in T_C(x_0) \cap [v_0]^\perp \text{ and } v \in N_C(x_0) \cap [x']^\perp,$$

with v sufficiently close to v_0 and x' sufficiently close to 0 (with $x' = x - x_0$). We can write $K = \big(T_C(x_0) \cap [v]^\perp\big) + [x']$ equally well, because $x' \perp v$.

If K has this form, let $F_1 = T_C(x_0) \cap [v]^\perp$, this being a closed face of the polyhedral cone K_0 for reasons already given. We have $x' \in F_1$ and therefore actually $K = F_1 - F_2$, where F_2 is the closed face of F_1 having x' in its relative interior. Then F_2 is also a closed face of K_0, and the desired representation of K is achieved.

Conversely, if $K = F_1 - F_2$ for closed faces F_1 and F_2 of K_0 with $F_1 \supset F_2$, there must be a vector $v \in N_C(x_0)$ with $T_C(x_0) \cap [v]^\perp = F_1$. Then F_2 is a closed face of

F_1. Let $x' \in \operatorname{ri} F_2$; in particular $x' \in T_C(x_0)$, so by taking the norm of x' sufficiently small we can arrange that the point $x = x_0 + x'$ lies in C. We have $x' \perp v$ and

$$\begin{aligned} F_1 - F_2 &= 7\big(T_C(x_0) \cap [v]^\perp\big) + [x'] = \big(T_C(x_0) + [x']\big) \cap [v]^\perp \\ &= \big(T_C(x_0) + [x - x_0]\big) \cap [v]^\perp = T_C(x) \cap [v]^\perp, \end{aligned}$$

which is the form required. □

COROLLARY 1. *A sufficient condition for the Aubin property to hold for L at (q_0, x_0), and therefore all the other equivalent properties in Theorem 1 as well, is that $\langle u, Au\rangle > 0$ for all vectors $u \neq 0$ in the subspace $K_0 - K_0$ spanned by the critical cone K_0.*

Proof. The inequality $\langle u, Au\rangle \leq 0$ is equivalent to $\langle u, A^\top u\rangle \leq 0$, which must hold in particular when u belongs to a cone $F_1 - F_2 \subset K_0 - K_0$ and $A^\top u \in (F_1 - F_2)^*$. In the circumstances described, this is impossible unless $u = 0$. □

Remark 3. In consequence of Theorem 2, the critical face condition is both necessary and sufficient for the coherent orientation of the normal map associated with the linear variational inequality (2).

3. The nonlinear problem. Now we extend our results to the nonlinear variational inequality (1), taking the linear variational inequality (2) to be its linearization as indicated by (3). We start by recording a background fact about our underlying assumption (A) in §1.

STRICT DIFFERENTIABILITY LEMMA. *Under* (A) *there exist for any $\varepsilon > 0$ neighborhoods U of x_0 and V of w_0 such that for all $x_1, x_2 \in U$ and $w \in V$,*

$$\big\|f(w, x_1) - f(w, x_2) - \nabla_x f(w_0, x_0)(x_1 - x_2)\big\| \;\leq\; \varepsilon\big\|x_1 - x_2\big\|.$$

Proof. This is classical, but we supply the proof for completeness. For an arbitrary $e \in \mathbb{R}^n$ with $\|e\| = 1$ and any $x_1, x_2 \in \mathbb{R}^n$ and $w \in \mathbb{R}^d$ we can apply the mean value theorem to $\varphi(t) = \langle e, f(w, tx_1 + (1-t)x_2)\rangle$ to get a value $\tau \in (0,1)$ such that $\varphi(1) - \varphi(0) = \varphi'(\tau)$; i.e.,

$$\big\langle e, f(w, x_1)\big\rangle - \big\langle e, f(w, x_2)\big\rangle = \big\langle e, \nabla_x f(w, \tau x_1 + (1-\tau)x_2)(x_1 - x_2)\big\rangle.$$

Choose neighborhoods U of x_0 and V of w_0 such that U is convex and $\|\nabla_x f(w, x) - \nabla_x f(w_0, x_0)\| \leq \varepsilon$ when $x \in U$ and $w \in V$, as is possible by virtue of the continuity of $\nabla_x f(w, x)$ in w and x that is assumed in (A). For all $x_1, x_2 \in U$ and $w \in V$ we have

$$\begin{aligned} &\langle e,\, [f(w, x_1) - f(w, x_2) - \nabla_x f(w_0, x_0)(x_1 - x_2)]\,\rangle \\ &\quad = \langle e,\, [\nabla_x f(w, \tau x_1 + (1-\tau)x_2) - \nabla_x f(w_0, x_0)](x_1 - x_2)\,\rangle \\ &\quad \leq \|\nabla_x f(w, \tau x_1 + (1-\tau)x_2) - \nabla_x f(w_0, x_0)\|\|x_1 - x_2\| \\ &\quad \leq \varepsilon\|x_1 - x_2\|. \end{aligned}$$

Since this is true for all e with $\|e\| = 1$, we get the required estimate. □

PROPOSITION 1. *The following are equivalent for the maps L and S:*

(i) *L has the Aubin property at (q_0, x_0);*

(ii) *S has the Aubin property at (p_0, x_0).*

Proof. This can be obtained at once from the observation that the Mordukhovich coderivative criterion, as invoked in the proof of Theorem 2, has the same form for (1) at (p_0, x_0) that it has for (2) at (q_0, x_0), because the coderivative map associated with S is the same as for L. But we proceed anyway with an independent proof which shows how this equivalence extends beyond such a framework; cf. Remark 4 below.

Let L have the Aubin property at (q_0, x_0) with a constant M; that is, for some $a > 0$ and $b > 0$ and for every $q', q'' \in \mathbb{B}_b(q_0)$ we have

$$L(q') \cap \mathbb{B}_a(x_0) \subset L(q'') + M\|q' - q''\|\mathbb{B}. \tag{14}$$

Let $\varepsilon > 0$ be such that $M\varepsilon < 1$. Choose $\alpha > 0$ and $\beta_1 > 0$ with

$$\alpha < \min\{a, b/\varepsilon\}$$

such that the inequality in the Strict Differentiability Lemma holds whenever $x', x'' \in \mathbb{B}_\alpha(x_0)$ and $w \in \mathbb{B}_{\beta_1}(w_0)$. Let $\beta > 0$ be such that

$$\beta \le \min\left\{\beta_1, \frac{\alpha(1-\varepsilon M)}{4M(1+l)}, \frac{b-\varepsilon\alpha}{1+l}\right\}. \tag{15}$$

It will be demonstrated that S has the Aubin property at (p_0, x_0) with constant $M' = M(l+1)/[1-\varepsilon M]$.

Fix $p', p'' \in \mathbb{B}_\beta(p_0)$, with $p' = (z', w')$ and $p'' = (z'', w'')$, and consider any $x' \in S(p') \cap \mathbb{B}_{\alpha/2}(x_0)$. Then

$$0 \in z' + f(w', x') + N_C(x') = [z' + f(w', x') - \nabla_x f(w_0, x_0)x'] + Ax' + N_C(x'),$$

so that $x' \in L(q') \cap \mathbb{B}_{\alpha/2}(x_0)$ for $q' = z' + f(w', x') - \nabla_x f(w_0, x_0)x'$, where in terms of the linearization map

$$g(x) = f(w_0, x_0) + \nabla_x f(w_0, x_0)(x - x_0) = q_0 - z_0 + \nabla_x f(w_0, x_0)x \tag{16}$$

we can write $q' - q_0 = z' - z_0 + f(w', x') - g(x')$. Using (15) we have

$$\begin{aligned}\|q' - q_0\| &= \|z' - z_0 + f(w', x') - g(x')\| \\ &\le \|z' - z_0\| + \|f(w', x') - f(w', x_0) - \nabla_x f(w_0, x_0)(x' - x_0)\| \\ &\qquad + \|f(w', x_0) - f(w_0, x_0)\| \\ &\le \|p' - p_0\| + \varepsilon\|x' - x_0\| + l\|w' - w_0\| \le \beta(1+l) + \frac{\varepsilon\alpha}{2},\end{aligned}$$

$$\text{so that } \|q' - q_0\| \le b; \quad \text{that is, } q' \in \mathbb{B}_b(q_0). \tag{17}$$

Analogously, for the vector $q'' = z'' + f(w'', x') - \nabla_x f(w_0, x_0)x' = q_0 + z'' - z_0 + f(w'', x') - g(x')$ we have $q'' \in \mathbb{B}_b(q_0)$. Let $x_1 = x'$. On the basis of (14) there exists then an x_2 such that

$$z'' + f(w'', x_1) + \nabla_x f(w_0, x_0)(x_2 - x_1) + N_C(x_2) \ni 0$$

and

$$\begin{aligned}\|x_2 - x_1\| &\le M\|q' - q''\| \le M(\|z' - z''\| + \|f(w', x_1) - f(w'', x_1)\|) \\ &\le M(\|z' - z''\| + l\|w' - w''\|) \le M(l+1)\|p' - p''\|.\end{aligned}$$

Suppose that there exist points $x_2, x_3, \ldots, x_{n-1}$ with

$$z'' + f(w'', x_{i-1}) + \nabla_x f(w_0, x_0)(x_i - x_{i-1}) + N_C(x_i) \ni 0$$

and

$$\|x_i - x_{i-1}\| \le M(l+1)\|p' - p''\|(M\varepsilon)^{i-2} \text{ for } i = 2, \dots, n-1.$$

Then for every i we have

$$\begin{aligned}\|x_i - x_0\| &\le \|x_1 - x_0\| + \sum_{j=2}^{i} \|x_j - x_{j-1}\| \\ &\le \frac{\alpha}{2} + M(l+1)\|p' - p''\| \sum_{j=2}^{i} (M\varepsilon)^{j-2} \\ &\le \frac{\alpha}{2} + \frac{M(l+1)}{1 - \varepsilon M}\|p' - p''\| \le \frac{\alpha}{2} + \frac{2M\beta(l+1)}{1 - \varepsilon M} \le \alpha\end{aligned}$$

because of (15). Setting $q_i = z'' + f(w'', x_i) - \nabla_x f(w_0, x_0)x_i = q_0 + z'' - z_0 + f(w'', x_i) - g(x_i)$ for $i = 2, 3, \dots, n-1$ we get

$$\begin{aligned}\|q_i - q_0\| &= \|z'' - z_0 + f(w'', x_i) - g(x_i)\| \\ &\le \|z'' - z_0\| + \|f(w'', x_i) - f(w'', x_0) - \nabla_x f(w_0, x_0)(x_i - x_0)\| \\ &\qquad + \|f(w'', x_0) - f(w_0, x_0)\| \\ &\le \|p' - p_0\| + \varepsilon\|x_i - x_0\| + l\|w'' - w_0\| \\ &\le \beta(1 + l) + \varepsilon\alpha \le b,\end{aligned}$$

so that $q_i \in \mathbb{B}_b(q_0)$. Since $x_{n-1} \in L(q_{n-2}) \cap \mathbb{B}_\alpha(x_0)$, we know from the Aubin property (14) that there exists x_n with

$$z'' + f(w'', x_{n-1}) + \nabla_x f(w_0, x_0)(x_n - x_{n-1}) + N_C(x_n) \ni 0 \tag{18}$$

and

$$\begin{aligned}\|x_n - x_{n-1}\| &\le M\|q_{n-1} - q_{n-2}\| \\ &\le M\|f(w'', x_{n-1}) - f(w'', x_{n-2}) - \nabla_x f(w_0, x_0)(x_{n-1} - x_{n-2})\| \\ &\le M\varepsilon\|x_{n-1} - x_{n-2}\| \le M(l+1)\|p' - p''\|(M\varepsilon)^{n-2}.\end{aligned}$$

The induction step is thereby joined. We obtain an infinite sequence of points $x_1, x_2, \dots, x_n, \dots$ in $\mathbb{B}_\alpha(x_0)$ that is a Cauchy sequence and therefore converges to some $x'' \in \mathbb{B}_\alpha(x_0)$. Since $f(w'', \cdot)$ is continuous in $\mathbb{B}_\alpha(x_0)$ and the normal cone map N_C has a closed graph, it follows from (18) that $x'' \in S(p'')$. Moreover, since

$$\begin{aligned}\|x_n - x'\| &\le \sum_{i=2}^{n} \|x_i - x_{i-1}\| \\ &\le M(l+1)\|p' - p''\| \sum_{i=2}^{n} (M\varepsilon)^{i-2} \le \frac{M(l+1)}{1 - \varepsilon M}\|p' - p''\|,\end{aligned}$$

we obtain in passing to the limit that

$$\|x'' - x'\| \le \frac{M(l+1)}{1 - \varepsilon M}\|p' - p''\| = M'\|p' - p''\|.$$

The implication (i) $\Rightarrow$ (ii) is thereby established.

To prove the implication (ii) $\Rightarrow$ (i), suppose S has the Aubin property at (p_0, x_0) with constant M. Choose ε, α, and β relative to M as above. It will be demonstrated that L has the Aubin property at (q_0, x_0) with constant $M' = M/(1-\varepsilon M)$. Consider $q', q'' \in \mathbb{B}_\beta(q_0)$, and $x' \in L(q') \cap \mathbb{B}_{\alpha/2}(x_0)$:

$$q' + \nabla_x f(w_0, x_0)x' + N_C(x') \ni 0.$$

Then $x' \in S(p') \cap \mathbb{B}_{\alpha/2}(x_0)$ for the parameter element $p' = (z', w_0)$ with $z' = q' + \nabla_x f(w_0, x_0)x' - f(w_0, x') = z_0 + [q' - q_0] - [f(w_0, x') - g(x')]$. Now also let $p'' = (z'', w_0)$ for the vector $z'' = q'' + \nabla_x f(w_0, x_0)x' - f(w_0, x') = z_0 + [q'' - q_0] - [f(w_0, x') - g(x')]$. As in the chain of estimates leading to (17) we get $p', p'' \in \mathbb{B}_b(p_0)$. Then there exists x_2 such that

$$z'' + f(w_0, x_2) + f(w_0, x_0) + \nabla_x f(w_0, x_0)(x' - x_0) - f(w_0, x') + N_C(x_2) \ni 0,$$

$$\|x_2 - x'\| \leq M\|p' - p''\| = M\|z' - z''\|.$$

By emulating the argument in the first part of the proof, we obtain by induction a sequence $x' = x_1, x_2, \ldots, x_n, \ldots$ convergent to x'' and such that

$$z'' + f(w_0, x_n) + f(w_0, x_0) + \nabla_x f(w_0, x_0)(x_{n-1} - x_0) - f(w_0, x_{n-1}) + N_C(x_n) \ni 0,$$

$$\|x_n - x'\| \leq M\|q' - q''\| \sum_{i=2}^{n} (M\varepsilon)^{i-2}.$$

Passing to the limit we obtain that $x'' \in L(q'')$ and

$$\|x' - x''\| \leq \frac{M}{1 - \varepsilon M}\|q' - q''\| = M'\|q' - q''\|.$$

This finishes the proof. □

Remark 4. The result in Proposition 1 carries over to a much wider setting, as may be gleaned from the proof we have given. Let F be a set-valued map with closed graph from a complete metric space X into the subsets of a linear normed space Z, and let f be a function from $W \times X$ to Z, where W is a metric space. Suppose that $g : X \mapsto Z$ is a continuous function which strongly approximates f around (w_0, x_0) in the sense of Robinson [25] and that f satisfies condition (B) (with the metric of W replacing the norm). Consider the maps

$$\Sigma(p) = \{x \in X \mid 0 \in z + f(w, x) + F(x)\},$$

where $p = (z, w)$, and

$$\Lambda(z) = \{x \in X \mid 0 \in z + g(x) + F(x)\},$$

and let $x_0 \in \Sigma(p_0)$. Then Σ has the Aubin property at (p_0, x_0) if and only if Λ has it at (z_0, x_0). Prototypes of such a theorem are contained in [5] and [6], where a fixed point argument is utilized. □

Next we give a new proof of the original result of Robinson in [23] (for the present context). In contrast to Robinson's argument, we do not appeal to a fixed point theorem but rely on Proposition 1 instead. Furthermore, whereas Robinson focused

on the implication from the property of L to the corresponding one for S, we point out that—in the presence of the canonical perturbation vector z alongside of w in the element $p = (z, w)$—the implication goes both ways and becomes an equivalence.

PROPOSITION 2. *The following properties are equivalent:*

(i) *L is locally single valued and Lipschitz continuous around (q_0, x_0);*

(ii) *S is locally single valued and Lipschitz continuous around (p_0, x_0).*

Proof. Let (i) hold. In particular, L has the Aubin property, hence S has it too by Proposition 2. To get (ii) it suffices therefore to verify that S is locally single valued.

Suppose to the contrary that in every neighborhood V of p_0 and X of x_0 there exist $\bar{p} = (\bar{z}, \bar{w})$ and $x_1, x_2 \in S(\bar{p}) \cap X$ such that $x_1 \neq x_2$. In particular,

$$\bar{z} + f(\bar{w}, x_i) + N_C(x_i) \ni 0 \text{ for } i = 1, 2.$$

Let M be the Lipschitz constant of L around (q_0, x_0) and choose $\varepsilon > 0$ small enough that $M\varepsilon < 1$. Using the Strict Differentiability Lemma, choose neighborhoods U of x_0 and V of w_0 with $U \subset X$ such that

$$\|f(w, x') - f(w, x'') - \nabla_x f(w_0, x_0)(x' - x'')\| \leq \varepsilon \|x' - x''\|$$

for every $x', x'' \in U$, and $w \in V$. Note that for $i = 1, 2$ and for some sufficiently small neighborhood $U' \subset U$ of x_0, we have $L(q_i) \cap U' = \{x_i\}$ for $q_i = q' + \nabla_x f(w_0, x_0)x' - f(w_0, x') = q_0 + [\bar{z} - z_0] + [f(\bar{w}, x_i) - g(x_i)]$, where g is given as before by (16). Then

$$\begin{aligned}
\|x_1 - x_2\| &\leq M\|q_1 - q_2\| \\
&= M\|[f(\bar{w}, x_1) - g(x_1)] - [f(\bar{w}, x_2) - g(x_2)]\| \\
&= M\|f(\bar{w}, x_1) - f(\bar{w}, x_2) - \nabla_x f(x_0, w_0)(x_1 - x_2)\| \\
&\leq M\varepsilon\|x_1 - x_2\| < \|x_1 - x_2\|,
\end{aligned}$$

which is a contradiction. Hence S is locally single valued around (p_0, x_0).

The converse implication (ii) $\Rightarrow$ (i) is established in the same way. □

Combining Propositions 1 and 2 with Theorems 1 and 2 we obtain the following result, in which the implication from (i) to (iv), already known from Robinson's theorem in [23], has ended up in a circle of equivalences.

THEOREM 3. *The following properties are equivalent:*

(i) *the nonlinear variational inequality* (1) *is strongly regular at (p_0, x_0);*

(ii) *the critical face condition of Definition* 2 *holds at (q_0, x_0);*

(iii) *the solution map S has the Aubin property at (p_0, x_0);*

(iv) *the solution map S is locally single valued and Lipschitz continuous around (p_0, x_0).*

4. Application to the complementarity problem. Next we apply our results to the nonlinear complementarity problem with canonical perturbations, namely

$$x \geq 0, \quad f(w, x) + z \geq 0, \quad \langle x, f(w, x) + z \rangle = 0, \tag{19}$$

which is the special case of (1) with $C = \mathbb{R}^n_+$. Our general assumptions and notation for (1) continue in the analysis of this case, in particular (3) and (4). We associate with the vector $v_0 \in N_C(x_0)$ the index sets J_1, J_2, J_3 in $\{1, 2, \ldots, n\}$ given by

$$\begin{aligned}
J_1 &= \{i \mid x_0^i > 0,\ v_0^i = 0\}, \\
J_2 &= \{i \mid x_0^i = 0,\ v_0^i = 0\}, \\
J_3 &= \{i \mid x_0^i = 0,\ v_0^i < 0\}.
\end{aligned}$$

PROPOSITION 3. *In the case of the nonlinear complementarity problem, the critical cone K_0 consists of the vectors x' satisfying*

$$\begin{cases} x'_i \text{ free} & \text{for } i \in J_1, \\ x'_i \geq 0 & \text{for } i \in J_2, \\ x'_i = 0 & \text{for } i \in J_3, \end{cases}$$

and the cones $F_1 - F_2$, where F_1 and F_2 are closed faces of K_0 with $F_1 \supset F_2$, are the cones K of the following form. There is a partition of $\{1, 2, \ldots, n\}$ into index sets J'_1, J'_2, J'_3 with $J_1 \subset J'_1 \subset J_1 \cup J_2$ and $J_3 \subset J'_3 \subset J_3 \cup J_2$, such that K consists of the vectors x' satisfying

$$\begin{cases} x'_i \text{ free} & \text{for } i \in J'_1, \\ x'_i \geq 0 & \text{for } i \in J'_2, \\ x'_i = 0 & \text{for } i \in J'_3. \end{cases} \tag{20}$$

The vectors $u \in K$ with $A^\top u \in K^*$ are then the ones such that

$$\begin{cases} u_i \text{ free}, \ (A^\top u)_i = 0 & \text{for } i \in J'_1, \\ u_i \geq 0, \ (A^\top u)_i \leq 0 & \text{for } i \in J'_2, \\ u_i = 0, \ (A^\top u)_i \text{ free} & \text{for } i \in J'_3. \end{cases}$$

Proof. It is easy to see that K_0 has the form described, so we focus on analyzing its closed faces. Each such face F has the form $K_0 \cap [v']^\perp$ for some vector $v' \in K_0^*$. The vectors v in question are those with

$$\begin{cases} v'_i = 0 & \text{for } i \in J_1, \\ v'_i \leq 0 & \text{for } i \in J_2, \\ v'_i \text{ free} & \text{for } i \in J_3. \end{cases}$$

The closed faces F of K_0 correspond one-to-one therefore with the subsets of J_2: the face F corresponding to an index set J_2^F consists of the vectors x' such that

$$\begin{cases} x'_i \text{ free} & \text{for } i \in J_1, \\ x'_i \geq 0 & \text{for } i \in J_2 \setminus J_2^F, \\ x'_i = 0 & \text{for } i \in J_3 \cup J_2^F. \end{cases}$$

If F_1 and F_2 have $J_2^{F_1} \subset J_2^{F_2}$, so that $F_1 \supset F_2$, then $F_1 - F_2$ is given by (20) with $J'_1 = J_1 \cup [J_2 \setminus J_2^{F_2}]$, $J'_2 = J_2^{F_2} \setminus J_2^{F_1}$, $J'_3 = J_3 \cup J_2^{F_1}$. □

THEOREM 4. *The general complementarity problem* (19) *is strongly regular at (p_0, x_0) if and only if the following condition holds for the entries a_{ij} of A: if u_i for $i \in J_1 \cup J_2$ are numbers satisfying*

$$\sum_{i \in J_1 \cup J_2} u_i a_{ij} \begin{cases} = 0 & \text{for } j \in J_1 \text{ and for } j \in J_2 \text{ with } u_j < 0, \\ \leq 0 & \text{for } j \in J_2 \text{ with } u_j > 0, \end{cases}$$

then $u_i = 0$ for all $i \in J_1 \cup J_2$.

Proof. This condition specializes the critical face condition to this setting, as seen from Proposition 3. It remains only to apply Theorem 3. □

Remark 5. The sufficiency of the condition in Theorem 4 for strong regularity can also be proved directly. Consider the linear complementarity problem (5) and its solution map L_0, which has $x_0 \in L_0(q_0)$. Assume temporarily that $J_1 = \emptyset$ and $J_3 = \emptyset$.

Let a be a subset of $\{1,2,\dots,n\}$ and let A_{aa} be the corresponding submatrix of A. Let $u^a \in \ker A_{aa}^\top$ and

$$u_j = \begin{cases} u_j^a & \text{if } j \in a, \\ 0 & \text{otherwise.} \end{cases}$$

Then $(A^\top u)_j = 0$ for all j for which $u_j \neq 0$. Hence, from the condition displayed in Theorem 4, we have $u = 0$. Thus A_{aa} is nonsingular, and we see every principal submatrix of A is nonsingular. Furthermore, let $j \in \{1,2,\dots,n\}$ and let $u \in \mathbb{R}^n$ be such that $u_j = 1$ and $u_i = 0$ for all $i \neq j$. Then $(A^\top u)_j = a_{jj}$. If $a_{jj} \leq 0$, the condition in Theorem 4 implies that $u = 0$, a contradiction. Hence $a_{ii} > 0$ for all i.

There is a linear one-to-one correspondence between the graph of the solution map L_0 and the graph of the solution map of any principal pivotal transform of the linear complementarity problem (5); see e.g., [4]. Then the solution map of any principal pivotal transform of (5) has the Aubin property at $(0,0)$. By the above argument all diagonal entries of any principal pivotal transform of A are positive. This means that A is a P-matrix; see [19, p. 205]. This in turn is equivalent to the condition that L_0 is single valued everywhere.

Consider now the general case. First observe that the submatrix A_{11} corresponding to the set J_1 is nonsingular. (Take $u^1 \in \ker A_{11}$ and $u_j = u_j^1$ if $j \in J_1$, but $u_j = 0$ otherwise; then $(A^\top u)_j = 0$ for all $j \in J_1$, hence from the condition in Theorem 4, $u^1 = 0$.) Utilizing the Reduction Lemma we come to a complementarity problem of the form

$$\begin{aligned} 0 &= A_{11}x^1 + A_{12}x^2 + q^1, \\ z^2 &= A_{12}x^1 + A_{22}x^2 + q^2, \\ x^2 &\geq 0,\ z^2 \geq 0,\ \langle x^2, z^2 \rangle = 0, \end{aligned}$$

where the superscripts of x correspond to the sets of indices J_1 and J_2. By solving the first equation and substituting to the second one we obtain a problem whose solution map has the Aubin property at $(0,0)$; that is, $J_1 = \emptyset$, $J_3 = \emptyset$. This case has already been treated. □

As a corollary of Theorem 4 we obtain the following characterization of P-matrices.

COROLLARY 2. *For an $n \times n$ matrix A, the following are equivalent:*

(i) *A is a P-matrix;*

(ii) *for $u \in \mathbb{R}^n$,*

$$(A^\top u)_j \begin{cases} = 0 & \text{for } j \text{ with } u_j < 0, \\ \leq 0 & \text{for } j \text{ with } u_j > 0 \end{cases} \quad \Longrightarrow \quad u = 0.$$

5. Application to nonlinear programming. For a further illustration of our general results we consider the nonlinear programming problem

$$\text{minimize } g_0(w,x) + \langle v, x \rangle \text{ in } x \text{ subject to}$$

$$g_i(w,x) - u_i \begin{cases} = 0 & \text{for } i \in [1,r], \\ \leq 0 & \text{for } i \in [r+1,m] \end{cases} \tag{21}$$

for $\mathcal{C}^2$ functions $g_i : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, $i = 0,1,\dots,m$, where the vectors $w \in \mathbb{R}^d$, $v \in \mathbb{R}^n$, and $u = (u_1,\dots,u_m) \in \mathbb{R}^m$ are parameters. In terms of

$$L(w,x,y) = g_0(w,x) + y_1 g_1(w,x) + \cdots + y_m g_m(w,x),$$

the first-order optimality conditions for this problem, namely the KKT conditions, take the form

$$\begin{cases} v + \nabla_x L(w,x,y) = 0, \\ -u + \nabla_y L(w,x,y) \in N_Y(y) \quad \text{for } Y = \mathbb{R}^r \times \mathbb{R}^{m-r}_+. \end{cases} \tag{22}$$

These can be written together as the variational inequality

$$(v,u) + f(w,x,y) + N_C(x,y) \ni (0,0) \tag{23}$$

under the choice of elements

$$f(w,x,y) = \big(\nabla_x L(w,x,y), -\nabla_y L(w,x,y)\big), \qquad C = \mathbb{R}^n \times Y. \tag{24}$$

(Obviously $\nabla_y L(w,x,y)$ is simply the vector in $\mathbb{R}^m$ having as its components the values $g_i(w,x)$ for $i = 1,\dots,m$.) Here (x,y) replaces the point x of the general theory, while (v,u) corresponds to the canonical perturbation vector z. The set C is a polyhedral convex cone, and the map f satisfies our blanket assumptions (A) and (B). (Weaker conditions on the g_i's would suffice, but we leave that aside.)

Consider any pair (x_0,y_0) satisfying the KKT conditions (22)—or equivalently the variational inequality (23)—for given u_0, v_0, and w_0. We wish to work out what our results say about strong regularity in this variational inequality at (u_0,v_0,w_0,x_0,y_0) and therefore about the local single valuedness and Lipschitz continuity of the map S_{KKT} that assigns to each (u,v,w) the set of KKT pairs (x,y) in problem (21).

Associate with the given elements u_0, v_0, w_0, x_0, y_0, the index sets I_1, I_2, I_3 in $\{1,2,\dots,m\}$ defined by

$$\begin{aligned} I_1 &= \big\{ i \in [r+1,m] \,\big|\, g_i(w_0,x_0) - u_{0i} = 0,\ y_{0i} > 0 \big\} \cup \{1,\dots,r\}, \\ I_2 &= \big\{ i \in [r+1,m] \,\big|\, g_i(w_0,x_0) - u_{0i} = 0,\ y_{0i} = 0 \big\}, \\ I_3 &= \big\{ i \in [r+1,m] \,\big|\, g_i(w_0,x_0) - u_{0i} < 0,\ y_{0i} = 0 \big\}. \end{aligned}$$

The tangent cone $T_C(x_0,y_0)$ consists of all the vectors $(x',y') \in \mathbb{R}^n \times \mathbb{R}^m$ such that

$$\begin{cases} x' \text{ free}, & \\ y'_i \text{ free} & \text{for } i \in I_1, \\ y'_i \geq 0 & \text{for } i \in I_2 \cup I_3. \end{cases}$$

By definition the critical cone K_0 consists of all $(x',y') \in T_C(x_0,y_0)$ orthogonal to the vector $(v_0,u_0) + f(w_0,x_0,y_0)$, but the KKT conditions imply that all the components of this vector are 0 except for the ones at the end corresponding to inactive inequality constraints. From this it is apparent that the critical cone is given by

$$(x',y') \in K_0 \iff \begin{cases} x' \text{ free}, & \\ y'_i \text{ free} & \text{for } i \in I_1, \\ y'_i \geq 0 & \text{for } i \in I_2, \\ y'_i = 0 & \text{for } i \in I_3. \end{cases} \tag{25}$$

On the other hand, the matrix A in the critical face condition specializes to

$$A = \begin{bmatrix} H(w_0,x_0,y_0) & G(w_0,x_0)^\top \\ -G(w_0,x_0) & 0 \end{bmatrix} \tag{26}$$

for the second-derivative matrix $H(w,x,y) = \nabla^2_{xx} L(w,x,y)$ and the matrix $G(w,x) = \nabla^2_{yx} L(w,x,y)$ having as its rows the constraint gradient vectors $\nabla_x g_i(w,x)$ for $i = 1,\dots,m$.

THEOREM 5. *The variational inequality* (23)–(24) *associated with the KKT conditions* (22) *is strongly regular for* $(u_0, v_0, w_0, x_0, y_0)$ *if and only if the following two requirements, specializing the critical face condition to this setting, are fulfilled:*

(a) *the vectors* $\nabla_x g_i(w_0, x_0)$ *for* $i \in I_1 \cup I_2$ *are linearly independent;*

(b) *for each partition of* $\{1, 2, \ldots, m\}$ *into index sets* I_1', I_2', I_3' *with* $I_1 \subset I_1' \subset I_1 \cup I_2$ *and* $I_3 \subset I_3' \subset I_3 \cup I_2$, *the cone* $K(I_1', I_2') \subset \mathbb{R}^n$ *consisting of all the vectors* x' *satisfying*

$$\langle \nabla_x g_i(w_0, x_0), x' \rangle \begin{cases} = 0 & \text{for } i \in I_1', \\ \leq 0 & \text{for } i \in I_2', \end{cases}$$

should be such that

$$x' \in K(I_1', I_2'), \quad \nabla_{xx}^2 L(w_0, x_0, y_0) x' \in K(I_1', I_2')^* \quad \Longrightarrow \quad x' = 0.$$

Proof. From the analysis in §§3 and 4 and the observations just made, it is evident that the cones of form $F_1 - F_2$ in which F_1 and F_2 are closed faces of K_0 with $F_1 \supset F_2$ correspond one-to-one with the partitions (I_1', I_2', I_3') by

$$(x', y') \in F_1 - F_2 \iff \begin{cases} x' \text{ free}, \\ y_i' \text{ free} & \text{for } i \in I_1', \\ y_i' \geq 0 & \text{for } i \in I_2', \\ y_i' = 0 & \text{for } i \in I_3', \end{cases} \tag{27}$$

in which case

$$(x'', y'') \in (F_1 - F_2)^* \iff \begin{cases} x'' = 0, \\ y_i'' = 0 & \text{for } i \in I_1', \\ y_i'' \leq 0 & \text{for } i \in I_2', \\ y_i'' \text{ free} & \text{for } i \in I_3'. \end{cases}$$

In view of the structure determined for A in (26) the critical face condition emerges as the requirement that whenever x' and y' satisfy (27) and have

$$H(w_0, x_0, y_0) x' - \sum_{i=1}^m y_i' \nabla_x g_i(w_0, x_0) = 0 \text{ with } x' \in K(I_1', I_2'),$$

then $x' = 0$ and $y' = 0$. The vectors of the form $\sum_{i=1}^m y_i' \nabla_x g_i(w_0, x_0)$ with y' satisfying (27) are of course the ones in the polar cone $K(I_1', I_2')^*$ (by Farkas's lemma), so we see that the critical cone condition comes down to (b) along with the requirement that no $\sum_{i=1}^m y_i' \nabla_x g_i(w_0, x_0)$ with y' satisfying (27) can vanish unless $y' = 0$. Since the partition $I_1' = I_1 \cup I_2$, $I_2' = 0$, $I_3' = I_3$ can be taken as a special case, the latter means neither more nor less than the linear independence in (a). □

By combining Theorem 5 with second-order conditions we obtain a characterization of the case where the KKT map also gives local optimality.

THEOREM 6. *The following are equivalent:*

(i) *the map* S_{KKT} *is locally single valued and Lipschitz continuous around* $(u_0, v_0, w_0, x_0, y_0)$, *moreover with the property that for all* $(u, v, w, x, y) \in \operatorname{gph} S_{KKT}$ *in some neighborhood of* $(u_0, v_0, w_0, x_0, y_0)$, x *is a locally optimal solution to the nonlinear programming problem* (21) *for* (u, v, w);

(ii) *the constraint gradients* $\nabla_x g_i(w_0, x_0)$ *for* $i \in I_1 \cup I_2$ *are linearly independent and the strong second-order sufficient condition for local optimality holds for* $(u_0, v_0, w_0, x_0, y_0)$. *One has*

$$\langle x', \nabla_{xx}^2 L(w_0, x_0, y_0) x' \rangle > 0 \text{ for all } x' \neq 0 \text{ in the subspace}$$

$$M = \{x' \mid x' \perp \nabla_x g_i(w_0, x_0) \text{ for all } i \in I_1\}.$$

Proof. From Theorem 5 we already know that the local single valuedness and Lipschitz continuity in (i) require the linear independence in (ii). On the other hand, the positive definiteness in (ii) suffices by Theorem 5 for S_{KKT} to be locally single valued and Lipschitz continuous around $(u_0, v_0, w_0, x_0, y_0)$, because $\langle x', \nabla^2_{xx} L(w_0, x_0, y_0)x' \rangle \leq 0$ when x' and $\nabla^2_{xx} L(w_0, x_0, y_0)x'$ belong to cones that are polar to each other. Henceforth we therefore work in the picture of S_{KKT} being locally single valued and Lipschitz continuous around $(u_0, v_0, w_0, x_0, y_0)$, with both (a) and (b) of Theorem 5 holding. In particular then we have

$$x' \in M, \quad \nabla^2_{xx} L(w_0, x_0, y_0)x' \in M^\perp \quad \Longrightarrow \quad x' = 0 \tag{28}$$

because $K(I_1', I_2') = M$ and $K(I_1', I_2') = M^\perp$ when $I_1' = I_1$, $I_2' = \emptyset$, $I_3' = I_2 \cup I_3$. The focus is on verifying that the local optimality in (i) corresponds in these circumstances to the positive definiteness in (ii).

For simplicity we denote by $S_0(u, v, w)$ the uniquely determined pair (x, y) in the local single valuedness of $S_{\mathrm{KKT}}(u, v, w)$. We limit attention to parameter elements (u, v, w) near enough to (u_0, v_0, w_0) for this to make sense.

Let $P(u, v, w)$ be the nonlinear programming problem associated with (u, v, w) in (21). This problem has $(x, y) = S_0(u, v, w)$ as a KKT pair, with $(x, y) \to (x_0, y_0)$ as $(u, v, w) \to (u_0, v_0, w_0)$. For (u, v, w) close to (u_0, v_0, w_0), the index sets $I_1(u, v, w)$, $I_2(u, v, w)$, and $I_2(u, v, w)$ that correspond to (x, y) the way I_1, I_2, and I_3 do to (x_0, y_0) must satisfy

$$I_1 \subset I_1(u, v, w) \subset I_1 \cup I_2, \qquad I_3 \subset I_3(u, v, w) \subset I_3 \cup I_2. \tag{29}$$

In particular, $I_1(u, v, w) \cup I_2(u, v, w) \subset I_1 \cup I_2$, so the gradients $\nabla_x g_i(w, x)$ for $i \in I_1(u, v, w) \cup I_2(u, v, w)$ must be linearly independent. We know then that for x to be locally optimal in $P(u, v, w)$ it is necessary that

$$\langle x', \nabla^2_{xx} L(w, x, y)x' \rangle \geq 0 \text{ for all } x' \text{ satisfying}$$

$$\langle \nabla_x g_i(w, x), x' \rangle \begin{cases} = 0 & \text{for } i \in I_1(u, v, w), \\ \leq 0 & \text{for } i \in I_2(u, v, w), \end{cases} \tag{30}$$

whereas a sufficient condition for x to be locally optimal in $P(u, v, w)$ is the same thing with "≥ 0" strengthened to "> 0" when $x' \neq 0$. (See [10, Thm. 10.1].)

The sufficiency just described leads immediately through (29) to the conclusion that the positive definiteness in (ii) entails the local optimality in (i). Conversely, if (i) holds, the second-order necessary condition must be satisfied by $(x, y) = S_0(u, v, w)$ for all (u, v, w) near enough to (u_0, v_0, w_0). The gradient linear independence property ensures that we can find a sequence of points $x^k \to x_0$ with

$$g_i(w_0, x^k) - u_{0i} \begin{cases} = 0 & \text{for } i \in I_1, \\ < 0 & \text{for } i \in I_2 \cup I_3. \end{cases}$$

For $v^k = -\nabla_x L(w_0, x^k, y_0)$ we have the KKT conditions in $P(u_0, v^k, w_0)$ satisfied by (x^k, y_0), hence $(x^k, y_0) = S_0(u_0, v^k, w_0)$ (for k sufficiently large). Then the necessary condition in (30) holds for these elements, with $I_1(u_0, v^k, w_0) = I_1$ and $I_2(u_0, v^k, w_0) = \emptyset$ so that the condition is just

$$\langle x', \nabla^2_{xx} L(w_0, x^k, y_0)x' \rangle \geq 0 \text{ for all } x' \in M.$$

In the limit as $k \to \infty$ we obtain (from the continuity of the second derivatives of L) that

$$\langle x', \nabla^2_{xx} L(w_0, x_0, y_0) x' \rangle \geq 0 \text{ for all } x' \in M.$$

This positive semidefiniteness relative to the subspace M must actually be positive definiteness, for otherwise there would have to exist by the symmetry of $\nabla^2_{xx} L(w_0, x_0, y_0)$ a vector $x' \neq 0$ in M with $\nabla^2_{xx} L(w_0, x_0, y_0) x' \in M^\perp$. That is impossible by (28). □

COROLLARY 3. *In the convex programming case (where $g_i(w, x)$ is affine in x for $i = 1, \ldots, r$ while $g_i(w, x)$ is convex in x for $i = 0$ and $i = r+1, \ldots, m$), condition* (ii) *of Theorem* 6 *is both necessary and sufficient for the map S_{KKT} to be locally single valued and Lipschitz continuous.*

Proof. In this case the local optimality in Theorem 6(i) is automatic. □

Several algebraic characterizations of strong regularity in optimization have previously been available in the literature. In his original paper [23], Robinson characterized the strong regularity of linear KKT systems. Kojima [14] introduced the concept of strong stability of the KKT system of a nonlinear program, which roughly means that the map S_{KKT} is locally single valued and continuous with respect to a sufficiently rich class of perturbations. He gave a characterization of this property and noted (in his Corollary 6.6.) that a KKT point (x_0, y_0) of the kind in which x_0 is a local minimizer and the gradients of the active constraints are linearly independent is strongly stable if and only if the strong second-order sufficient condition holds. Jongen et al. [11] proved through a far-reaching inertia-type theorem that strong stability in the sense of Kojima and strong regularity in the sense of Robinson are equivalent properties. Another characterization of the strong regularity of a KKT point was obtained by Kummer [15], who based his argument on a general implicit-function-type theorem for a nonsmooth equation equivalent to the KKT system. A related approach was developed in Jongen, Klatte, and Tammer [12] and Klatte and Tammer [13], the latter containing a survey of characterizations of strong regularity. Our critical face condition in Theorem 5 differs from the conditions of Robinson, Kojima, and Kummer, and it is not clear to us how one could derive the equivalence between these various conditions directly.

The implication (ii) $\Rightarrow$ (i) in Theorem 6 was noted by Robinson in [23, Thm. 4.1]. Note that the condition (i) in Theorem 6 implies, via Proposition 2, that x_0 is a local minimizer and (x_0, y_0) is a strongly regular KKT point. Furthermore, from the strong regularity one can obtain directly that the gradients of the active constraints are linearly independent. Then, by combining the results of Kojima [14] and Jongen et al. [11], one obtains (ii). Recently, Bonnans and Sulem [2] gave a different proof of this implication.

REFERENCES

[1] J.-P. AUBIN, *Lipschitz behavior of solutions to convex minimization problems*, Math. Oper. Res., 9 (1984), pp. 87–111.

[2] J. F. BONNANS AND A. SULEM, *Pseudopower expansion of solutions of generalized equations and constrained optimization problems*, Math. Programming, 70 (1995), pp. 123–148.

[3] J. M. BORWEIN AND D. M. ZHUANG, *Verifiable necessary and sufficient conditions for openness and regularity of set-valued maps*, J. Math. Anal. Appl., 134 (1988), pp. 441–459.

[4] R. W. COTTLE, JONG-SHI PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, Boston 1992.

[5] A. L. DONTCHEV AND W. W. HAGER, An inverse function theorem for set-valued maps, Proc. Amer. Math. Soc., 121 (1994), pp. 481–489.

[6] ———, *Implicit functions, Lipschitz maps and stability in optimization*, Math. Oper. Res., 19 (1994), pp. 753–768.
[7] B. C. EAVES AND U. G. ROTHBLUM, *Relationship of properties of piecewise affine maps over ordered fields*, Linear Algebra Appl., 132 (1990), pp. 1–63.
[8] M. S. GOWDA, On the continuity of the solution map in linear complementarity problems, SIAM J. Optim., 2 (1992), pp. 619–634.
[9] M. S. GOWDA AND R. SZNAJDER, *On the Lipschitz properties of polyhedral multifunctions*, Math. Programming, 74 (1996), pp. 267–278.
[10] M. HESTENES, *Calculus of Variations and Optimal Control Theory*, John Wiley, New York, 1966.
[11] H. TH. JONGEN, T. MÖBERT, J. RÜCKMANN, AND K. TAMMER, *Implicit functions and sensitivity of stationary points*, Linear Algebra Appl., 95 (1987), pp. 97–109.
[12] H. TH. JONGEN, D. KLATTE, AND K. TAMMER, *Implicit functions and sensitivity of stationary points*, Math. Programming, 19 (1990), pp. 123–138.
[13] D. KLATTE AND K. TAMMER, Strong stability of stationary solutions and Karush-Kuhn-Tucker points in nonlinear optimization, Ann. Oper. Res., 27 (1990), pp. 285–308.
[14] M. KOJIMA, *Strongly stable stationary solutions in nonlinear programming*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1988, pp. 93–138.
[15] B. KUMMER, *Lipschitzian inverse functions directional derivatives and application in $C^{1,1}$ optimization*, J. Optim. Theory Appl., 70 (1991), pp. 561–582.
[16] O. L. MANGASARIAN AND T.-H. SHIAU, *Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems*, SIAM J. Control Optim., 25 (1987), pp. 582–595.
[17] B. MORDUKHOVICH, *Stability theory for parametric generalized equations and variational inequalities via nonsmooth analysis*, Trans. Amer. Math. Soc., 343 (1994), pp. 609–657.
[18] G. S. R. MURTHY, T. PARTHASARATHY, AND M. SABATINI, *On Lipschitzian Q-matrices*, Technical report 20, Statistical Quality Control and Operations Research Unit, Indian Statistical Institute, Madras, India, July 1994.
[19] K. G. MURTY, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, Berlin, 1988.
[20] JONG-SHI PANG, *A degree-theoretic approach to parametric nonsmooth equations with multivalued perturbed solution sets*, Math. Programming, 62 (1993), pp. 359–383.
[21] J.-P. PENOT, *Metric regularity, openness and Lipschitz multifunctions*, Nonlinear Anal., 13 (1989), pp. 629–643.
[22] D. RALPH, *A new proof of Robinson's homeomorphism theorem for PL-normal maps*, Linear Algebra Appl., 178 (1993), pp. 249–260.
[23] S. M. ROBINSON, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.
[24] ———, *Local structure of feasible sets in nonlinear programming, Part* II: *Nondegeneracy*, Math. Programming Study, 22 (1984), pp. 217–230.
[25] ———, *An implicit-function theorem for a class of nonsmooth functions*, Math. Oper. Res., 16 (1991), pp. 292–309.
[26] ———, *Normal maps induced by linear transformations*, Math. Oper. Res., 17 (1992), pp. 691–714.
[27] R. T. ROCKAFELLAR, Lipschitz properties of multifunctions, Nonlinear Analysis, 9 (1985), pp. 867-885.
[28] ———, *Proto-differentiability of set-valued mappings and its applications in optimization*, Ann. Inst. H. Poincaré, Anal. Non Linéaire, 6 (1989), suppl. pp. 449–482.
[29] H. SAMELSON, R. M. THRALL, AND O. WESLER, *A partition theorem for Euclidean n-space*, Proc. Amer. Math. Soc., 9 (1958), pp. 805–807.
[30] S. SCHOLTES, *Introduction to piecewise differentiable equations*, Institut für Statistik und Mathematische Wirtschaftstheorie, Universität Karlsruhe, Karlsruhe, Germany, May 1994, preprint 53/1994.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1106–1120, November 1996
© 1996 Society for Industrial and Applied Mathematics
012

A GLOBALLY AND SUPERLINEARLY CONVERGENT ALGORITHM FOR NONSMOOTH CONVEX MINIMIZATION*

MASAO FUKUSHIMA† AND LIQUN QI‡

Abstract. It is well known that a possibly nondifferentiable convex minimization problem can be transformed into a differentiable convex minimization problem by way of the Moreau–Yosida regularization. This paper presents a globally convergent algorithm that is designed to solve the latter problem. Under additional semismoothness and regularity assumptions, the proposed algorithm is shown to have a Q-superlinear rate of convergence.

Key words. nonsmooth convex optimization, Moreau–Yosida regularization, global convergence, superlinear convergence, semismoothness

AMS subject classification. 90C25, 90C30

1. Introduction.

Let $f : \Re^n \to \Re$ be a possibly nondifferentiable convex function and consider the problem

$$\text{minimize} \quad f(x). \tag{1}$$

Associated with this problem is the problem

$$\text{minimize} \quad F(x), \tag{2}$$

where $F : \Re^n \to \Re$ is the so-called Moreau–Yosida regularization of f, which is defined by

$$F(x) = \min_{z \in \Re^n} \left\{ f(z) + \frac{1}{2\lambda} \|z - x\|^2 \right\}, \tag{3}$$

where λ is a positive parameter and $\|\cdot\|$ denotes the Euclidean norm. Throughout the paper we use a fixed parameter λ and we shall not make explicit the dependence of various functions and vectors on λ. It will be simple if we just let $\lambda = 1$. But there are at least two reasons to keep open the possibility that λ can be a constant other than 1. One reason is that an improvement of this algorithm may need to vary λ at each step to get better efficiency. Another reason is stated in the conclusion section of this paper.

It is well known that problems (1) and (2) are equivalent in the sense that the solution sets of the two problems coincide with each other. In fact, the classical proximal point algorithm [8, 22] is regarded as a gradient method for solving problem (2). A remarkable feature of problem (2) is that the objective function F is a differentiable convex function, even though the function f is nondifferentiable. Moreover, F has a Lipschitz continuous gradient. Rockafellar showed that F is twice differentiable at a point if and only if f is twice differentiable in the general sense at a corresponding

* Received by the editors December 19, 1994; accepted for publication (in revised form) August 7, 1995.

† Department of Applied Mathematics and Physics, Graduate School of Engineering, Kyoto University, Kyoto 606-01, Japan (fuku@kuamp.kyoto-u.ac.jp). The research of this author was supported in part by the Scientific Research Grant-in-Aid from the Ministry of Education, Science, and Culture, Japan.

‡ School of Mathematics, The University of New South Wales, Sydney, New South Wales 2052, Australia (l.qi@unsw.edu.au). The research of this author was supported in part by the Australian Research Council.

point relative to a subgradient at that corresponding point [23]. In general, F is not twice differentiable at some kinks and boundary points [23, 6]. However, it was shown recently that under some reasonable conditions, the gradient function of F is semismooth [18]. This paves a way to construct Q-superlinearly convergent algorithms for solving nonsmooth convex minimization problems.

The differentiability of F motivates us to develop a rapidly convergent algorithm for solving (2), particularly when f is nondifferentiable. In practice, however, it is difficult or even impossible to evaluate the functional value $F(x)$ and the gradient $\nabla F(x)$ exactly at any given point x. The purpose of this paper is to present an implementable algorithm that makes use of approximate function and gradient values together with an approximate generalized Hessian of F. The proposed algorithm is therefore considered a globalized approximate Newton method designed to solve problem (2). We prove global convergence and, under additional semismoothness and regularity assumptions, Q-superlinear convergence of the proposed algorithm.

2. Preliminaries. In this section, we review some basic results in convex analysis and nonsmooth analysis.

Since the minimand on the right-hand side of (3) is strongly convex, the minimum is attained uniquely for each $x \in \Re^n$. We denote the unique minimizer by $p(x)$. The results stated in the following propositions are fundamental and useful in the subsequent discussions.

PROPOSITION 2.1. *The function F is finite-valued, convex, and everywhere differentiable with gradient*

$$\nabla F(x) = \frac{1}{\lambda}(x - p(x)).$$

Moreover, the gradient mapping $\nabla F : \Re^n \to \Re^n$ is co-coercive with modulus $1/\lambda$, i.e.,

$$\|\nabla F(x) - \nabla F(x')\|^2 \leq \frac{1}{\lambda}\langle \nabla F(x) - \nabla F(x'), x - x' \rangle \qquad \text{for all } x, x' \in \Re^n,$$

and hence Lipschitz continuous with modulus $1/\lambda$, i.e.,

$$\|\nabla F(x) - \nabla F(x')\| \leq \frac{1}{\lambda}\|x - x'\| \qquad \text{for all } x, x' \in \Re^n.$$

Proof. See [5, Thm. XV.4.1.4]. □

PROPOSITION 2.2. *The following statements are equivalent:*

(i) *x minimizes f;*
(ii) $x = p(x)$;
(iii) $\nabla F(x) = 0$;
(iv) *x minimizes F;*
(v) $f(x) = f(p(x))$;
(vi) $f(x) = F(x)$.

Proof. See [5, Thm. XV.4.1.7]. □

In what follows, we denote the gradient mapping ∇F by g. By Proposition 2.1, the mapping $g : \Re^n \to \Re^n$ is globally Lipschitz continuous. Hence, by Rademacher's theorem, g is differentiable almost everywhere and the set

$$\partial_B g(x) = \left\{ A \in \Re^{n \times n} \mid A = \lim_{x^k \to x} \nabla g(x^k), \ \ x^k \in \Omega_g \right\}, \tag{4}$$

where $\Omega_g = \{x \in \Re^n \mid g \text{ is differentiable at } x\}$, is nonempty and bounded for each x. Notice that for a Lipschitz continuous function, there is no difference between Fréchet differentiability and Gâteaux differentiability. We say that g is BD-regular at x if all matrices $A \in \partial_B g(x)$ are nonsingular. Since g is a gradient mapping of the convex function F, its Jacobian $\nabla g(x)$ is symmetric and positive semidefinite for any $x \in \Omega_g$. The next proposition is a consequence of these observations.

PROPOSITION 2.3. *For each $x \in \Re^n$, every $V \in \partial_B g(x)$ is a symmetric positive semidefinite matrix. Moreover, if g is BD-regular at x, then there exists a constant $c > 0$ and a neighborhood $\mathcal{N}$ of x such that for all $y \in \mathcal{N}$,*

$$\langle d, Vd \rangle \geq c\|d\|^2 \qquad \text{for all} \quad d \in \Re^n \quad \text{and} \quad V \in \partial_B g(y).$$

Proof. See [16, Lem. 2.6]. □

3. Algorithm. Since F is a differentiable convex function with Lipschitz continuous gradient, it is tempting to apply a suitable gradient-based algorithm to solve problem (2). However, by (3) and Proposition 2.1, $F(x)$ and $\nabla F(x)$ are obtained through the optimal solution of another minimization problem

$$\text{minimize}_z \quad f(z) + \frac{1}{2\lambda}\|z - x\|^2, \tag{5}$$

which is difficult or even impossible to solve exactly. In practice, we may only expect to compute an approximation to the optimal solution $p(x)$ of problem (5). Thus we cannot apply an existing gradient-based method directly to problem (2), but need to develop an implementable algorithm that takes into account the inevitable approximate evaluation of the function value and gradient of F.

In the following, we suppose that, for each $x \in \Re^n$, we can evaluate $F(x)$ and $g(x)$ $(= \nabla F(x))$ only approximately but with any desired accuracy. To be more specific, we suppose that, for each $x \in \Re^n$ and any $\varepsilon > 0$, we can find a vector $p^a(x, \varepsilon) \in \Re^n$ such that

$$f(p^a(x,\varepsilon)) + \frac{1}{2\lambda}\|p^a(x,\varepsilon) - x\|^2 \leq F(x) + \varepsilon. \tag{6}$$

Implementable algorithms that are designed to find such $p^a(x, \varepsilon)$ for a nondifferentiable convex function f may be found in [4, 1, 2]. With vector $p^a(x, \varepsilon)$, define approximations to $F(x)$ and $\nabla F(x)$ by

$$F^a(x,\varepsilon) = f(p^a(x,\varepsilon)) + \frac{1}{2\lambda}\|p^a(x,\varepsilon) - x\|^2 \tag{7}$$

and

$$g^a(x,\varepsilon) = \frac{1}{\lambda}(x - p^a(x,\varepsilon)), \tag{8}$$

respectively.

LEMMA 3.1. *Let $p^a(x, \varepsilon)$ be a vector satisfying* (6) *and $F^a(x, \varepsilon)$ and $g^a(x, \varepsilon)$ be given by* (7) *and* (8)*, respectively. Then we have*

$$F(x) \leq F^a(x,\varepsilon) \leq F(x) + \varepsilon, \tag{9}$$

$$\|p^a(x,\varepsilon) - p(x)\| \leq \sqrt{2\lambda\varepsilon}, \tag{10}$$

and

$$\|g^a(x,\varepsilon) - g(x)\| \le \sqrt{\frac{2\varepsilon}{\lambda}}. \tag{11}$$

Proof. First, (9) is obvious from (6) and (7). Now define the function $\theta : \Re^n \to \Re$ by

$$\theta(z) = f(z) + \frac{1}{2\lambda}\|z - x\|^2.$$

It is easy to see that the function θ is strongly convex with modulus $1/\lambda$. Then, by [22, Prop. 6], we have the inequality

$$\theta(y) \ge \theta(z) + \langle y - z, w\rangle + \frac{1}{2\lambda}\|y - z\|^2 \qquad \text{for all } y, z \in \Re^n \tag{12}$$

whenever $w \in \partial\theta(z)$. Let us put $y = p^a(x,\varepsilon)$ and $z = p(x)$ in (12). We have $\theta(y) = \theta(p^a(x,\varepsilon)) = F^a(x,\varepsilon)$ and $\theta(z) = \theta(p(x)) = F(x)$. Moreover, since $p(x)$ minimizes θ, $w = 0$ belongs to $\partial\theta(p(x))$. So it follows from (12) that

$$F^a(x,\varepsilon) \ge F(x) + \frac{1}{2\lambda}\|p^a(x,\varepsilon) - p(x)\|^2,$$

which together with the right-hand inequality in (9) yields (10). Finally, by Proposition 2.1 and (8), we have

$$\|g^a(x,\varepsilon) - g(x)\| = \frac{1}{\lambda}\|p^a(x,\varepsilon) - p(x)\|,$$

which together with (10) gives (11). □

The above lemma says that we can compute approximations $F^a(x,\varepsilon)$ and $g^a(x,\varepsilon)$, which may be made arbitrarily close to $F(x)$ and $g(x)$, respectively, by choosing parameter ε small enough. To design a rapidly convergent algorithm, we have to make use of a "second-order" approximation to F at each x. Recently much effort has been made to extend the Newton method to semismooth equations [11, 20, 16] and SC^1 minimization problems [17, 12, 3]. The basic assumption made in those methods is that, in the context of the problem under consideration, a matrix $V \in \partial_B g(x)$ is available at every current point x. Unfortunately, like the function value $F(x)$ and gradient $g(x)$, an exact generalized Hessian $V \in \partial_B g(x)$ is in general not computable at the current point x. Again, we have to compute it approximately. Note that, when f is actually twice continuously differentiable at $p(x)$, $\partial_B g(x)$ consists of the single element, which is precisely

$$\nabla^2 F(x) = \frac{1}{\lambda} I - \frac{1}{\lambda}\left[I + \lambda\nabla^2 f(p(x))\right]^{-1}. \tag{13}$$

Another important case is that the conjugate function f^* is twice continuously differentiable at $g(x)$. In this case, $\nabla^2 F(x)$ also exists and is given by

$$\nabla^2 F(x) = \left[\lambda I + \nabla^2 f^*(g(x))\right]^{-1}. \tag{14}$$

(See [23, 6, 18] for more details and further results on the second-order differentiability of F.) Since $p^a(x,\varepsilon)$ is available as an approximation to the uncomputable vector $p(x)$, a possible choice is to use the matrix

$$V = \frac{1}{\lambda} I - \frac{1}{\lambda}\left[I + \lambda\nabla^2 f(p^a(x,\varepsilon))\right]^{-1}, \tag{15}$$

if f is twice differentiable at $p^a(x,\varepsilon)$ or

$$V = \left[\lambda I + \nabla^2 f^*(g^a(x,\varepsilon))\right]^{-1}, \tag{16}$$

if f^* is twice differentiable at $g^a(x,\varepsilon)$, as a substitute for the Hessian $\nabla^2 F(x)$ given by (13) and (14), respectively. It is easy to see that matrices V given by (15) and (16) are symmetric and positive definite.

Consider the following example: Let $\lambda = 1$ and let $f : \Re^2 \to \Re$ be defined by

$$f(x) = |x_1| + \frac{1}{2}x_2^2.$$

Then we have

$$F(x) = \begin{cases} x_1 + \frac{1}{4}x_2^2 - \frac{1}{2} & \text{if } x_1 \geq 1, \\ \frac{1}{2}x_1^2 + \frac{1}{4}x_2^2 & \text{if } -1 \leq x_1 < 1, \\ -x_1 + \frac{1}{4}x_2^2 - \frac{1}{2} & \text{if } x_1 < -1, \end{cases}$$

$$g(x) = \begin{cases} (1, \frac{1}{2}x_2)^T & \text{if } x_1 \geq 1, \\ (x_1, \frac{1}{2}x_2)^T & \text{if } -1 \leq x_1 < 1, \\ (-1, \frac{1}{2}x_2)^T & \text{if } x_1 < -1, \end{cases}$$

and

$$p(x) = \begin{cases} (x_1 - 1, \frac{1}{2}x_2)^T & \text{if } x_1 \geq 1, \\ (0, \frac{1}{2}x_2)^T & \text{if } -1 \leq x_1 < 1, \\ (x_1 + 1, \frac{1}{2}x_2)^T & \text{if } x_1 < -1. \end{cases}$$

Now let $\hat{x} = (1,0)^T$. Then it is easy to see that g is not differentiable at $\hat{x}$ and

$$\partial_B g(\hat{x}) = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \right\}.$$

Suppose that, as an approximation to $p(\hat{x}) = (0,0)^T$, we get

$$p^a(\hat{x},\varepsilon) = (\varepsilon_1, \varepsilon_2)^T,$$

where ε_1 and ε_2 are real numbers such that $|\varepsilon_1|$ and $|\varepsilon_2|$ are sufficiently small and $\varepsilon_1 \neq 0$. Then, by (15), we have

$$V = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix},$$

which does not depend on $(\varepsilon_1, \varepsilon_2)$ and is precisely one of the elements of $\partial_B g(\hat{x})$. This example suggests that formula (15) is a viable one. However, in some cases, it is not necessarily true, as the following example reveals: Let $\lambda = 1$ and let $f : \Re^2 \to \Re$ be defined by

$$f(x) = |x_1| + \phi(x_2),$$

where $\phi : \Re \to \Re$ is given by

$$\phi(t) = \begin{cases} t + \frac{1}{2}t^2 & \text{if } t \geq 0, \\ -t + \frac{1}{2}t^2 & \text{if } t < 0. \end{cases}$$

Then we have $F(x) = F_1(x_1) + F_2(x_2)$ with

$$F_1(x_1) = \begin{cases} x_1 - \frac{1}{2} & \text{if } x_1 \geq 1, \\ \frac{1}{2}x_1^2 & \text{if } -1 \leq x_1 < 1, \\ -x_1 - \frac{1}{2} & \text{if } x_1 < -1, \end{cases}$$

$$F_2(x_2) = \begin{cases} \frac{1}{4}(x_2^2 + 2x_2 - 1) & \text{if } x_2 \geq 1, \\ \frac{1}{2}x_2^2 & \text{if } -1 \leq x_2 < 1, \\ \frac{1}{4}(x_2^2 - 2x_2 - 1) & \text{if } x_2 < -1 \end{cases}$$

and $g(x) = (g_1(x_1), g_2(x_2))^T$ with

$$g_1(x_1) = \begin{cases} 1 & \text{if } x_1 \geq 1, \\ x_1 & \text{if } -1 \leq x_1 < 1, \\ -1 & \text{if } x_1 < -1, \end{cases}$$

$$g_2(x_2) = \begin{cases} \frac{1}{2}(x_2 + 1) & \text{if } x_2 \geq 1, \\ x_2 & \text{if } -1 \leq x_2 < 1, \\ \frac{1}{2}(x_2 - 1) & \text{if } x_2 < -1. \end{cases}$$

Also we have $p(x) = (p_1(x_1), p_2(x_2))^T$ with

$$p_1(x_1) = \begin{cases} x_1 - 1 & \text{if } x_1 \geq 1, \\ 0 & \text{if } -1 \leq x_1 < 1, \\ x_1 + 1 & \text{if } x_1 < -1, \end{cases}$$

$$p_2(x_2) = \begin{cases} \frac{1}{2}(x_2 - 1) & \text{if } x_2 \geq 1, \\ 0 & \text{if } -1 \leq x_2 < 1, \\ \frac{1}{2}(x_2 + 1) & \text{if } x_2 < -1. \end{cases}$$

Now let $\hat{x} = (1, 0)^T$. Then we obtain

$$\partial_B g(\hat{x}) = \left\{ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}.$$

Again suppose that, as an approximation to $p(\hat{x}) = (0, 0)^T$, we have

$$p^a(\hat{x}, \varepsilon) = (\varepsilon_1, \varepsilon_2)^T,$$

where ε_1 and ε_2 are, as before, real numbers such that $|\varepsilon_1|$ and $|\varepsilon_2|$ are sufficiently small. Then, by (15), we have

$$V = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

for any $(\varepsilon_1, \varepsilon_2)$ with $\varepsilon_1 \neq 0$ and $\varepsilon_2 \neq 0$. This time we see that the last matrix V is close to none of the elements of $\partial_B g(\hat{x})$. This example indicates that formula (15) can yield a bad approximation to any element of $\partial_B g(x)$. The situation can be more complicated if f is not separable. In the conclusion section, we will discuss three possible approaches toward a better choice of V.

Now we state the algorithm.

ALGORITHM.

Step 0 (Initialization). Let $\lambda > 0$, $\sigma \in (0,1)$, and $\rho \in (0,1)$ be given constants. Choose $x^0 \in \Re^n$ and $\varepsilon_0 > 0$. Set $k := 0$.

Step 1 (Search direction). Compute $g^a(x^k, \varepsilon_k)$ and pick a positive-definite symmetric $n \times n$ matrix V_k. Compute the search direction

$$d^k := -V_k^{-1} g^a(x^k, \varepsilon_k).$$

Step 2 (Line search). Choose a scalar ε_{k+1} such that $0 < \varepsilon_{k+1} < \varepsilon_k$. Let m_k be the smallest nonnegative integer m such that

$$F^a(x^k + \rho^m d^k, \varepsilon_{k+1}) \le F^a(x^k, \varepsilon_k) + \sigma \rho^m \langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_k.$$

Set $\tau_k := \rho^{m_k}$ and $x^{k+1} := x^k + \tau_k d^k$. Let $k := k+1$ and return to Step 1. □

In Step 1, there are some possibilities in choosing matrix V_k. For example, from the above arguments, we may set

$$V_k := \begin{cases} \left(\frac{1}{\lambda} + \alpha_k\right) I - \frac{1}{\lambda}\left[I + \lambda \nabla^2 f(p^a(x^k, \varepsilon_k))\right]^{-1} & \text{if } \nabla^2 f(p^a(x^k, \varepsilon_k)) \text{ exists,} \\ \alpha_k I + \left[\lambda I + \nabla^2 f^*(g^a(x^k, \varepsilon_k))\right]^{-1} & \text{if } \nabla^2 f^*(g^a(x^k, \varepsilon_k)) \text{ exists,} \\ V_{k-1} & \text{otherwise,} \end{cases} \tag{17}$$

with $V_{-1} = I$, where α_k is a small constant to ensure that V_k is positive definite. For more discussions on the choice of V_k, see the conclusion section.

Notice that by Lemma 3.1, we have

$$F^a(x^k + \tau d^k, \varepsilon_{k+1}) \le F(x^k + \tau d^k) + \varepsilon_{k+1}, \tag{18}$$

$$F(x^k) \le F^a(x^k, \varepsilon_k), \tag{19}$$

and

$$\begin{aligned} \langle \nabla F(x^k), d^k \rangle - \langle g^a(x^k, \varepsilon_k), d^k \rangle &= \langle \nabla F(x^k) - g^a(x^k, \varepsilon_k), d^k \rangle \\ &\le \|\nabla F(x^k) - g^a(x^k, \varepsilon_k)\| \, \|d^k\| \\ &\le \sqrt{\frac{2\varepsilon_k}{\lambda}} \|d^k\|. \end{aligned} \tag{20}$$

The next proposition ensures that, at each iteration k of the algorithm, m_k is well defined and hence the step size $\tau_k > 0$ can be determined finitely in Step 2.

PROPOSITION 3.2. *For every k, there exists $\bar{\tau}_k > 0$ such that*

$$F^a(x^k + \tau d^k, \varepsilon_{k+1}) \le F^a(x^k, \varepsilon_k) + \sigma \tau \langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_k \tag{21}$$

for all $\tau \in (0, \bar{\tau}_k)$.

Proof. Adding (18), (19), and (20) multiplied by τ and rearranging terms, we obtain

$$\begin{aligned} F^a(x^k + \tau d^k, \varepsilon_{k+1}) \le F^a(x^k, \varepsilon_k) &+ \tau \langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_{k+1} + \tau \sqrt{\frac{2\varepsilon_k}{\lambda}} \|d^k\| \\ &+ F(x^k + \tau d^k) - F(x^k) - \tau \langle \nabla F(x^k), d^k \rangle. \end{aligned} \tag{22}$$

If $d^k = 0$, then (22) implies that (21) holds for any $\tau > 0$ because $\varepsilon_{k+1} < \varepsilon_k$. Consider the case $d^k \neq 0$. Then, since V_k is positive definite, we have

$$\langle g^a(x^k, \varepsilon_k), d^k \rangle = -\langle V_k d^k, d^k \rangle < 0,$$

so that

$$\langle g^a(x^k, \varepsilon_k), d^k \rangle < \sigma \langle g^a(x^k, \varepsilon_k), d^k \rangle. \tag{23}$$

Moreover, since $\varepsilon_{k+1} < \varepsilon_k$, there exists $\bar{\tau}_k > 0$ such that

$$F(x^k + \tau d^k) - F(x^k) - \tau \langle \nabla F(x^k), d^k \rangle + \tau \sqrt{\frac{2\varepsilon_k}{\lambda}} \|d^k\| + \varepsilon_{k+1} < \varepsilon_k \tag{24}$$

for all $\tau \in (0, \bar{\tau}_k)$. That (21) holds for all $\tau \in (0, \bar{\tau}_k)$ now follows from (22)–(24). □

4. Global convergence. The following theorem establishes global convergence of the algorithm.

THEOREM 4.1. *Assume that the objective function f of problem* (1) *is bounded from below. Suppose that there exists a constant $\gamma > 0$ such that*

$$\langle V_k d, d \rangle \geq \gamma \|d\|^2 \qquad \text{for all } d \in \Re^n \text{ and all } k. \tag{25}$$

If the sequence $\{\varepsilon_k\}$ of strictly decreasing positive numbers satisfies the condition $\sum_{k=0}^{\infty} \varepsilon_k < +\infty$, then any accumulation point of $\{x^k\}$ is an optimal solution of problem (1).

Proof. By the line search rule, it holds that

$$F^a(x^{k+1}, \varepsilon_{k+1}) \leq F^a(x^k, \varepsilon_k) + \sigma \tau_k \langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_k \tag{26}$$

for all k. Since $\langle g^a(x^k, \varepsilon_k), d^k \rangle \leq 0$, it follows from (26) that

$$F^a(x^{k+1}, \varepsilon_{k+1}) \leq F^a(x^k, \varepsilon_k) + \varepsilon_k, \tag{27}$$

and hence

$$F^a(x^k, \varepsilon_k) \leq F^a(x^0, \varepsilon_0) + \sum_{i=0}^{k-1} \varepsilon_i,$$

which together with the assumption $\sum_{k=0}^{\infty} \varepsilon_k < +\infty$ implies that $\{F^a(x^k, \varepsilon_k)\}$ is bounded from above. On the other hand, f is bounded from below by assumption, and hence F is also bounded from below. Since $F^a(x^k, \varepsilon_k) \geq F(x^k)$ for all k, the sequence $\{F^a(x^k, \varepsilon_k)\}$ is bounded from below. Therefore the sequence $\{F^a(x^k, \varepsilon_k)\}$ has at least one accumulation point. In fact, it can be shown that the whole sequence $\{F^a(x^k, \varepsilon_k)\}$ is convergent. To see this, let $\{F^a(x^k, \varepsilon_k)\}$ have two accumulation points F_1^* and F_2^* such that $F_1^* < F_2^*$ and let $\{F^a(x^k, \varepsilon_k)\}_{k \in \kappa_1}$ and $\{F^a(x^k, \varepsilon_k)\}_{k \in \kappa_2}$ be subsequences converging to F_1^* and F_2^*, respectively. Put $\hat{\varepsilon} = (F_2^* - F_1^*)/3$. Then there exist integers $\bar{k}_1, \bar{k}_2 > 0$ such that $F^a(x^k, \varepsilon_k) < F_1^* + \hat{\varepsilon}$ for all $k \in \kappa_1$ with $k > \bar{k}_1$ and $F^a(x^k, \varepsilon_k) > F_2^* - \hat{\varepsilon}$ for all $k \in \kappa_2$ with $k > \bar{k}_2$. Moreover, since $\sum_{k=0}^{\infty} \varepsilon_k < +\infty$, there exists an integer $\bar{k}_3 > 0$ such that $\sum_{k=\bar{k}_3}^{\infty} \varepsilon_k < \hat{\varepsilon}$. Pick an integer $\bar{k} \in \kappa_1$ such that $\bar{k} \geq \max\{\bar{k}_1, \bar{k}_2, \bar{k}_3\}$. Then, by (27), we have $F^a(x^k, \varepsilon_k) < F^a(x^{\bar{k}}, \varepsilon_{\bar{k}}) + \sum_{i=\bar{k}}^{k-1} \varepsilon_i < F_1^* + 2\hat{\varepsilon} = F_2^* - \hat{\varepsilon}$ for all $k > \bar{k}$. This is a contradiction, and hence $\{F^a(x^k, \varepsilon_k)\}$ has at most one accumulation point.

Then, since ε_k tends to zero and since $\langle g^a(x^k, \varepsilon_k), d^k\rangle \leq 0$, it follows from (26) that

$$\lim_{k\to\infty} \tau_k \langle g^a(x^k, \varepsilon_k), d^k\rangle = 0. \tag{28}$$

Now let $\bar{x}$ be an arbitrary accumulation point of $\{x^k\}$ and let $\{x^k\}_{k\in\kappa}$ be a subsequence converging to $\bar{x}$. Then by Lemma 3.1, we have

$$\lim_{k\to\infty, k\in\kappa} g^a(x^k, \varepsilon_k) = g(\bar{x}). \tag{29}$$

Since $\{V_k^{-1}\}$ is bounded, we may suppose by taking a further subsequence if necessary that

$$\lim_{k\to\infty, k\in\kappa} d^k = \bar{d} \tag{30}$$

for some vector $\bar{d} \in \Re^n$. Moreover, under the given assumption, we have

$$\lim_{k\to\infty, k\in\kappa} \langle g^a(x^k, \varepsilon_k), d^k\rangle = \langle g(\bar{x}), \bar{d}\rangle \leq -\gamma\|\bar{d}\|^2. \tag{31}$$

If $\liminf_{k\to\infty, k\in\kappa} \tau_k > 0$, then (28) and (31) immediately yield

$$\bar{d} = 0. \tag{32}$$

On the other hand, if $\liminf_{k\to\infty, k\in\kappa} \tau_k = 0$, then again by taking a further subsequence if necessary we may suppose that $\tau_k \to 0$ on the subsequence κ. The definition of m_k in the line search rule gives

$$F^a(x^k + \rho^{m_k-1}d^k, \varepsilon_{k+1}) > F^a(x^k, \varepsilon_k) + \sigma\rho^{m_k-1}\langle g^a(x^k, \varepsilon_k), d^k\rangle + \varepsilon_k. \tag{33}$$

By Lemma 3.1, we have

$$F^a(x^k, \varepsilon_k) \geq F(x^k),$$
$$F^a(x^k + \rho^{m_k-1}d^k, \varepsilon_{k+1}) \leq F(x^k + \rho^{m_k-1}d^k) + \varepsilon_{k+1}.$$

Combining these inequalities with (33) and taking $\varepsilon_k > \varepsilon_{k+1}$ into account yield

$$F(x^k + \rho^{m_k-1}d^k) > F(x^k) + \sigma\rho^{m_k-1}\langle g^a(x^k, \varepsilon_k), d^k\rangle;$$

that is,

$$\frac{F(x^k + \rho^{m_k-1}d^k) - F(x^k)}{\rho^{m_k-1}} > \sigma\langle g^a(x^k, \varepsilon_k), d^k\rangle. \tag{34}$$

Since $\rho^{m_k-1} = \tau_k/\rho \to 0$, we have

$$\lim_{k\to\infty, k\in\kappa} \frac{F(x^k + \rho^{m_k-1}d^k) - F(x^k)}{\rho^{m_k-1}} = \langle g(\bar{x}), \bar{d}\rangle.$$

So, by taking the limit in (34) on the subsequence $k \in \kappa$, we obtain

$$\langle g(\bar{x}), \bar{d}\rangle \geq \sigma\langle g(\bar{x}), \bar{d}\rangle.$$

In view of (31), the last inequality also gives

$$\bar{d} = 0.$$

Since $g^a(x^k, \varepsilon_k) = -V_k d^k$ and since $\{V_k\}$ is bounded, it follows from $\bar{d} = 0$ that

$$\lim_{k \to \infty, k \in \kappa} g^a(x^k, \varepsilon_k) = g(\bar{x}) = 0;$$

namely,

$$\nabla F(\bar{x}) = 0.$$

Therefore, by Proposition 2.2, $\bar{x}$ is a solution of problem (1). □

5. Sequential and superlinear convergence. Our next goal is to establish sequential and Q-superlinear convergence of the algorithm.

THEOREM 5.1. *Suppose that the conditions of Theorem* 4.1 *hold and that* $\bar{x}$ *is an optimal solution of* (1). *Assume that* g *is BD-regular at* $\bar{x}$. *Then* $\bar{x}$ *is the unique optimal solution of* (1) *and the entire sequence* $\{x^k\}$ *converges to* $\bar{x}$.

Proof. By the BD-regularity of g at $\bar{x}$ and the convexity of F, $\bar{x}$ is the unique optimal solution of (2); for a proof, see Theorem 2.2 of [21]. By Proposition 2.2, $\bar{x}$ is also the unique optimal solution of (1). This implies that both f and F must have compact level sets. By the line search rule and (9),

$$F(x^{k+1}) \leq F(x^k) + \sigma \tau_k \langle g^a(x^k, \varepsilon_k), d^k \rangle + 2\varepsilon_k.$$

Since $\langle g^a(x^k, \varepsilon_k), d^k \rangle \leq 0$, we have

$$F(x^{k+1}) \leq F(x^k) + 2\varepsilon_k$$

and hence

$$F(x^k) \leq F(x^0) + 2 \sum_{i=0}^{k-1} \varepsilon_i,$$

which together with the assumption $\sum_{k=0}^{\infty} \varepsilon_k < +\infty$ implies that the sequence $\{F(x^k)\}$ is bounded from above. Since the level sets of F are compact, this implies that $\{x^k\}$ has at least one accumulation point. By Theorem 4.1, this accumulation point must be $\bar{x}$ since $\bar{x}$ is the unique solution of (1).

Let $\delta > 0$. By Theorem 4.1, there are only a finite number of k's such that

$$\delta \leq \|x^k - \bar{x}\| \leq 2\delta.$$

Otherwise, there will be an accumulation point $\hat{x}$ of $\{x^k\}$ such that

$$\delta \leq \|\hat{x} - \bar{x}\| \leq 2\delta.$$

Then $\hat{x}$ must be an optimal solution of (1). This contradicts the fact that $\bar{x}$ is the unique optimal solution of (1). Hence, there is a $k_0 > 0$ such that if $k \geq k_0$, either

$$\|x^k - \bar{x}\| \leq \delta$$

or

$$\|x^k - \bar{x}\| \geq 2\delta.$$

Let $\kappa_1 = \{k \mid \|x^k - \bar{x}\| \leq \delta, k \geq k_0\}$, $\kappa_2 = \{k \mid \|x^k - \bar{x}\| \geq 2\delta, k \geq k_0\}$, $\kappa_3 = \{k \mid k \in \kappa_1, k+1 \in \kappa_2\}$. Then, if $k \in \kappa_3$, we have

$$\|d^k\| \geq \|x^k - x^{k+1}\| \geq \delta. \tag{35}$$

Assume κ_2 is infinite. Then κ_3 is also infinite. By the same argument as above, $\{x^k \mid k \in \kappa_1\}$ cannot have an accumulation point other than $\bar{x}$. Hence, $\{x^k \mid k \in \kappa_1\}$ converges to $\bar{x}$. By the proof of Theorem 4.1, at least for a subsequence κ_4 of κ_3,

$$\lim_{k \to \infty, k \in \kappa_4} d^k = 0,$$

which contradicts (35). Hence, κ_2 is finite. This proves that the entire sequence $\{x^k\}$ converges to $\bar{x}$. □

The next theorem requires that g be semismooth at the unique solution $\bar{x}$. In [18], some cases in which this condition holds were identified. It was proven there that if f is the maximum of several affine functions, then g is a semismooth function, and conjectured there that this would be true if f were the maximum of several twice continuously differentiable convex functions. Sun and Han [24] proved this conjecture under the assumption that f satisfies the constant rank constraint qualification. Mifflin, Qi, and Sun [10] further extended that result by proving that g is semismooth under a mild condition if f is a piecewise twice continuously differentiable convex function.

THEOREM 5.2. *Suppose that the conditions of Theorem* 5.1 *hold and that g is semismooth at the unique solution $\bar{x}$ to* (1). *Suppose furthermore that*

(i) $\varepsilon_k = o(\|g(x^k)\|^2)$;

(ii) $\lim_{k \to \infty} dist(V_k, \partial_B g(x^k)) = 0$;

(iii) $\tau_k \equiv 1$ *for all large* k.

Then $\{x^k\}$ converges to $\bar{x}$ Q-superlinearly.

Proof. First note that, by Theorem 5.1, the sequence $\{x^k\}$ converges to $\bar{x}$. Then, by condition (i), inequality (11), and Propositions 2.1 and 2.2, we have

$$\begin{aligned} \|g^a(x^k, \varepsilon_k) - g(x^k)\| &= O(\sqrt{\varepsilon_k}) \\ &= o(\|g(x^k)\|) \\ &= o(\|g(x^k) - g(\bar{x})\|) \\ &= o(\|x^k - \bar{x}\|). \end{aligned} \tag{36}$$

By condition (ii), there is a $\bar{V}_k \in \partial_B g(x^k)$ such that

$$\|V_k - \bar{V}_k\| = o(1). \tag{37}$$

Since g is semismooth at $\bar{x}$, according to [16, 20], we have

$$\|g(x^k) - g(\bar{x}) - \bar{V}_k(x^k - \bar{x})\| = o(\|x^k - \bar{x}\|). \tag{38}$$

Notice that $\|V_k^{-1}\| = O(1)$. By the algorithm, (36)–(38), and condition (iii), for all large k, we have

$$\begin{aligned} \|x^{k+1} - \bar{x}\| &= \|x^k - \bar{x} - V_k^{-1} g^a(x^k, \varepsilon_k)\| \\ &\leq \|V_k^{-1}\| \cdot \{\|g^a(x^k, \varepsilon_k) - g(x^k)\| + \|g(x^k) - g(\bar{x}) - \bar{V}_k(x^k - \bar{x})\| \\ &\quad + \|V_k - \bar{V}_k\| \cdot \|x^k - \bar{x}\|\} \\ &= o(\|x^k - \bar{x}\|). \end{aligned}$$

This establishes Q-superlinear convergence of $\{x^k\}$ to $\bar{x}$. □

In the above theorem, condition (i) might seem somewhat problematic since, in the algorithm, ε_k is chosen before x^k is determined. The next result shows that we may still ensure fast convergence even if condition (i) is replaced by a more realistic condition.

COROLLARY 5.3. *Suppose that the conditions of Theorem* 5.2 *hold, except that condition* (i) *is replaced by* (i′) $\varepsilon_k = o(\|g(x^{k-1})\|^2)$. *Then* $\{x^k\}$ *converges to* $\bar{x}$ *at least* 2*-step Q-superlinearly.*

Proof. Under condition (i′), we have the estimate

$$\|g^a(x^k, \varepsilon_k) - g(x^k)\| = o(\|x^{k-1} - \bar{x}\|),$$

instead of (36). Then an argument similar to the proof of Theorem 5.2 yields

$$\|x^{k+1} - \bar{x}\| = o(\max\{\|x^{k-1} - \bar{x}\|, \|x^k - \bar{x}\|\}).$$

That is, the rate of convergence is at least 2-step Q-superlinear. □

We now prove that condition (iii) in Theorem 5.2 will be satisfied automatically; i.e., the Maratos effect will not occur.

THEOREM 5.4. *Suppose that the conditions of Theorem* 5.2, *except condition* (iii), *hold and that* $\sigma < \frac{1}{2}$. *Then condition* (iii) *holds and* $\{x^k\}$ *converges to the unique solution* $\bar{x}$ *to* (1) *Q-superlinearly.*

Proof. Theorem 5.2 shows that

$$\|x^k + d^k - \bar{x}\| = o(\|x^k - \bar{x}\|), \tag{39}$$

which implies

$$\|d^k\| = \|x^k - \bar{x}\| + o(\|x^k - \bar{x}\|). \tag{40}$$

By Theorem 1 of [12], we have

$$\begin{aligned} F(x^k + d^k) &= F(\bar{x}) + \langle g(\bar{x}), x^k + d^k - \bar{x}\rangle \\ &\quad + \frac{1}{2}\langle \hat{V}_k(x^k + d^k - \bar{x}), x^k + d^k - \bar{x}\rangle + o(\|x^k + d^k - \bar{x}\|^2) \\ &= F(\bar{x}) + o(\|x^k - \bar{x}\|^2), \end{aligned} \tag{41}$$

where $\hat{V}_k \in \partial_B g(x^k + d^k)$ and the last equality follows from (39), the local boundedness of $\partial_B g$ at $\bar{x}$, and $g(\bar{x}) = 0$. Similarly,

$$\begin{aligned} F(x^k) &= F(\bar{x}) + \langle g(\bar{x}), x^k - \bar{x}\rangle + \frac{1}{2}\langle \bar{V}_k(x^k - \bar{x}), x^k - \bar{x}\rangle + o(\|x^k - \bar{x}\|^2) \\ &= F(\bar{x}) + \frac{1}{2}\langle \bar{V}_k(x^k - \bar{x}), x^k - \bar{x}\rangle + o(\|x^k - \bar{x}\|^2), \end{aligned} \tag{42}$$

where $\bar{V}_k \in \partial_B g(x^k)$. By (39), (41), and (42), we have

$$\begin{aligned} &F(x^k + d^k) - F(x^k) - \frac{1}{2}\langle g(x^k), d^k\rangle \\ &\quad = -\frac{1}{2}\langle \bar{V}_k(x^k - \bar{x}), x^k - \bar{x}\rangle - \frac{1}{2}\langle g(x^k), d^k\rangle + o(\|x^k - \bar{x}\|^2) \\ &\quad = -\frac{1}{2}\langle g(x^k) - g(\bar{x}) - \bar{V}_k(x^k - \bar{x}), d^k\rangle \\ &\qquad -\frac{1}{2}\langle \bar{V}_k(x^k - \bar{x}), x^k + d^k - \bar{x}\rangle + o(\|x^k - \bar{x}\|^2). \end{aligned} \tag{43}$$

By (38) and (40),

$$\langle g(x^k) - g(\bar{x}) - \bar{V}_k(x^k - \bar{x}), d^k \rangle = o(\|x^k - \bar{x}\|^2), \tag{44}$$

while by (39) and the local boundedness of $\partial_B g$ at $\bar{x}$,

$$\langle \bar{V}_k(x^k - \bar{x}), x^k + d^k - \bar{x} \rangle = o(\|x^k - \bar{x}\|^2). \tag{45}$$

Combining (43)–(45), we have

$$F(x^k + d^k) - F(x^k) - \frac{1}{2}\langle g(x^k), d^k \rangle = o(\|x^k - \bar{x}\|^2). \tag{46}$$

By (20), condition (i) of Theorem 5.2, and (40),

$$\begin{aligned} \frac{1}{2}\langle g(x^k) - g^a(x^k, \varepsilon_k), d^k \rangle &\le \sqrt{\frac{\varepsilon_k}{2\lambda}} \|d^k\| \\ &= o(\|g(x^k)\|)\|x^k - \bar{x}\| \\ &= o(\|g(x^k) - g(\bar{x})\|)\|x^k - \bar{x}\| \\ &= o(\|x^k - \bar{x}\|^2), \end{aligned} \tag{47}$$

where the last equality follows from Proposition 2.1.

Adding (47), (46), (19), and (18) with $\tau = 1$, we obtain

$$\begin{aligned} F^a(x^k + d^k, \varepsilon_{k+1}) &\le F^a(x^k, \varepsilon_k) + \sigma\langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_{k+1} \\ &\quad + \left(\frac{1}{2} - \sigma\right) \langle g^a(x^k, \varepsilon_k), d^k \rangle + o(\|x^k - \bar{x}\|^2) \\ &\le F^a(x^k, \varepsilon_k) + \sigma\langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_k \\ &\quad - \left(\frac{1}{2} - \sigma\right) \langle V_k d^k, d^k \rangle + o(\|x^k - \bar{x}\|^2). \end{aligned}$$

However, by (25) and (40),

$$\begin{aligned} -\left(\frac{1}{2} - \sigma\right) \langle V_k d^k, d^k \rangle + o(\|x^k - \bar{x}\|^2) &\le -\gamma \left(\frac{1}{2} - \sigma\right) \|x^k - \bar{x}\|^2 + o(\|x^k - \bar{x}\|^2) \\ &< 0 \end{aligned}$$

for all large k. Hence, for all large k,

$$F^a(x^k + d^k, \varepsilon_{k+1}) \le F^a(x^k, \varepsilon_k) + \sigma\langle g^a(x^k, \varepsilon_k), d^k \rangle + \varepsilon_k.$$

Therefore, $m_k \equiv 0$; i.e., $\tau_k \equiv 1$ in Step 2 of the algorithm for all large k. Q-superlinear convergence of $\{x^k\}$ to $\bar{x}$ now follows from Theorem 5.2. This proves the theorem. □

6. Conclusion. We have proposed an algorithm for solving a convex minimization problem via the Moreau–Yosida regularization. We have shown that the algorithm is globally convergent and, under some additional conditions, the convergence rate of the outer iterations of this algorithm is superlinear. We note that the algorithm inherently involves the inner iteration for finding a vector $p^a(x, \varepsilon)$ satisfying (6) for given x and ε. There are a number of iterative methods [4, 1, 2] that can be used as a procedure to obtain $p^a(x, \varepsilon)$. In general, those methods tend to spend more

(inner) iterations as the outer iteration proceeds. More efforts are needed to improve the efficiency of the inner iterations.

Superlinear convergence of the outer iterations of an algorithm for solving a nonsmooth convex optimization problem has appeared before. For example, the proximal point algorithm discussed in [22] also has this property but it requires the parameter $\lambda = \lambda_k \to \infty$ for superlinear convergence of the outer iterations in [22], while λ can be fixed here. Actually, it will be fine if we let $\lambda = 1$ for the convex case considered in this paper. We may also think to extend the algorithm to the nonconvex case. It is known that the Moreau–Yosida regularizations of some nonconvex functions also have nice properties. One example is the lower C^2 function (strongly subsmooth function) [23]. Recently, Poliquin and Rockafellar [13]–[15] considered the properties of the Moreau–Yosida regularization of a broader function class, which they called prox-regular functions. Convex functions and lower C^2 functions are all special cases of prox-regular functions. It will be interesting to see whether the algorithm discussed in this paper can be extended to some nondifferentiable nonconvex function classes, such as lower C^2 functions or prox-regular functions. The theoretical discussions in [23, 13]–[15] inform us that for those nonconvex functions to have nice properties of the Moreau–Yosida regularization, the parameter λ has to be in an interval $(0, \lambda_0)$. It will be difficult to extend those superlinear convergence results for outer iterations to these nonconvex cases if it requires $\lambda = \lambda_k \to \infty$. In this sense, the algorithm discussed here has the potential to be extended to these nonconvex cases. To keep this potential open, we keep the parameter λ fixed in the algorithm in this paper.

In the algorithm, the choice of V_k is not specified explicitly, but it is certainly a very important ingredient of the algorithm. Formula (17) suggested in §3 is one of the possible choices of V_k. Although we may expect that V_k determined by (17) provides a reasonable approximation to an element of $\partial_B g(x^k)$ for some cases, it is also true that, in some cases, such matrices V_k may be far from what we should approximate. The cause for such an ill-conditioning phenomenon is that although $p^a(x, \varepsilon)$ can be made arbitrarily close to $p(x)$, it does not mean that there is a y such that $p(y) = p^a(x, \varepsilon)$ and y is arbitrarily close to x. There are three possible approaches to overcome this ill-conditioning phenomenon. The first approach was recently explored by Mifflin. In [9], he considered using a convex combination of Hessians of f, instead of a single one, as well as the decomposition of the space. This may smooth the approximation to $\partial_B g(x^k)$. The second approach is strongly related to the recent work of Lemaréchal and Sagastizábal [6, 7], in which they considered the kinky case of f and decomposed the space into two complementary subspaces associated with the normal cone and the tangent cone of the subdifferential of f at the proximal point. More second-order information may be extracted from f by this decomposition. We hope that a better choice of V_k can be obtained by further study of this approach. The third approach is currently investigated in [19]. In [19], $\frac{1}{\lambda} I$ is replaced by a positive-definite symmetric matrix M_k at each step and M_k is preconditioned in such a way that $p^a(x^k, \varepsilon_k)$ is exactly $p_{M_k}(x^k)$. This eliminates the source of the ill-conditioning. Since separate papers are needed to discuss these approaches, we leave these as further research topics.

Acknowledgments. We are grateful to C. Lemaréchal, R. Mifflin, S. Robinson, and two anonymous referees for their helpful comments.

REFERENCES

[1] A. AUSLENDER, *Numerical methods for nondifferentiable convex optimization*, Math. Programming Study, 30 (1987), pp. 102–126.

[2] R. CORREA AND C. LEMARÉCHAL, *Convergence of some algorithms for convex minimization*, Math. Programming, 62 (1993), pp. 261–275.

[3] F. FACCHINEI, *Minimization of SC^1 functions and the Maratos effect*, Oper. Res. Lett., 17 (1995), pp. 131–137.

[4] M. FUKUSHIMA, *A descent algorithm for nonsmooth convex optimization*, Math. Programming, 30 (1984), pp. 163–175.

[5] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms* II, Springer-Verlag, Berlin, Heidelberg, 1993.

[6] C. LEMARÉCHAL AND C. SAGASTIZÁBAL, *Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries*, SIAM J. Optim., to appear.

[7] ———, *More than first-order developments of convex functions: Primal-dual relations*, Preprint, INRIA, Le Chesnay, France, 1995.

[8] B. MARTINET, *Régularisation d'inéquations variationelles par approximations successives*, Rev. Française d'Aut. Inf. Rech. Opér., 4 (1970), pp. 154–159.

[9] R. MIFFLIN, *A quasi-second-order proximal bundle algorithm*, Math. Programming, 73 (1996), pp. 51–72.

[10] R. MIFFLIN, L. QI, AND D. SUN, *Properties of the Moreau–Yosida regularization of a piecewise C^2 convex function*, Applied Mathematics Report AMR95/27, School of Mathematics, The University of New South Wales, Sydney, Australia, 1995.

[11] J.-S. PANG AND L. QI, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[12] ———, *A globally convergent Newton method for convex SC^1 minimization problems*, J. Optim. Theory Appl., 85 (1995), pp. 633–648.

[13] R. A. POLIQUIN AND R. T. ROCKAFELLAR, *Prox-Regular Functions in Variational Analysis*, Preprint, Department of Mathematical Sciences, University of Alberta, Edmonton, Alberta, Canada, 1994.

[14] ———, *Second-order nonsmooth analysis in nonlinear programming*, in Recent Advances in Nonsmooth Optimization, D. Z. Du, L. Qi, and R. S. Womersley, eds., World Scientific, Singapore, 1995, pp. 322–350.

[15] ———, *Generalized Hessian properties of regularized nonsmooth functions*, SIAM J. Optim., 6 (1996), pp. 1121–1137.

[16] L. QI, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[17] ———, *Superlinearly convergent approximate Newton methods for LC^1 optimization problems*, Math. Programming, 64 (1994), pp. 277–294.

[18] ———, *Second-Order Analysis of the Moreau–Yosida Approximation of a Convex Function*, Applied Mathematics Report AMR94/20, School of Mathematics, The University of New South Wales, Sydney, Australia, 1994.

[19] L. QI AND X. CHEN, *A preconditioning proximal Newton method for nondifferentiable convex optimization*, Math. Programming, to appear.

[20] L. QI AND J. SUN, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.

[21] L. QI AND R. S. WOMERSLEY, *An SQP algorithm for extended linear problems in stochastic programming*, Ann. Oper. Res., 56 (1995), pp. 251–285.

[22] R. T. ROCKAFELLAR, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[23] ———, *Maximal monotone relations and the second derivatives of nonsmooth functions*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 2 (1985), pp. 167–184.

[24] D. SUN AND J. HAN, *On a conjecture in Moreau–Yosida regularization of a nonsmooth convex function*, Bull. Sci. China, to appear.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1121–1137, November 1996
© 1996 Society for Industrial and Applied Mathematics
013

GENERALIZED HESSIAN PROPERTIES OF REGULARIZED NONSMOOTH FUNCTIONS*

R. A. POLIQUIN† AND R. T. ROCKAFELLAR‡

Abstract. We take up the question of second-order expansions for a class of functions of importance in optimization, namely, Moreau envelope regularizations of nonsmooth functions f. It is shown that when f is prox-regular, which includes convex functions and the extended real-valued functions representing problems of nonlinear programming, the many second-order properties that can be formulated around the existence and stability of expansions of the envelopes of f or of their gradient mappings are linked by surprisingly extensive lists of equivalences with each other and with generalized differentiation properties of f itself. This clarifies the circumstances conducive to developing computational methods based on envelope functions, such as second-order approximations in nonsmooth optimization and variants of the proximal point algorithm. The results establish that generalized second-order expansions of Moreau envelopes, at least, can be counted on in most situations of interest in finite-dimensional optimization.

Key words. prox-regularity, amenable functions, primal-lower-nice functions, Hessians, first- and second-order expansions, strict proto-derivatives, proximal mappings, Moreau envelopes, regularization, subgradient mappings, nonsmooth analysis, variational analysis, proto-derivatives, second-order epi-derivatives, Attouch's theorem

AMS subject classifications. Primary, 49A52, 58C06, 58C20; Secondary, 90C30

1. Introduction.

Any problem of optimization in $\mathbb{R}^n$ can be posed abstractly as one of minimizing a function $f : \mathbb{R}^n \to \overline{\mathbb{R}} := \mathbb{R} \cup \pm\infty$ over all of $\mathbb{R}^n$; constraints are represented by infinite penalization. It is natural then that f be lower semicontinuous (l.s.c.). In this setting, useful for many purposes, consider a point $\bar{x}$ where f achieves its minimum, $f(\bar{x})$ being finite. For any $\lambda > 0$, $\bar{x}$ also minimizes the Moreau envelope function

$$e_\lambda(x) := \min_{x'} \left\{ f(x') + \frac{1}{2\lambda}|x' - x|^2 \right\}, \tag{1.1}$$

which provides a *regularization* of f. While f may have ∞ values and exhibit discontinuities, e_λ is finite and locally Lipschitz continuous, and it approximates f in the sense that e_λ increases pointwise to f as $\lambda \downarrow 0$. Among other regularizing effects, e_λ has one-sided directional derivatives at all points, even Taylor expansions of degree 2 almost everywhere, and at $\bar{x}$ it is differentiable with

$$e_\lambda(\bar{x}) = f(\bar{x}), \qquad \nabla e_\lambda(\bar{x}) = 0. \tag{1.2}$$

When f is convex, e_λ is convex too and actually of class C^{1+}: differentiable with locally Lipschitz continuous gradient mapping ∇e_λ.

---

*Received by the editors December 27, 1994; accepted for publication (in revised form) June 13, 1995.

†Department of Mathematical Sciences, University of Alberta, Edmonton, Alberta, Canada T6G 2G1 (rene.poliquin@ualberta.ca). The work of this author was supported by Natural Sciences and Engineering Research Council of Canada grant OGP41983.

‡Department of Mathematics, University of Washington, Seattle, WA 98195 (rtr@math.washington.edu). The work of this author was supported by National Science Foundation grant DMS-9200303.

The prospect is raised that if the properties of e_λ around $\bar{x}$ were adequately understood in their relation to f, it might be possible to develop methods for minimizing e_λ which in effect would open new approaches to minimizing f despite nonsmoothness. Here we analyze e_λ in ways suggested by this notion. Although we do not explore the algorithmic potential directly, we try to identify the circumstances in which e_λ has favorable second-order properties around $\bar{x}$ for the support of numerical work. These include the existence of a Hessian matrix at $\bar{x}$ and the presence of continuity of some sort at $\bar{x}$ in certain generalized second derivatives of e_λ.

In this effort we are following the lead of Lemaréchal and Sagastizábal [13], who have recently drawn attention to the numerical motivations, noting the connections between Moreau envelopes e_λ and basic techniques like the proximal point algorithm. They replaced the canonical squared Euclidean norm in (1.1) by a quadratic form corresponding to any symmetric, positive definite matrix, as obviously would be useful for computational purposes. In theoretical discussions the canonical notation can be retained for simplicity, since the extension can also be viewed as amounting just to a linear change of variables.

Other work building on [13] has been carried out by Qi in [30]. For motivations coming from recent efforts at establishing superlinearly convergent algorithms for solving nonsmooth optimization problems, we refer to [1], [5], [8], [9], [18], [28], and [29].

Lemaréchal and Sagastizábal concentrated in [13] on the case of finite, convex functions f. In this paper, however, we drop both finiteness and convexity as prerequisites on f so as to be able to cover a greater range of problems of optimization. We aim in particular at including the extended real-valued functions f that correspond to standard models in nonlinear programming, for instance, the minimization of

$$f(x) = \begin{cases} f_0(x) & \text{for } x \in C, \\ \infty & \text{for } x \notin C, \end{cases} \tag{1.3}$$

when C is specified by a system of constraints, say

$$C = \{\, x \mid (f_1(x), \ldots, f_m(x)) \in D \,\} \tag{1.4}$$

with D a closed, convex set in $\mathbb{R}^m$ and the functions f_i of class $\mathcal{C}^2$, and with f_0 likewise such a function or perhaps nonsmooth of the form

$$f_0(x) = \max\{\, f_{01}(x), \ldots, f_{0s}(x) \,\} \tag{1.5}$$

for functions f_{0k} of class $\mathcal{C}^2$.

To achieve such coverage we must be careful in selecting the general class of functions f on which to focus our efforts. Obviously the class has to be very broad, if all l.s.c. convex functions are to be encompassed along with functions of the kind just described. Yet if the class were taken to be too broad, it would not have enough structure to secure properties of the kind we are looking for in the envelopes e_λ.

Fortunately a natural choice can be made: the class of *prox-regular* functions f. Such functions, introduced and investigated in [25], include all "strongly amenable" functions, hence all the examples mentioned so far and much more. The theory of their generalized derivatives is available in a form that makes translation of properties between f and e_λ relatively straightforward. The key facts about prox-regular functions developed in [25] will be reviewed in §2. Prox-regularity of f at $\bar{x}$ ensures in particular that e_λ is of class $\mathcal{C}^{1+}$ on a neighborhood of $\bar{x}$.

Functions that are $\mathcal{C}^{1+}$ have been the focus of much research recently. The reader interested in the study of generalized second-order directional derivatives and Hessians of these functions will want to consult the work of Cominetti and Correa [6], Hiriart-Urruty [10], Jeyakumar and Yang [12], Páles and Zeidan [17], and Yang and Jeyakumar [39]. Note that here the function e_λ is not only $\mathcal{C}^{1+}$ but also lower-$\mathcal{C}^2$.

In the exploration of second-order aspects of e_λ, the following notions will be useful.

DEFINITION 1.1. *Consider a function g on $\mathbb{R}^n$ and a point $\hat{x}$ where g is differentiable.*

(a) *g has a second-order expansion at $\hat{x}$ if there is a finite, continuous function h such that the second-order difference quotient functions*

$$\Delta^2_{\hat{x},t} g(\xi) := \big[g(\hat{x}+t\xi) - g(\hat{x}) - t\langle \nabla g(\hat{x}), \xi\rangle\big] \big/ \tfrac{1}{2}t^2$$

converge to h uniformly on bounded sets as $t \downarrow 0$. The expansion is strict if g is differentiable not only at $\hat{x}$ but also on a neighborhood of $\hat{x}$, and the functions

$$\Delta^2_{x,t} g(\xi) := \big[g(x+t\xi) - g(x) - t\langle \nabla g(x), \xi\rangle\big] \big/ \tfrac{1}{2}t^2$$

converge to h uniformly on bounded sets as $t \downarrow 0$ and $x \to \hat{x}$.

(b) *g has a Hessian matrix H at $\hat{x}$, this being a symmetric $n \times n$ matrix, if g has a second-order expansion with $h(\xi) = \langle \xi, H\xi\rangle$. The Hessian is strict if the expansion is strict.*

(c) *g is twice differentiable at $\hat{x}$ if its first partial derivatives exist on a neighborhood of $\hat{x}$ and are themselves differentiable at $\hat{x}$, i.e., the second partial derivatives of g exist at $\hat{x}$. Then $\nabla^2 g(\hat{x})$ denotes the matrix formed by these second partial derivatives.*

A second-order expansion in the sense of Definition 1.1 automatically requires the function h also to be positively homogeneous of degree 2: $h(\lambda\xi) = \lambda^2 h(\xi)$ for $\lambda > 0$, and, in particular, $h(0) = 0$. In traditional notation it means that

$$g(\hat{x}+t\xi) \;=\; g(\hat{x}) + t\langle \nabla g(\hat{x}), \xi\rangle + \tfrac{1}{2}t^2 h(\xi) + o\big(t^2|\xi|^2\big)$$

for such a function h that is finite and continuous. The existence of a Hessian corresponds to h actually being quadratic. Note that, in general, this property might be present without the Hessian matrix H being the matrix $\nabla^2 g(\hat{x})$ of classical twice differentiability, since the latter requires more for its definition than the mere assumption that the first partial derivatives of g exist at $\hat{x}$.

Our fundamental theorem, in §3, says that when f is prox-regular at a point $\bar{x} \in \operatorname{argmin} f$ $(= \operatorname{argmin} e_\lambda)$, the existence of a second-order expansion of e_λ at $\bar{x}$ (when λ is sufficiently small) is equivalent to the "twice epi-differentiability" of f at $\bar{x}$. This is especially illuminating because it shows one cannot expect the regularized function e_λ to have significant second-order properties without some such second-order property already being present in f, moreover a kind of property satisfied by many, but definitely not all, convex or lower-$\mathcal{C}^2$ functions. This is in sharp contrast with the first-order theory, where no first-order assumptions on f are needed to conclude that e_λ is differentiable (in fact C^{1+}) around a point $\bar{x} \in \operatorname{argmin} f$ when f is prox-regular. Twice epi-differentiability is enjoyed by all fully amenable functions in particular, so second-order expansions of e_λ nonetheless are the rule for most applications.

Proceeding from this theorem, we are able to demonstrate another special feature of the Moreau envelopes e_λ when f is prox-regular. A Hessian H exists for e_λ at $\bar{x} \in$

argmin f if and only if the gradient mapping ∇e_λ is differentiable at $\bar{x}$, in which case H is the matrix $\nabla(\nabla e_\lambda)(\bar{x}) = \nabla^2 e_\lambda(\bar{x})$ of second-partial derivatives—which therefore must be symmetric. This is somewhat surprising, since the ordinary criterion for the symmetry of the matrix of second derivatives of a function is that the function be of class $\mathcal{C}^2$, whereas here the first derivatives are only known to be Lipschitz continuous, hence perhaps only differentiable almost everywhere.

We show further that the existence of a Hessian for e_λ does not require the finiteness of f but merely that the corresponding second-order epi-derivative function for f be a "generalized quadratic function" (i.e., a quadratic function on a certain subspace but ∞ elsewhere). This covers situations where, for example, f has the nonlinear programming form (1.3)–(1.5) and $\bar{x}$ is a point where f is not differentiable, much less twice differentiable. But in particular, if f does have a Hessian at a point $\bar{x} \in \operatorname{argmin} f$, then e_λ inherits this property.

In §4 we establish that e_λ cannot have a strict second-order expansion at $\bar{x} \in \operatorname{argmin} f$ unless it actually has a strict Hessian there. We characterize this situation in many ways, tying it to various properties of ∇e_λ and f. Our efforts disclose that this is the one and only case, short of e_λ actually being $\mathcal{C}^2$ around $\bar{x}$, where second-order properties of e_λ and f exhibit a degree of local stability.

2. Prox-regularity and derivatives. Continuing under the assumption that f is a proper l.s.c. function on $\mathbb{R}^n$, we denote by $\partial f(x)$ the set of limiting proximal subgradients of f at any point $x \in \operatorname{dom} f$. For a recent exposition of the theory of subgradients, see Loewen [16].

DEFINITION 2.1. *The function f is prox-regular at $\hat{x}$ relative to $\hat{v}$ if $\hat{x} \in \operatorname{dom} f$, and $\hat{v} \in \partial f(\hat{x})$ and there exist $\epsilon > 0$ and $r > 0$ such that*

$$f(x') > f(x) + \langle v, x' - x \rangle - \frac{r}{2}|x' - x|^2 \tag{2.1}$$

whenever $|x' - \hat{x}| < \epsilon$ and $|x - \hat{x}| < \epsilon$, with $x' \neq x$ and $|f(x) - f(\hat{x})| < \epsilon$, while $|v - \hat{v}| < \epsilon$ with $v \in \partial f(x)$.

It is obvious that if f is convex, it is prox-regular at $\hat{x}$ for every subgradient $\hat{v} \in \partial f(\hat{x})$. The same is true for lower-$\mathcal{C}^2$ functions, primal-lower-nice (p.l.n.) functions, and *strongly amenable* functions; cf. [25]. Strong amenability of f at $\hat{x}$ refers to the existence of a representation $f = g \circ F$ in a neighborhood of $\hat{x}$ for a mapping $F : \mathbb{R}^n \to \mathbb{R}^m$ of class C^2 and a proper l.s.c. convex function $g : \mathbb{R}^m \to \overline{\mathbb{R}}$ satisfying at $\hat{x}$ with respect to the convex set $D = \operatorname{dom} g$ the basic constraint qualification that

$$\text{there is no vector } y \neq 0 \text{ in } N_D\big(F(\hat{x})\big) \text{ with } \nabla F(\hat{x})^* y = 0. \tag{2.2}$$

Amenability itself merely requires F to be of class $\mathcal{C}^1$, whereas *full* amenability is the subcase of strong amenability where g is also piecewise linear-quadratic.

Full amenability already covers most applications that arise in the framework of nonlinear programming and its extensions. A function f as in (1.3)–(1.5), for instance, is fully amenable at any point $\hat{x} \in C$ at which the constraint qualification (2.2) is satisfied with $F = (f_1, \ldots, f_m)$; cf. [33] and [35]. If D consists of the vectors $u = (u_1, \ldots, u_m)$ with $u_i \leq 0$ for $i = 1, \ldots, s$ but $u_i = 0$ for $i = s+1, \ldots, m$, so that C is specified by constraints $f_i(x) \leq 0$ and $f_i(x) = 0$, this constraint qualification reduces to the standard one of Mangasarian–Fromovitz. For more on amenability, see [22]–[24]; for composite functions more broadly, see [3], [11], [12], [14], [15], [19]–[27].

Note that when f is *convex*, the parameter $\epsilon > 0$ in the definition of prox-regularity can be taken *arbitrarily large*, while the parameter $r > 0$ can be taken *arbitrarily small*. This lends a stronger interpretation to some of the results below.

For any $\epsilon > 0$, we speak of the *f-attentive ϵ-localization* of ∂f at $(\hat{x}, \hat{v})$ in referring to the mapping $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by

$$T(x) = \begin{cases} \{ v \in \partial f(x) \,|\, |v - \hat{v}| < \epsilon \} & \text{when } |x - \hat{x}| < \epsilon \text{ and } |f(x) - f(\hat{x})| < \epsilon, \\ \emptyset & \text{otherwise.} \end{cases} \tag{2.3}$$

The graph of T is the intersection of $\operatorname{gph} \partial f$ with the product of an f-attentive neighborhood of $\hat{x}$ and an ordinary neighborhood of $\hat{v}$. (The f-attentive topology on $\mathbb{R}^n$ is the weakest topology in which f is continuous.)

In cases where f is subdifferentially continuous at $\hat{x}$ for $\hat{v}$, in the sense that $f(x) \to f(\hat{x})$ automatically when $(x, v) \to (\hat{x}, \hat{v})$ in the graph of ∂f, the $|f(x) - f(\hat{x})| < \epsilon$ enforcing f-attentiveness in (2.3) could be dropped without affecting any of the results below. All amenable functions, in particular, l.s.c. convex functions and lower-$\mathcal{C}^2$ functions, are subdifferentially continuous.

THEOREM 2.2 (see [25, Thm. 3.2]). *The function f is prox-regular at a point $\hat{x}$ relative to $\hat{v}$ if and only if the vector $\hat{v}$ is a proximal subgradient of f at $\hat{x}$ and there exist $\epsilon > 0$ and $r > 0$ such that, for the f-attentive ϵ-localization T of ∂f, the mapping $T + rI$ is monotone, i.e., one has $\langle v_1 - v_0, x_1 - x_0 \rangle \geq -r|x_1 - x_0|^2$ whenever $|x_i - \hat{x}| < \epsilon$, $|f(x_i) - f(\hat{x})| < \epsilon$, and $|v_i - \hat{v}| < \epsilon$ with $v_i \in \partial f(x_i)$, $i = 0, 1$.*

Concepts of generalized second-order differentiability beyond those already given in Definition 1.1 will be crucial in our analysis. Recall f is *twice epi-differentiable* at $\hat{x}$ for a vector $\hat{v}$ if $\hat{v} \in \partial f(\hat{x})$ and the second-order difference quotient functions $\Delta^2_{\hat{x},\hat{v},t} f : \mathbb{R}^n \to \overline{\mathbb{R}}$ defined by

$$\Delta^2_{\hat{x},\hat{v},t} f(\xi) = \big[f(\hat{x} + t\xi) - f(\hat{x}) - t\langle \hat{v}, \xi \rangle \big] / \tfrac{1}{2} t^2 \text{ for } t > 0 \tag{2.4}$$

epi-converge to a proper function as $t \downarrow 0$. The limit is then the *second epi-derivative* function $f''_{\hat{x},\hat{v}} : \mathbb{R}^n \to \overline{\mathbb{R}}$; see [23], [33], and [35]. (Epi-convergence of functions refers to set convergence of their epigraphs.) We say that f is *strictly* twice epi-differentiable at $\hat{x}$ for $\hat{v}$ if more generally the functions

$$\Delta^2_{x,v,t} f(\xi) = \big[f(x + t\xi) - f(x) - t\langle v, \xi \rangle \big] / \tfrac{1}{2} t^2 \text{ for } t > 0 \tag{2.5}$$

epi-converge as $t \downarrow 0$, $x \to \hat{x}$ with $f(x) \to f(\hat{x})$, and $v \to \hat{v}$ with $v \in \partial f(x)$. When $\partial f(\hat{x})$ is a singleton consisting of $\hat{v}$ alone (as when f is strictly differentiable at $\hat{x}$), the notation $f''_{\hat{x},\hat{v}}$ can be simplified to $f''_{\hat{x}}$ and we can just speak of f being twice epi-differentiable, or strictly twice epi-differentiable, at $\hat{x}$, the particular vector $\hat{v}$ being implicit.

Next, a set-valued mapping $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is *proto-differentiable* at $\hat{x}$ for $\hat{v}$ if $\hat{v} \in T(\hat{x})$ and the set-valued mappings

$$\Delta_{\hat{x},\hat{v},t} T : \xi \mapsto \big[T(\hat{x} + t\xi) - \hat{v} \big] / t \text{ for } t > 0 \tag{2.6}$$

graph-converge as $t \downarrow 0$ (i.e., their graphs converge as subsets of $\mathbb{R}^n \times \mathbb{R}^n$). If so, the limit mapping is denoted by $T'_{\hat{x},\hat{v}}$ and called the *proto-derivative* of T at $\hat{x}$ for $\hat{v}$; see [24], [34], [36]. This mapping assigns to each $\xi \in \mathbb{R}^n$ a subset $T'_{\hat{x},\hat{v}}(\xi)$ of $\mathbb{R}^n$, which could be empty for some choices of ξ. We say that T is *strictly* proto-differentiable at $\hat{x}$ for $\hat{v}$ if more generally the mappings

$$\Delta_{x,v,t} T : \xi \mapsto \big[T(x + t\xi) - v \big] / t \text{ for } t > 0 \tag{2.7}$$

graph-converge as $t \downarrow 0$, $x \to \hat{x}$, and $v \to \hat{v}$ with $v \in T(x)$. Again, in the case where $T(\hat{x})$ is a singleton consisting of $\hat{v}$ only (as for instance in the case where T is actually single-valued everywhere), the notation $T'_{\hat{x},\hat{v}}(\xi)$ can be simplified to $T'_{\hat{x}}(\xi)$, and we can just speak of T being proto-differentiable at $\hat{x}$ or strictly proto-differentiable at $\hat{x}$.

THEOREM 2.3 (see [25, Thm. 6.1]). *Suppose that f is prox-regular at $\hat{x}$ for $\hat{v}$ with respect to ϵ and r, and let T be the f-attentive ϵ-localization of ∂f at $(\hat{x},\hat{v})$. Then f is twice epi-differentiable at $\hat{x}$ for $\hat{v}$ if and only if T is proto-differentiable at $\hat{x}$ for $\hat{v}$, in which event one has*

$$T'_{\hat{x},\hat{v}}(\xi) = \partial\big[\tfrac{1}{2} f''_{\hat{x},\hat{v}}\big](\xi) \text{ for all } \xi.$$

Furthermore, f is strictly twice epi-differentiable at $\hat{x}$ for $\hat{v}$ if and only if T is strictly proto-differentiable at $\hat{x}$ for $\hat{v}$.

Henceforth we concentrate on the case of minimizing points $\bar{x} \in \operatorname{argmin} f$. Such points have $\bar{v} = 0$ as subgradient $\bar{v} \in \partial f(\bar{x})$. As a companion to the envelope function e_λ we utilize the *proximal mapping* $P_\lambda : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by

$$P_\lambda(x) := \operatorname*{argmin}_{x'} \left\{ f(x') + \frac{1}{2\lambda}|x' - x|^2 \right\}. \tag{2.8}$$

THEOREM 2.4 (see [25, Thms. 4.4, 4.6, 5.2]). *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r, and let T be the f-attentive ϵ-localization of ∂f at $(\bar{x},0)$. Then for any $\lambda \in (0, 1/r)$ there is a convex neighborhood X_λ of $\bar{x}$ such that*

(a) *the mapping P_λ is single-valued and Lipschitz continuous on X_λ with*

$$P_\lambda = (I + \lambda T)^{-1}, \qquad |P_\lambda(x') - P_\lambda(x)| \le \frac{1}{1 - \lambda r}|x' - x|, \qquad P_\lambda(\bar{x}) = \bar{x};$$

(b) *the function e_λ is $\mathcal{C}^{1+}$ and lower-$\mathcal{C}^2$ on X_λ with*

$$e_\lambda + \frac{r}{2(1 - \lambda r)}|\cdot|^2 \text{ convex}, \qquad \nabla e_\lambda = \lambda^{-1}\big[I - P_\lambda\big] = \big[\lambda I + T^{-1}\big]^{-1}.$$

3. Second-order expansions. Before looking at second-order expansions of the Moreau envelope functions e_λ, we inspect the case of f itself.

THEOREM 3.1. *Suppose that f is prox-regular at $\hat{x}$ for $\hat{v}$. Then the following are equivalent and imply that f is lower-$\mathcal{C}^2$ on a neighborhood of $\hat{x}$, with*

$$f(\hat{x} + t\xi) \;=\; f(\hat{x}) + t\langle \hat{v}, \xi \rangle + \tfrac{1}{2} t^2 f''_{\hat{x},\hat{v}}(\xi) + o\big(t^2|\xi|^2\big). \tag{3.1}$$

(a) *f is differentiable at $\hat{x}$, $\nabla f(\hat{x}) = \hat{v}$, and f has a second-order expansion at $\hat{x}$.*

(b) *f is twice epi-differentiable at $\hat{x}$ for $\hat{v}$ with $f''_{\hat{x},\hat{v}}$ finite everywhere.*

Proof. The equivalence of (b) with the expansion (3.1) is asserted by [25, Thm. 6.7] with the function $f''_{\hat{x},\hat{v}}$ being in this case lower-$\mathcal{C}^2$ (hence continuous), and the same for f on some neighborhood of $\hat{x}$. This in turn implies (a) through Definition 1.1 with $h = f''_{\hat{x},\hat{v}}$. Conversely, if we had (a) the functions $\Delta^2_{\hat{x},\hat{v},t} f$ would in particular epi-converge to h, so $h = f''_{\hat{x},\hat{v}}$. This would give (b). □

It might be conjectured that when f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ and also twice epi-differentiable for these elements, the second-order epi-derivative

function $f''_{\bar{x},\bar{v}}$ ought to be *convex* (in addition to being nonnegative, l.s.c., and positively homogeneous of degree 2). But this is false, even if $f''_{\bar{x},\bar{v}}$ is finite everywhere. An example in $\mathbb{R}^2$ is

$$f(x_1, x_2) = |x_1 x_2| = \max\{-x_1 x_2,\ x_1 x_2\}$$

at the origin. This function is lower-$\mathcal{C}^2$, hence certainly prox-regular everywhere, but its second-order epi-derivative function at the origin is f itself, which is not convex.

THEOREM 3.2. *Suppose that f is lower-$\mathcal{C}^2$ on a neighborhood of $\hat{x}$ and also differentiable on such a neighborhood (or equivalently, that f can be represented around $\hat{x}$ as the difference between a differentiable convex function and a $\mathcal{C}^2$ function). Then the following are equivalent:*

(a) *f has a Hessian matrix H at $\hat{x}$;*

(b) *∇f is differentiable at $\hat{x}$ with Jacobian matrix H;*

(c) *f is twice differentiable at $\hat{x}$, and $H = \nabla^2 f(\hat{x})$.*

Proof. The parenthetical alternative to the hypothesis on f is known from [37]. From this it is clear that differentiable, lower-$\mathcal{C}^2$ functions f are $\mathcal{C}^1$ with $\partial f(x) = \{\nabla f(x)\}$. Then Theorem 2.3 is applicable, with T simply a localization of ∇f. Denote by H_a and H_b the matrices that appear in (a) and (b), respectively. Conditions (a) and (b) both imply that f has a second-order expansion at $\hat{x}$ with $h(\xi) = \langle \xi, H_a \xi \rangle = \langle \xi, H_b \xi \rangle$. Theorem 3.1 tells us that f is twice epi-differentiable at $\hat{x}$ for $\hat{v} = \nabla f(\hat{x})$ with $f''_{\hat{x},\hat{v}}(\xi) = h(\xi)$. This in turn means by Theorem 2.3 that

$$H_b(\xi) = \nabla^2 f(\hat{x})(\xi) = \big(\partial f\big)'_{\hat{x},\hat{v}}(\xi) = \partial\big[\tfrac{1}{2} f''_{\hat{x},\hat{v}}\big](\xi) = H_a(\xi) = \tfrac{1}{2}\big[H_b + H_b{}^T\big](\xi) \text{ for all } \xi$$

(recall that H_a is symmetric). This yields the equivalence of (a) and (b), while (c) just restates (b). □

Theorem 3.2 is not new—it has recently been proved even in a Banach-space version by Borwein and Noll [2, Thm. 3.1] for convex functions f, from which it readily extends to lower-$\mathcal{C}^2$ functions. Our proof is different, however, in its reliance on fundamental relationships between second-order epi-derivatives of functions and proto-derivatives of their subgradient mappings.

COROLLARY 3.3. *For a function f of the kind in Theorem 3.2, and in particular for any differentiable convex function f, twice differentiability at $\hat{x}$ automatically implies the symmetry of the second-derivative matrix $\nabla^2 f(\hat{x})$.*

This fact is interesting because second-derivative matrices do not have to be symmetric for differentiable functions in general. The standard criterion for ensuring symmetry is continuous twice differentiability of f, but that is not assumed in the situation at hand. Also, apart from the symmetry issue, the existence of a Hessian H for a differentiable function f does not automatically mean, unless f is lower-C^2 as in Theorem 3.2, the existence of the second-derivative matrix $\nabla^2 f(\hat{x})$. This is confirmed by the example of

$$f(x) = \begin{cases} |x|^2 \sin\left(\frac{1}{|x|^2}\right) & \text{for } x \neq 0, \\ 0 & \text{for } x = 0, \end{cases}$$

which is everywhere differentiable and has Hessian $H = 0$ at $\hat{x} = 0$ but has ∇f not even bounded on a neighborhood of 0, much less differentiable there; no Jacobian exists at all for ∇f at 0. These facts highlight the special nature of lower-$\mathcal{C}^2$ functions, and more specifically finite convex functions, in the absence of such pathology.

Our basic result about second-order expansions of envelope functions will involve first-order expansions of their gradient mappings. For this we need another definition.

DEFINITION 3.4. *A single-valued mapping G from an open neighborhood of $\hat{x} \in \mathbb{R}^n$ into $\mathbb{R}^m$ has a first-order expansion at a point $\hat{x} \in O$ if there is a continuous mapping D such that the difference quotient mappings*

$$\Delta_{\hat{x},t}G : \big[G(\hat{x}+t\xi) - G(\hat{x})\big]/t \text{ for } t > 0$$

converge to D uniformly on bounded sets as $t \downarrow 0$. The expansion is strict if actually the mappings

$$\Delta_{x,t}G : \big[G(x+t\xi) - G(x)\big]/t \text{ for } t > 0$$

converge to D uniformly on bounded sets as $t \downarrow 0$ and $x \to \hat{x}$.

When a first-order expansion exists, the mapping D giving the approximating term must be not only continuous but also positively homogeneous: $D(\lambda\xi) = \lambda D(\xi)$ for $\lambda > 0$, and in particular $D(0) = 0$. The expansion can be expressed by

$$G(\hat{x}+t\xi) \;=\; G(\hat{x}) + tD(\xi) + o\big(|t\xi|\big).$$

Differentiability of G at $\hat{x}$ in the classical sense is the case where D happens to be a linear mapping. If D is linear and the expansion is strict, one has the *strict differentiability* of G at $\hat{x}$.

In general, the existence of a first-order expansion means the *directional differentiability* of G at $\hat{x}$. It is equivalent (in our finite-dimensional setting) to the existence, for every vector $\hat{\xi} \in \mathbb{R}^n$, of the *directional derivative* limit

$$\lim_{\substack{\xi \to \hat{\xi} \\ t \downarrow 0}} \frac{G(\hat{x}+t\xi) - G(\hat{x})}{t}.$$

Likewise, a strict first-order expansion means *strict* directional differentiability at $\hat{x}$ and corresponds to the existence for every $\hat{\xi}$ of the more complicated limit where $\hat{x}$ is replaced by x, and $x \to \hat{x}$ along with $\xi \to \hat{\xi}$ and $t \downarrow 0$. Either way, the mapping D in Definition 3.4 gives for each $\hat{\xi}$ the directional derivative $D(\hat{\xi})$.

The existence of a first-order expansion can also be identified with a special case of proto-differentiability (a concept applicable to single-valued mappings in particular). Indeed, G has a first-order expansion at $\hat{x}$ in terms of D if and only if G is proto-differentiable at $\hat{x}$ and the proto-derivative mapping $G'_{\hat{x}}$ has no empty values. Then $D = G'_{\hat{x}}$, so we get

$$G(\hat{x}+t\xi) \;=\; G(\hat{x}) + tG'_{\hat{x}}(\xi) + o\big(|t\xi|\big).$$

THEOREM 3.5. *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r, and let $\lambda \in (0, 1/r)$. Then the following properties are equivalent:*

(a) *f is twice epi-differentiable at $\bar{x}$ for $\bar{v} = 0$;*
(b) *e_λ is twice epi-differentiable at $\bar{x}$ for $\bar{v} = 0$;*
(c) *e_λ has a second-order expansion at $\bar{x}$;*
(d) *∇e_λ has a first-order expansion at $\bar{x}$;*
(e) *∇e_λ is proto-differentiable at $\bar{x}$;*
(f) *P_λ has a first-order expansion at $\bar{x}$;*
(g) *P_λ is proto-differentiable at $\bar{x}$.*

In that event, the expansions

$$\begin{aligned} e_\lambda(\bar{x}+t\xi) &= e_\lambda(\bar{x}) + t^2 d_\lambda(\xi) + o\big(|t\xi|^2\big), \\ \nabla e_\lambda(\bar{x}+t\xi) &= t\nabla d_\lambda(\xi) + o\big(|t\xi|\big), \\ P_\lambda(\bar{x}+t\xi) &= \bar{x} + t\big[I - \lambda\nabla d_\lambda(\xi)\big] + o\big(|t\xi|\big) \end{aligned}$$

hold for a function d_λ that is both $\mathcal{C}^{1+}$ and lower-$\mathcal{C}^2$, the gradient mapping ∇d_λ being Lipschitz continuous globally. In fact d_λ is the Moreau λ-envelope for $\frac{1}{2}f''_{\bar{x},\bar{v}}$; one has

$$d_\lambda(\xi) = \min_{\xi'}\left\{\frac{1}{2}f''_{\bar{x},\bar{v}}(\xi') + \frac{1}{2\lambda}|\xi' - \xi|^2\right\} \text{ for all } \xi. \tag{3.2}$$

Moreover,

$$\partial\left[\tfrac{1}{2}(d_\lambda)\right] = \left[\lambda I + D^{-1}\right]^{-1} = \lambda^{-1}\left[I - (I + \lambda D)^{-1}\right] \text{ for } D := \partial\left[\tfrac{1}{2}f''_{\bar{x},\bar{v}}\right]. \tag{3.3}$$

Proof. The equivalence of (a) and (b) is asserted along with (3.2) and (3.3) by [25, Cor. 6.6]. On the other hand, the equivalence of (b) and (c) follows from applying Theorem 3.2 to e_λ in place of f, which is possible because of the properties of e_λ in Theorem 2.4(b). In the same context, (b) is equivalent to (e) through Theorem 2.3.

As noted above, the existence of a first-order expansion is a special case of proto-differentiability; (d) implies (e). Conversely, the local Lipschitz continuity of ∇e_λ guarantees that the difference quotient mappings

$$\xi \mapsto \left[\nabla e_\lambda(\bar{x} + t\xi) - \nabla e_\lambda(\bar{x})\right]/t$$

enjoy the same Lipschitz constant on some neighborhood of $\xi = 0$ for sufficiently small $t > 0$, so that graphical convergence of these mappings, as demanded by proto-differentiability, is impossible unless they converge pointwise. But uniformly Lipschitz continuous mappings that converge pointwise must converge uniformly on bounded sets, moreover to a limit mapping that likewise is single-valued Lipschitz continuous with the same constant on a neighborhood of 0. In this case the limit mapping is positively homogeneous, so the latter properties hold not just on a neighborhood of 0 but globally on $\mathbb{R}^n$. Thus, (e) implies (d) and the single-valuedness and global Lipschitz continuity of the proto-derivative $(e_\lambda)'_{\bar{x}}$. The relationships with P_λ come immediately now out of Theorem 2.4. □

COROLLARY 3.6. *The properties in Theorem* 3.5 *hold when f is fully amenable at $\bar{x}$.*

Proof. Fully amenable functions are twice epi-differentiable everywhere; cf. [33]. □

Moving on now to the question of the existence and characterization of Hessian expansions, we require two special concepts.

DEFINITION 3.7. *A function $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ is a generalized (purely) quadratic function if it is expressible in the form*

$$h(x) = \begin{cases} \frac{1}{2}\langle x, Qx\rangle & \text{if } x \in N, \\ \infty & \text{if } x \notin N, \end{cases} \tag{3.4}$$

where N is a linear subspace of $\mathbb{R}^n$ and Q is a symmetric $n \times n$ matrix. A possibly set-valued mapping $D : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is a generalized linear mapping if its graph is a linear subspace of $\mathbb{R}^n \times \mathbb{R}^m$.

Generalized quadratic functions can be interpreted as "quadratic functions with $+\infty$ as a possible eigenvalue" (the subspace $N^\perp$ complementary to N being the eigenspace for that eigenvalue). They are precisely the functions h on $\mathbb{R}^n$ whose subgradient mappings ∂h are generalized linear mappings, the graph of ∂h necessarily being in fact an n-dimensional subspace of $\mathbb{R}^n \times \mathbb{R}^n$; see [32, §4].

THEOREM 3.8. *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r, and let $\lambda \in (0, 1/r)$. Then the following properties are equivalent:*

(a) *f is twice epi-differentiable at $\bar{x}$ for $\bar{v}$, and $f''_{\bar{x},\bar{v}}$ is generalized quadratic;*

(b) *e_λ has a Hessian matrix H_λ at $\bar{x}$;*

(c) *∇e_λ is differentiable at $\bar{x}$ with Jacobian matrix H_λ;*

(d) *e_λ is twice differentiable at $\bar{x}$ for $\bar{v}$, and $H_\lambda = \nabla^2 e_\lambda(\bar{x})$;*

(e) *P_λ is differentiable at $\bar{x}$ with Jacobian matrix $I - \lambda H_\lambda$;*

(f) *the f-attentive ϵ-localization T of ∂f at $(\bar{x}, \bar{v})$ is proto-differentiable there, and the proto-derivative mapping $T'_{\bar{x},\bar{v}}$ is generalized linear.*

Proof. We have (b), (c), and (e) equivalent as the special case of Theorem 3.5 (a), (b), and (f) in which d_λ is quadratic. Next, (d) is just another way of saying (c), which is also the same as asserting that the mapping ∇e_λ is proto-differentiable at $\bar{x}$ for $\bar{v} = 0$ and the graph of the proto-derivative mapping is a linear subspace of $\mathbb{R}^n$ (cf. the remarks about proto-differentiability in the proof of Theorem 3.5). Through the last formula in Theorem 2.4, which sets up a linear transformation in $\mathbb{R}^n \times \mathbb{R}^n$ under which the graph of ∇e_λ corresponds locally to that of T, proto-differentiability of ∇e_λ corresponds to that of T. Thus, (c) is equivalent to (f). But by Theorem 2.3, (f) is equivalent to (a). □

THEOREM 3.9. *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r. Then there exist $\lambda_0 > 0$ and a neighborhood U of $(\bar{x}, \bar{v})$ such that for all $0 < \lambda < \lambda_0$ and $(x, v) \in \operatorname{gph} T \cap U$ (with T the f-attentive ϵ-localization of ∂f at $(\bar{x}, \bar{v})$) the following properties are equivalent:*

(a) *f is twice epi-differentiable at x for v, and $f''_{x,v}$ is generalized quadratic;*

(b) *e_λ has a Hessian matrix H_λ at $x + \lambda v$;*

(c) *∇e_λ is differentiable at $x + \lambda v$ with Jacobian matrix H_λ;*

(d) *e_λ is twice differentiable at $x + \lambda v$ for v, and $H_\lambda = \nabla^2 e_\lambda(x + \lambda v)$;*

(e) *P_λ is differentiable at $x + \lambda v$ with Jacobian matrix $I - \lambda H_\lambda$;*

(f) *the f-attentive ϵ-localization T of ∂f at (x, v) is proto-differentiable there, and the proto-derivative mapping $T'_{x,v}$ is generalized linear.*

Proof. We can assume that $\bar{x} = 0$, $f(\bar{x}) = 0$. Fix x and $v \in \partial f(x)$ with $|x| < \epsilon/4$, $|v| < \epsilon$, and $|f(x)| < \epsilon$. We claim that by choosing $R > r$ sufficiently large, we can make the function $\bar{f}(x') := f(x') - \langle v, x' - x \rangle + (R/2)|x' - x|^2$ have $x \in \operatorname{argmin} \bar{f}$. To see this, notice that because f is prox-regular at 0 for 0 we have for any R and any x' with $|x'| < \epsilon$ that

$$\begin{aligned}\bar{f}(x') &\geq f(x) + \langle v, x' - x \rangle - \left(\frac{r}{2}\right)|x' - x|^2 - \langle v, x' - x \rangle + \left(\frac{R}{2}\right)|x' - x|^2 \\ &= f(x) + \left(\frac{(R - r)}{2}\right)|x' - x|^2 \geq f(x) = \bar{f}(x).\end{aligned}$$

On the other hand, we have for any R and any x' with $|x'| \geq \epsilon$ (and hence in particular with $|x' - x| \geq (3/4)\epsilon$) that

$$\begin{aligned}\bar{f}(x') &\geq f(0) - \langle v, x' - x \rangle + \left(\frac{R}{2}\right)|x' - x|^2 \\ &= f(x) - f(x) - \langle v, x' - x \rangle + \left(\frac{R}{2}\right)|x' - x|^2 \\ &\geq f(x) + \left(\frac{R}{2}\right)|x' - x|^2 - \epsilon|x' - x| - \epsilon.\end{aligned}$$

This last quantity is greater than $f(x)$ provided $|x' - x| \geq (1/R)[\epsilon + \sqrt{\epsilon^2 + 2R\epsilon}]$. By choosing R large enough we can have this last quantity less than $(3/4)\epsilon$. Then $\bar{f}(x') \geq \bar{f}(x)$ for all x', so $x \in \operatorname{argmin} \bar{f}$.

With R having been chosen large enough that $x \in \operatorname{argmin} \bar{f}$, and with the observation that $\bar{f}$ is prox-regular at x for 0, we now turn our attention to the Moreau λ-envelope of $\bar{f}$, which we denote by $\bar{f}_\lambda$. This is given by

$$\begin{aligned}
\bar{f}_\lambda(w) &= \inf_{x'} \left\{ g(x') + \frac{1}{2\lambda}|x' - w|^2 \right\} \\
&= \inf_{x'} \left\{ f(x') - \langle v, x' - x \rangle + \frac{R}{2}|x' - x|^2 + \frac{1}{2\lambda}|x' - w|^2 \right\} \\
&= \inf_{x'} \left\{ f(x') + \frac{1}{2\mu}|u - x'|^2 \right\} - \frac{1}{2\mu}|u|^2 + \frac{1}{2\lambda}|w|^2 + \langle v, x \rangle + \frac{R}{2}|x|^2 \\
&= e_\mu(u) - \frac{1}{2\mu}|u|^2 + \frac{1}{2\lambda}|w|^2 + \langle v, x \rangle + \frac{R}{2}|x|^2,
\end{aligned}$$

where $\mu = \lambda/(\lambda R + 1)$ and $u = \mu[v + Rx + (1/\lambda)w]$; to obtain the penultimate step in the preceding calculation simply note that

$$\frac{1}{2\mu}|u - x'|^2 = \frac{1}{2\mu}|u|^2 - \langle v + Rx + \left(\frac{1}{\lambda}\right) w, x' \rangle + \left(\frac{R}{2} + \frac{1}{2\lambda}\right)|x'|^2.$$

From the formula for $\bar{f}_\lambda$ we see that the second-order properties of $\bar{f}_\lambda$ at x for 0 translate to second-order properties of e_μ at $\mu[v + Rx + (1/\lambda)x] = \mu v + x$ for v (since by Theorem 2.4, $\nabla e_\mu(x + \mu v) = v$). Besides, $\bar{f}$ is twice epi-differentiable at x for 0 if and only if f is twice epi-differentiable at x for v.

Now we apply the equivalent properties in Theorem 3.8 not just at $(\bar{x}, \bar{v})$ but locally. □

COROLLARY 3.10. *Suppose that f is prox-regular at a point $\hat{x}$ for $\hat{v}$ with constants ϵ and r. Then there is a neighborhood U of $(\hat{x}, \hat{v})$ such that for all $(x, v) \in \operatorname{gph} T \cap U$ (with T the f-attentive ϵ-localization of ∂f at $(\hat{x}, \hat{v})$) the following properties are equivalent:*

(a) *f is twice epi-differentiable at x for v and $f''_{x,v}$ is generalized quadratic;*

(b) *T is proto-differentiable at x for v, and the proto-derivative mapping $T'_{x,v}$ is generalized linear.*

Proof. Take $(x, v) \in \operatorname{gph} T$ with $|x - \hat{x}| < (\epsilon/2)$. We have $f(x') \geq f(x) + \langle v, x' - x \rangle - (r/2)|x' - x|^2$ whenever $|x' - \hat{x}| < \epsilon$. Therefore, for some neighborhood U of x the function $g(x') := f(x') - \langle v, x' - x \rangle + (r/2)|x' - x|^2 + \delta_U(x')$ has $x \in \operatorname{argmin} g$ and $0 \in \partial g(x)$. In addition, the function g is prox-regular at x for 0. Now apply Theorem 3.8 to g at x for 0; the equivalent properties for g easily translate into equivalent properties for f. □

COROLLARY 3.11. *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r.*

(a) *If for some $\lambda \in (0, 1/r)$ the function e_λ has a Hessian matrix at $\bar{x}$, this must be true for every $\lambda \in (0, 1/r)$.*

(b) *If for some $\lambda \in (0, 1/r)$ the mapping ∇e_λ is differentiable at $\bar{x}$, this must be true for every $\lambda \in (0, 1/r)$.*

(c) *If for some $\lambda \in (0, 1/r)$ the mapping P_λ is differentiable at $\bar{x}$, this must be true for every $\lambda \in (0, 1/r)$.*

COROLLARY 3.12. *Suppose f is lower-$\mathcal{C}^2$ on a neighborhood of $\bar{x} \in \operatorname{argmin} f$ and there exists $s > 0$ such that $f(x) \leq f(\bar{x}) + s|x - \bar{x}|^2$ for all x in some neighborhood of $\bar{x}$. Then for $\lambda > 0$ sufficiently small, e_λ has a Hessian at $\bar{x}$ if and only if f itself has a Hessian at $\bar{x}$.*

Proof. The bound given on the growth of f around $\bar{x}$ has the consequence that if the second-order epi-derivative function $f''_{\bar{x},\bar{v}}$ exists it must be finite. □

4. Stability of approximation. The existence of a second-order expansion of e_λ at $\bar{x}$ provides a type of second-order approximation which may be useful for a number of purposes. To get full benefit from such an approximation in a numerical context, however, it must display some degree of stability. In a standard framework of analysis relying on differentiability, the property to focus on is that of e_λ being of class $\mathcal{C}^2$ on a neighborhood of $\bar{x}$; then Hessian matrices exist at points x near $\bar{x}$ and depend continuously on x. We must ascertain what property of f is necessary and sufficient for this but also try to explore less demanding notions of stability which might nonetheless provide a numerical handle in the environment where e_λ is not $\mathcal{C}^2$, just C^{1+} and lower-$\mathcal{C}^2$ around $\bar{x}$ (through Theorem 2.4).

Strict second-order expansions of e_λ (Definition 1.1) and strict first-order expansions of ∇e_λ (Definition 3.4) offer promising territory. Strict differentiability, as the case of a strict first-order expansion where the approximating mapping is linear, is known to have the interpretation as the single-point localization of the $\mathcal{C}^1$ property, and something similar might be anticipated for the existence of a strict Hessian in relation to the $\mathcal{C}^2$ property. In exploring this, we appeal to the strict versions of proto-differentiability and twice epi-differentiability defined before Theorem 2.3.

THEOREM 4.1. *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r, and let $\lambda \in (0, 1/r)$. Then the following properties are equivalent and imply that $f''_{\bar{x},\bar{v}}$ is generalized quadratic and $T'_{\bar{x},\bar{v}}$ is generalized linear:*

(a) *f is strictly twice epi-differentiable at $\bar{x}$ for $\bar{v}$;*

(b) *e_λ has a strict Hessian at $\bar{x}$;*

(c) *∇e_λ is strictly differentiable at $\bar{x}$;*

(d) *e_λ is twice differentiable at $\bar{x}$, and $\nabla^2 e_\lambda(x) \to \nabla^2 e_\lambda(\bar{x})$ as $x \to \bar{x}$ in the set of points where e_λ is twice differentiable;*

(e) *e_λ is strictly twice epi-differentiable at $\bar{x}$ for $\bar{v}$;*

(f) *∇e_λ is strictly proto-differentiable at $\bar{x}$ for $\bar{v}$;*

(g) *P_λ is strictly differentiable at $\bar{x}$;*

(h) *P_λ is strictly proto-differentiable at $\bar{x}$;*

(i) *the f-attentive ϵ-localization T of ∂f at $(\bar{x}, \bar{v})$ is strictly proto-differentiable there.*

Proof. Theorem 2.3 as applied to e_λ in the context of the properties of this function in Theorem 2.4(b) yields the equivalence of (b) with (c) and more generally that of (e) with (f). In principle, (f) is a weaker property than (c), but we shall argue next that, in this context, (f) implies (c) through (d).

The linear mappings $\xi \mapsto \nabla^2 e_\lambda(x)\xi$ at points x where e_λ is twice differentiable are limits of difference quotient mappings for ∇e_λ at such points, which in particular can be construed as graphical limits. The definition of strict proto-differentiability of ∇e_λ at $\bar{x}$ ensures (through diagonalization) that such linear mappings must graph-converge to the proto-derivative mapping $(\nabla e_\lambda)'_{\bar{x}}$ as $x \to \bar{x}$. Graphical convergence of linear mappings coincides with pointwise convergence of such mappings and requires the limit to be linear; thus it corresponds to convergence of the coefficient matrices. Therefore (f) implies (d).

When (d) holds, however, powerful information is provided through the fact in Theorem 2.4(b) that the mapping ∇e_λ is Lipschitz continuous around $\bar{x}$, hence differentiable almost everywhere in some neighborhood of $\bar{x}$ (Rademacher's theorem). This means that e_λ is twice differentiable at almost every x in some neighborhood of $\bar{x}$. The norms $|\nabla^2 e_\lambda(x)|$ are uniformly bounded in such a neighborhood. Indeed, any upper bound to these norms serves as a local Lipschitz constant for ∇e_λ, and vice versa. In this framework we see that for the locally Lipschitz continuous mapping $D_\lambda(x) = \nabla e_\lambda(x) - \nabla e_\lambda(\bar{x})$ condition (c) implies the existence for any $\epsilon > 0$ of $\delta > 0$ such that ϵ serves as a Lipschitz constant for D_λ on the δ-ball around $\bar{x}$. But this is equivalent to D_λ being strictly differentiable at $\bar{x}$ with the zero matrix as its Jacobian. Hence (d) implies (c).

So far we have established the equivalence of conditions (b)–(f). Clearly (g) and (h) are equivalent to (b) and (f) through the formula in Theorem 2.4 connecting P_λ with ∇e_λ. On the other hand, (a) and (i) are equivalent by Theorem 2.3. It remains only to argue that (i) corresponds to (f). This is true because strict proto-differentiability is a geometric property of the graph of a mapping and is preserved under any linear transformation of the graph. The formula connecting ∇e_λ with T in Theorem 2.4 sets up an invertible linear transformation between the graphs of these mappings.

Finally, note that the equivalence of (g) and (h) shows that for the mapping P_λ strict proto-differentiability automatically entails generalized linearity (actually true linearity) of the proto-derivative mapping. Since the graphs of T and P_λ are "isomorphic" under a certain linear transformation in graph space, the same must hold for T as well. Once we know that $T'_{\bar{x},\bar{v}}$ is generalized linear, we have that $f''_{\bar{x},\bar{v}}$ is generalized quadratic. □

THEOREM 4.2. *Under the hypothesis of Theorem* 4.1, *and with* T *the* f*-attentive* ϵ*-localization of* ∂f *at* $(\bar{x},\bar{v})$, *the following conditions can be added to the list of equivalences:*

(a) $f''_{x,v}$ *epi-converges (to something) as* $(x,v) \to (\bar{x},\bar{v})$ *with* $f(x) \to f(\bar{x})$ *in the set of pairs* (x,v) *with* $v \in \partial f(x)$ *for which* f *is twice epi-differentiable*;

(b) $(e_\lambda)''_x$ *epi-converges as* $x \to \bar{x}$ *in the set where* e_λ *is twice epi-differentiable*;

(c) $(\nabla e_\lambda)'_x$ *graph-converges as* $x \to \bar{x}$ *in the set where* ∇e_λ *is proto-differentiable*;

(d) $(\nabla P_\lambda)'_x$ *graph-converges as* $x \to \bar{x}$ *in the set where* P_λ *is proto-differentiable*;

(e) $T'_{x,v}$ *graph-converges as* $x \to \bar{x}$ *and* $v \to \bar{v}$ *in the set of pairs* (x,v) *with* $v \in T(x)$ *for which* T *is proto-differentiable*;

(f) $f''_{x,v}$ *epi-converges as* $(x,v) \to (\bar{x},\bar{v})$ *with* $f(x) \to f(\bar{x})$ *in the set of pairs* (x,v) *with* $v \in \partial f(x)$ *for which* f *is twice epi-differentiable and* $f''_{x,v}$ *is generalized quadratic*;

(g) $T'_{x,v}$ *graph-converges as* $x \to \bar{x}$ *and* $v \to \bar{v}$ *in the set of pairs* (x,v) *with* $v \in T(x)$ *for which* T *is proto-differentiable and* $T'_{x,v}$ *is generalized linear.*

Proof. The proof of Theorem 4.1 already shows that (c) can be added to the list of equivalences. We have (c) equivalent to (d) and (e) because of the formulas relating these mappings in Theorem 2.4: not only proto-differentiability but also proto-derivatives correspond when the graphs of mappings can be identified under an invertible linear transformation. Under our assumption that f is prox-regular at $\bar{x}$ for $\bar{v}$ with respect to ϵ and r, we know that $T + rI$ is monotone (Theorem 2.2), so that $T'_{x,v} + rI$ is monotone when the proto-derivative exists. But by Theorem 2.3, $T'_{x,v}$ is the subgradient mapping for $\frac{1}{2}\big(f''_{x,v} + r|\cdot|^2\big)$ so its monotonicity implies the convexity of $\frac{1}{2}\big(f''_{x,v} + r|\cdot|^2\big)$ [19]. Graphical convergence of $T'_{x,v} + rI$ as $(x,v) \in (\bar{x},\bar{v})$ with

$v \in T(x)$ is equivalent then by Attouch's theorem to epi-convergence of $\frac{1}{2}\big(f''_{x,v} + r|\cdot|^2\big)$ as $(x,v) \in (\bar{x},\bar{v})$ with $v \in T(x)$; the latter is the same as having $(x,v) \in (\bar{x},\bar{v})$ with $v \in \partial f(x)$ and $f(x) \to f(\bar{x})$ by [25, Prop. 2.3]. Hence (e) is equivalent to (a).

Similarly by Attouch's theorem, we have the equivalence of (c) with (b) because this likewise can be reduced to the convex case because of the convexity property asserted in Theorem 2.4(b). Finally, (f) and (g) come into the picture as viable substitutes for (a) and (e), respectively, because the graph of T is a Lipschitz manifold of dimension n around $(\bar{x},\bar{v})$ [25, Thm. 4.7] and consequently has a linear tangent space at almost every pair (x,v) in some neighborhood of $(\bar{x},\bar{v})$; cf. [32]. At such (x,v), this subspace is the graph of $T'_{x,v}$, which therefore is generalized linear, and the function $\frac{1}{2}f''_{x,v}$ having $T'_{x,v}$ as its subgradient mapping is accordingly a generalized quadratic function. □

Again we note, as in Corollary 3.11, that if one of the λ-dependent properties in Theorems 4.1 and 4.2 holds for some $\lambda \in (0, 1/r)$, they all hold for *every* $\lambda \in (0, 1/r)$.

COROLLARY 4.3. *Suppose that f is prox-regular at a point $\hat{x}$ for $\hat{v}$ with constants ϵ and r. Then there is a neighborhood U of $(\hat{x},\hat{v})$ such that for all $(x,v) \in \operatorname{gph} T \cap U$ (where T is the f-attentive ϵ-localization of ∂f at $(\hat{x},\hat{v})$) the following properties are equivalent and imply that $f''_{x,v}$ is generalized quadratic and $T'_{x,v}$ is generalized linear:*

(a) *f is strictly twice epi-differentiable at x for v;*

(b) *T is strictly proto-differentiable at x for v;*

(c) *$f''_{x',v'}$ epi-converges (to something) as $(x',v') \to (x,v)$ with $f(x') \to f(x)$ in the set of pairs (x',v') with $v' \in \partial f(x')$ for which f is twice epi-differentiable;*

(d) *$T'_{x',v'}$ graph-converges (to something) as $x' \to x$ and $v' \to v$ in the set of pairs (x',v') with $v' \in T(x')$ for which T is proto-differentiable;*

(e) *$f''_{x',v'}$ epi-converges (to something) as $(x',v') \to (x,v)$ with $f(x') \to f(x)$ in the set of pairs (x',v') with $v' \in \partial f(x')$ for which f is twice epi-differentiable and $f''_{x',v'}$ is generalized quadratic;*

(f) *$T'_{x',v'}$ graph-converges (to something) as $x' \to x$ and $v' \to v$ in the set of pairs (x',v') with $v' \in T(x')$ for which T is proto-differentiable and $T'_{x',v'}$ is generalized linear.*

Proof. The proof is similar to that of Corollary 3.10 except that at the end we use Theorems 4.1 and 4.2 instead. □

The existence of a strict Hessian for e_λ at $\bar{x}$ is a very natural property to look for in any numerical method for minimizing e_λ that relies on second-order approximations. It is remarkable that this property is equivalent to so many others in Theorems 4.1 and 4.2, especially those of f and the localization T of ∂f. In taking any step further toward stability or continuity properties of generalized derivatives of f it appears one has to pass to the case described in the next theorem, our last.

We have already made use of the fact proved in [25, Thm. 4.7] that when f is prox-regular at $\bar{x}$ for $\bar{v}$, an f-attentive localization of the graph of ∂f around $(\bar{x},\bar{v})$ is a Lipschitz manifold. Such a manifold, as defined in [32], can be identified locally under some invertible linear transformation from $\mathbb{R}^n \times \mathbb{R}^n$ onto itself with the graph of a Lipschitz continuous mapping. Similarly we can speak of a *smooth manifold* in the case where identification is made with the graph of a $\mathcal{C}^1$ mapping.

THEOREM 4.4. *Suppose that f is prox-regular at $\bar{x} \in \operatorname{argmin} f$ for $\bar{v} = 0$ with respect to ϵ and r. Let T be the f-attentive ϵ-localization of ∂f at $(\bar{x},\bar{v})$. Then there exists $\lambda_0 > 0$ such that for all $0 < \lambda < \lambda_0$ the following properties are equivalent:*

(a) *e_λ is $\mathcal{C}^2$ on a neighborhood of $\bar{x}$;*

(b) *e_λ is strictly twice epi-differentiable on a neighborhood of $\bar{x}$;*

(c) *P_λ is $\mathcal{C}^1$ on a neighborhood of $\bar{x}$;*

(d) *P_λ is strictly proto-differentiable on a neighborhood of $\bar{x}$;*

(e) *f is twice epi-differentiable at x for v with respect to all (x,v) sufficiently near to $(\bar{x},\bar{v})$ in the graph of T; moreover, $f''_{x,v}$ depends epi-continuously on (x,v); i.e., $f''_{x',v'}$ epi-converges to $f''_{x,v}$ as $(x',v') \to (x,v)$ with $f(x') \to f(x)$ and $v' \in \partial f(x')$;*

(f) *f is strictly twice epi-differentiable at x for v with respect to all (x,v) sufficiently near to $(\bar{x},\bar{v})$ in the graph of T;*

(g) *the graph of T is a smooth manifold relative to some neighborhood of $(\bar{x},\bar{v})$;*

(h) *T is strictly proto-differentiable at x for v with respect to all (x,v) sufficiently near to $(\bar{x},\bar{v})$ in the graph of T.*

Proof. We can assume that $\bar{x}=0$, $f(\bar{x})=0$. Fix x and $v \in \partial f(x)$ with $|x|<\epsilon/4$, $|v|<\epsilon$, and $|f(x)|<\epsilon$. As in the proof of Theorem 3.9, we take $R>r$ sufficiently large so that $x \in \operatorname{argmin} \bar{f}$ where $\bar{f}(x') := f(x') - \langle v, x'-x\rangle + (R/2)|x'-x|^2$. Note that $\bar{f}$ is prox-regular at x for 0. Again from the proof of Theorem 3.9 we have

$$\bar{f}_\lambda(w) = e_\mu(u) - \frac{1}{2\mu}|u|^2 + \frac{1}{2\lambda}|w|^2 + \langle v, x\rangle + \frac{R}{2}|x|^2,$$

where $\mu = \lambda/(\lambda R+1)$ and $u = \mu[v + Rx + (1/\lambda)w]$.

From the formula for $\bar{f}_\lambda$ we see that $\bar{f}_\lambda$ is strictly twice epi-differentiable at x for 0 if and only if e_μ is strictly twice epi-differentiable at $\mu[v+Rx+(1/\lambda)x] = \mu v + x$ for v (since by Theorem 2.4, $\nabla e_\mu(x+\mu v) = v$). Besides, $\bar{f}$ is strictly twice epi-differentiable at x for 0 if and only if f is strictly twice epi-differentiable at x for v.

Now we apply the equivalent properties in Theorems 4.1 and 4.2 not just at $(\bar{x},\bar{v})$ but locally, keeping in mind that a mapping is $\mathcal{C}^1$ on an open set if and only if it is strictly differentiable there, and the fact that there is a one-to-one correspondence between the points x' close to 0 and the points $x+\mu v$ with $v \in T(x)$ (by Theorem 2.4). The local identification of the graph of T with a Lipschitz manifold is achieved through parameterization by P_λ, so the smoothness of P_λ under the present circumstances corresponds to having the graph of T be in fact a smooth manifold. □

COROLLARY 4.5. *If f is $\mathcal{C}^2$ on a neighborhood of a point $\bar{x} \in \operatorname{argmin} f$, then for all $\lambda>0$ sufficiently small, the function e_λ is $\mathcal{C}^2$ on a neighborhood of $\bar{x}$.*

COROLLARY 4.6. *Suppose f is lower-$\mathcal{C}^2$ on a neighborhood of $\bar{x} \in \operatorname{argmin} f$ and there exists $s>0$ such that $f(x) \le f(\bar{x}) + s|x-\bar{x}|^2$ for all x in some neighborhood of $\bar{x}$. Then for $\lambda>0$ sufficiently small, e_λ is $\mathcal{C}^2$ around $\bar{x}$ if and only if f itself is $\mathcal{C}^2$ around $\bar{x}$.*

Proof. By adding $s_0|x-\bar{x}|^2$ to f for sufficiently high s_0, we can reduce to the case where f is actually convex on a neighborhood of $\bar{x}$. As already observed in the proof of Corollary 3.12, the bound on the growth of f around $\bar{x}$ has ensured that if the second-order epi-derivative function $f''_{\bar{x},\bar{v}}$ exists it must be finite. But when f is convex, its derivatives $f''_{x,v}$ are convex as well. Such a function, being positively homogeneous of degree 2, is finite everywhere if it is finite on a neighborhood of 0. When convex functions epi-converge to a convex function that is finite on a neighborhood of 0, they must eventually themselves be finite on a neighborhood of 0; cf. [38]. In the setting of Theorem 4.4(c), therefore, we must have finite quadratic, convex functions that depend epi-continuously on (x,v). In light of Theorem 3.1, this is the same as f being $\mathcal{C}^2$ around $\bar{x}$. □

COROLLARY 4.7. *A function f is $\mathcal{C}^2$ on a neighborhood of a point $\hat{x}$ if and only if f is prox-regular at $\hat{x}$ and there exist $\hat{v} \in \partial f(\hat{x})$ and $\epsilon>0$ such that whenever*

$$v \in \partial f(x) \text{ with } |v-\hat{v}|<\epsilon, \quad |x-\hat{x}|<\epsilon, \quad f(x) + f(\hat{x}) + \epsilon,$$

f is strictly twice epi-differentiable at x for v, the function $f''_{\hat{x},\hat{v}}$ itself being finite. Then the relation $v \in \partial f(x)$ reduces locally to $v = \nabla f(x)$, while $f''_{x,v}(\xi) = \langle \xi, \nabla^2 f(x)\xi \rangle$.

Proof. The necessity is obvious. For the sufficiency we can use the fact that $\hat{v}$ is in particular a proximal subgradient of f at $\hat{x}$ to add a quadratic function to f so as to reduce to the case where $\hat{x} \in \operatorname{argmin} f$ and $\hat{v} = 0$. Then the equivalences in Theorem 4.4 come into force. The finiteness of $f''_{\hat{x},\hat{v}}$ puts us at the same time in the picture of Theorems 3.1 and 3.2; the generalized quadratic functions $f''_{x,v}$ for (x, v) near enough to $(\hat{x}, \hat{v})$ have to be ordinary quadratic functions. □

REFERENCES

[1] J. F. BONNANS, J.-CH. GILBERT, C. LEMARÉCHAL, AND C. A. SAGASTIZÁBAL, *A family of variable metric proximal methods*, Math. Programming, 68 (1995), pp. 15–47.

[2] J. M. BORWEIN AND D. NOLL, *Second-order differentiability of convex functions in Banach spaces*, Trans. Amer. Math. Soc., 342 (1994), pp. 43–81.

[3] J. V. BURKE AND R. A. POLIQUIN, *Optimality conditions for non-finite convex composite functions*, Math. Programming, 57 (1992), pp. 103–120.

[4] F. H. CLARKE, *Generalized gradients and applications*, Trans. Amer. Math. Soc., 205 (1975), pp. 247–262.

[5] X. CHEN, L. QI, AND R. WOMERSLEY, *Newton's method for quadratic stochastic programs with recourse*, J. Comput. Appl. Math., 60 (1995), pp. 29–46.

[6] R. COMINETTI AND R. CORREA, *A generalized second-order derivative in nonsmooth optimization*, SIAM J. Control Optim., 28 (1990), pp. 789–809.

[7] S. DOLECKI, G. SALINETTI, AND R. J.-B. WETS, *Convergence of functions: Equi-semicontinuity*, Trans. Amer. Math. Soc., 276 (1983), pp. 409–429.

[8] F. FACCHINEI, *Minimization of SC^1 functions and the Maratos effect*, Oper. Res. Lett., 17 (1995), pp. 131–137.

[9] M. FUKUSHIMA AND L. QI, *A Globally and Superlinearly Convergent Algorithm for Nonsmooth Convex Minimization*, Applied Math. Report, AMR 95/1, School of Mathematics, University of New South Wales, Sydney, Australia, 1995.

[10] J.-B. HIRIART-URRUTY, *Characterization of the plenary hull of the generalized Jacobian matrix*, Math. Prog. Study, 17 (1982), pp. 1–12.

[11] A. D. IOFFE, *Variational analysis of a composite function: A formula for the lower second order epi-derivative*, J. Math. Anal. Appl., 160 (1991), pp. 379–405.

[12] V. JEYAKUMAR AND X. Q. YANG, *Convex composite minimization with $C^{1,1}$ functions*, J. Optim. Theory Appl., 86 (1995), pp. 631–648.

[13] C. LEMARÉCHAL AND C. SAGASTIZÁBAL, *Practical Aspects of the Moreau-Yosida Regularization I: Theoretical Properties*, 1994, INRIA, Rocquencourt, France, preprint.

[14] A. LEVY, R. A. POLIQUIN, AND L. THIBAULT, *Partial extension of Attouch's theorem with applications to proto-derivatives of subgradient mappings*, Trans. Amer. Math. Soc., 347 (1995), pp. 1269–1294.

[15] A. LEVY AND R. T. ROCKAFELLAR, *Variational conditions and the proto-differentiation of partial subgradient mappings*, Nonlinear Anal. Theory Methods Appl., to appear.

[16] P. D. LOEWEN, *Optimal Control via Nonsmooth Analysis*, CRM-AMS Lecture Notes in Mathematics 2, American Mathematical Society, Providence, RI, 1993.

[17] Z. PÁLES AND V. ZEIDAN, *Generalized Hessian for C^{1+} functions in infinite dimensional normed spaces*, 1994, Department of Mathematics, Michigan State University, East Lansing, MI, preprint.

[18] J. S. PANG AND L. QI, *A globally convergent Newton method for convex SC^1 minimization problems*, J. Optim. Theory Appl., 85 (1995), pp. 633–648.

[19] R. A. POLIQUIN, *Subgradient monotonicity and convex functions*, Nonlinear Anal. Theory Methods Appl., 14 (1990), pp. 305–317.

[20] ———, *Integration of subdifferentials of nonconvex functions*, Nonlinear Anal. Theory Methods Appl., 17 (1991), pp. 385–398.

[21] ———, *An extension of Attouch's Theorem and its application to second-order epi-differentiation of convexly composite functions*, Trans. Amer. Math. Soc., 332 (1992), pp. 861–874.

[22] R. A. POLIQUIN AND R. T. ROCKAFELLAR, *Amenable functions in optimization*, in Nonsmooth Optimization Methods and Applications, F. Giannessi (ed.), Gordon and Breach, Philadelphia, 1992, pp. 338–353.

[23] ———, *A calculus of epi-derivatives applicable to optimization*, Canad. J. Math., 45 (1993), pp. 879–896.

[24] ———, *Proto-derivative formulas for basic subgradient mappings in mathematical programming*, Set-Valued Anal., 2 (1994), pp. 275–290.

[25] ———, *Prox-regular functions in variational analysis*, Trans. Amer. Math. Soc., 348 (1996), pp. 1805–1838.

[26] ———, *Second-order nonsmooth analysis in nonlinear programming*, in Recent Advances in Optimization, D. Du, L. Qi, and R. Womersley, eds., World Scientific Publishers, River Edge, NJ, 1995.

[27] R. A. POLIQUIN, J. VANDERWERFF, AND V. ZIZLER, *Convex composite representation of lower semicontinuous functions and renormings*, C. R. Acad. Sci. Paris Sér. I, 317 (1993), pp. 545–549.

[28] L. QI, *Superlinearly convergent approximate Newton methods for LC^1 optimization problems*, Math. Programming, 64 (1994), pp. 277–294.

[29] L. QI AND R. WOMERSLEY, *An SQP algorithm for extended linear-quadratic problems in stochastic programming*, Ann. Oper. Res., 56 (1995), pp. 251–285.

[30] L. QI, *Second-order analysis of the Moreau-Yosida approximation of a convex function*, Applied Mathematics Report, AMR 94/20, School of Mathematics, University of New South Wales, Sydney, Australia, 1994.

[31] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[32] ———, *Maximal monotone relations and the second derivatives of nonsmooth functions*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 2 (1985), pp. 167–184.

[33] ———, *First- and second-order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 307 (1988), pp. 75–107.

[34] ———, *Proto-differentiability of set-valued mappings and its applications in optimization*, in Analyse Non Linéaire, H. Attouch et al. (eds.), Gauthier-Villars, Paris (1989), pp. 449–482.

[35] ———, *Second-order optimality conditions in nonlinear programming obtained by way of epi-derivatives*, Math. Oper. Res., 14 (1989), pp. 462–484.

[36] ———, *Generalized second derivatives of convex functions and saddle functions*, Trans. Amer. Math. Soc., 320 (1990), pp. 810–822.

[37] ———, *Favorable classes of Lipschitz continuous functions in subgradient optimization*, in Progress in Nondifferentiable Optimization, E. Nurminski, ed., IIASA Collaborative Proceedings Series, IIASA, Laxenburg, Austria, 1982, pp. 125–144.

[38] G. SALINETTI AND R. J-B WETS, *On the relation between two types of convergence for convex functions*, J. Math. Anal. Appl., 60 (1977), pp. 211–226.

[39] X. Q. YANG AND V. JEYAKUMAR, *Generalized second-order directional derivatives and optimization with $C^{1,1}$ functions*, Optimization, 26 (1992), pp. 165–185.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1138–1152, November 1996
© 1996 Society for Industrial and Applied Mathematics
014

RATES OF CONVERGENCE IN STOCHASTIC PROGRAMS WITH COMPLETE INTEGER RECOURSE*

RÜDIGER SCHULTZ†

Abstract. The stability of stochastic programs with mixed-integer recourse and random right-hand sides under perturbations of the integrating probability measure is considered from a quantitative viewpoint. Objective-function values of perturbed stochastic programs are related to each other via a variational distance of probability measures based on a suitable Vapnik–Červonenkis class of Borel sets in a Euclidean space. This leads to Hölder continuity of local optimal values. In the context of estimation via empirical measures the general results imply qualitative and quantitative statements on the asymptotic convergence of local optimal values and optimal solutions.

Key words. stochastic integer programming, parametric integer programming, Hölder continuity, stability, variational distance of probability measures, Vapnik–Červonenkis class, law of the iterated logarithm

AMS subject classifications. 90C15, 90C11, 90C31

1. Introduction.

Consider the following two-stage stochastic integer program:

$$\min\{f(x) + Q(x,\mu) : x \in C\}, \tag{1}$$

where

$$Q(x,\mu) = \int_{\mathbb{R}^s} \Phi(z - Ax)\mu(\mathrm{d}z) \tag{2}$$

and

$$\Phi(b) = \min\{q^T y + q'^T y' : Wy + W'y' = b,\ y' \geq 0,\ y \geq 0,\ y' \in \mathbb{R}^{s'},\ y \in \mathbb{Z}^{\bar{s}}\}. \tag{3}$$

Generally, we assume that $f : \mathbb{R}^m \to \mathbb{R}$ is continuous; $C \subset \mathbb{R}^m$ nonempty, closed; $q \in \mathbb{R}^{\bar{s}}$; $q' \in \mathbb{R}^{s'}$; that $W \in L(\mathbb{R}^{\bar{s}}, \mathbb{R}^s)$, $W' \in L(\mathbb{R}^{s'}, \mathbb{R}^s)$ are matrices with rational entries; and that μ belongs to $\mathcal{P}(\mathbb{R}^s)$, the set of all Borel probability measures on $\mathbb{R}^s$. Throughout, $\mathbb{Z}$ denotes the set of integers.

The model (1) arises from a minimization problem with uncertain constraint parameters whose realizations are not known when having to fix the (first-stage) decision variable x. Infeasibilities b occurring after the realization of the uncertain parameters can be compensated at cost $\Phi(b)$ by the second-stage optimization procedure (3). Altogether, (1) aims at finding a first-stage decision x such that the sum of the first-stage costs $f(x)$ and the expected compensation (or recourse) costs $Q(x,\mu)$ become minimal. Of course, for the latter to be well defined one needs (at least) a probability distribution of the uncertain parameters.

The above model essentially differs from traditional two-stage stochastic programs (cf. [12], [40]) by the integrality constraints in the second stage. Whereas integrality in the first stage can be dealt with by fairly conventional means [41], its presence in the second stage is much more cumbersome since the integrand Φ in (2) is discontinuous. However, there are several examples in the literature showing that integrality of

* Received by the editors July 25, 1994; accepted for publication (in revised form) June 9, 1995. This research was supported by the Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung" of the Deutsche Forschungsgemeinschaft. Parts of the paper were written while the author was an assistant at the Department of Mathematics at Humboldt University, Berlin.

† Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustrasse 7, D-14195 Berlin, Germany (schultz@zib.de).

second-stage decisions has to or at least should be included into the model [20], [26], [34].

Structural properties of stochastic programs with integer recourse have been studied in [1], [17], [21], [26], [31], [32], [33], [36]. The present paper extends convergence results from [1] and [33] by rates of convergence with respect to suitable distances of probability measures. More specifically, we consider (1) as a parametric program with the parameter μ varying in some (metric) space of probability measures. Then we derive Hölder estimates for Q (cf. (2)) as a function of the integrating probability measure which leads to corresponding results for local optimal values of (1). Our analysis is motivated by the incomplete information on the underlying measures μ that is often encountered (e.g., [10]). If μ is approximated by empirical measures then our general results specify to qualitative and quantitative results on the asymptotic convergence of (local) optimal values and optimal solutions. Another motivation for studying (1) as a parametric program in μ is given by numerical techniques that rely on approximating μ by simpler measures [6], [13].

Qualitative (i.e., continuity) results for Q as a function of μ were obtained in [1], [19], [33]. In these papers, $\mathcal{P}(\mathbb{R}^s)$ was equipped with weak convergence of probability measures, a notion covering many specific applications. Recall that a sequence $\{\mu_n\}$ of probability measures in $\mathcal{P}(\mathbb{R}^s)$ converges weakly to $\mu \in \mathcal{P}(\mathbb{R}^s)$ if $\int_{\mathbb{R}^s} g(z)\mu_n(\mathrm{d}z) \underset{n\to\infty}{\longrightarrow} \int_{\mathbb{R}^s} g(z)\mu(\mathrm{d}z)$ for any bounded continuous function $g : \mathbb{R}^s \to \mathbb{R}$ [5].

The key problem whose solution is pushed ahead in the present paper is the choice of a probability distance for quantitative stability of mixed-integer recourse stochastic programs. We present a distance that, on the one hand, fits to the discontinuous integrand Φ and, on the other hand, covers weak convergence of probability measures under mild hypotheses. Although we are not able to prove that our distance is in some respect the finest one allowing a quantitative stability analysis of stochastic integer programs, it already covers a substantial share of specific instances due to its relation to weak convergence of probability measures. The difficulty of selecting such a distance is illustrated by an example in [33] showing that metrics like the Wasserstein distance [25] that have led to convergence rates for $Q(x,.)$ in the noninteger recourse case [29] fail when it comes to integer recourse.

Our distance is a specific variational distance (or discrepancy) of probability measures that is given by

$$\alpha_{\mathcal{B}_\mathcal{K}} := \sup\{|\mu(B) - \nu(B)| : \ B \in \mathcal{B}_\mathcal{K}\},$$

where $\mathcal{B}_\mathcal{K}$ is a class of Borel sets related to Φ. It will be specified in §3. If the second stage (3) is a pure integer linear program, then $\mathcal{B}_\mathcal{K}$ can be taken as the class of all lower left orthants in $\mathbb{R}^s$, and $\alpha_{\mathcal{B}_\mathcal{K}}$ coincides with the uniform (or Kolmogorov–Smirnov) distance of distribution functions. In the general case, $\mathcal{B}_\mathcal{K}$ still has the pleasant property to be a Vapnik–Červonenkis class of subsets of $\mathbb{R}^s$, which allows some interesting conclusions on the asymptotic behaviour of estimators based on empirical measures (see §5 for recalling the concept of a Vapnik–Červonenkis class and for further details).

Our paper is organized as follows. In §2 we collect some prerequisites on the value function Φ (cf. (3)) from parametric integer programming. Section 3 contains the central Hölder estimates for $Q(x,.)$. Consequences for the quantitative stability of local optimal values of (1) are derived in §4. In §5 we elaborate the special case where μ is estimated via empirical measures. Finally we have a conclusions section.

Throughout, $\|.\|$ denotes the ℓ^∞-norm in the Euclidean space under consideration, $\boldsymbol{B}(\xi, r)$ denotes the closed ball around ξ with radius r (with respect to $\|.\|$).

2. Properties of the value function. As a prerequisite for our subsequent considerations we have to collect some properties of the value function Φ (cf. (3)). The basic literature in this respect consists of [2], [3], [7]. Let us assume that for each $b \in \mathbb{R}^s$ the constraint set $M(b) \subseteq \mathbb{Z}^{\bar{s}} \times \mathbb{R}^{s'}$ in (3) is nonempty and that $\Phi(0) = 0$. Then $\Phi(b) \in \mathbb{R}$ for all $b \in \mathbb{R}^s$ (cf., e.g., [22], Prop. 1.6.7). Since Φ is discontinuous in general it is interesting to ask for continuity regions. Denoting by $\mathrm{pr}_1(M(b))$ the projection of $M(b)$ to $\mathbb{Z}^{\bar{s}}$ we have the following result.

LEMMA 2.1 (see [2, Thm. 3.4.3, Thm. 5.6.5]). *The restrictions of Φ to subsets of $\mathbb{R}^s$ where $pr_1(M(b))$ does not depend on b are continuous.*

Subsets of $\mathbb{R}^s$ where the projection $\mathrm{pr}_1(M(.))$ is constant in the sense that $\mathrm{pr}_1(M(b))$ does not depend on b can be described via $\mathcal{K} := W'(\mathbb{R}^{s'}_+)$, the positive span of W':

LEMMA 2.2 (see [2, Lem. 5.6.1, Lem. 5.6.2, Thm. 5.6.3]). *There exists a countable partition $\mathbb{R}^s = \bigcup_{i=1}^{\infty} \mathcal{B}_i$ such that*

(i) *$pr_1(M(.))$ is constant on $\mathcal{B}_i$ and, hence, $\Phi\big|_{\mathcal{B}_i}$ is continuous;*

(ii) *each of the sets $\mathcal{B}_i$ has a representation $\mathcal{B}_i = \{b_i + \mathcal{K}\} \setminus \bigcup_{j=1}^{N_o} \{b_{ij} + \mathcal{K}\}$, where $b_i, b_{ij} \in \mathbb{R}^s$ $(i \in \mathbb{N},\ j = 1, \ldots, N_o)$ and N_o does not depend on i.*

In [2] the rationality of both W and W' is employed to establish the above result. The authors first show that $\mathcal{B}_i$ can be represented as the (set-theoretic) difference of an (infinite) intersection and an (infinite) union of polyhedral cones. Utilizing the rationality of W they show that the infinite intersection can be replaced by just one cone; the rationality of W' implies that the infinite union of cones can be replaced by a finite one. In this context, we refer to an example in [2, p. 58] where some pathologies are illustrated that can occur if W contains irrational elements.

Lemma 2.2 implies a representation of $\Phi\big|_{\mathcal{B}_i}$. It holds that

$$\begin{aligned}(4)\qquad \Phi\big|_{\mathcal{B}_i}(b) &= \min\{q^T y + q'^T y' : W'y' = b - Wy;\ y \in \mathrm{pr}_1(M(b)),\ y' \geq 0\} \\ &= \min\{q^T y + \tilde{\Phi}(b - Wy) : y \in \mathrm{pr}_1(M(b))\},\end{aligned}$$

where $\tilde{\Phi}(\tilde{b}) := \min\{q'^T y' : W'y' = \tilde{b},\ y' \geq 0\}$ denotes a value function of a linear program with parameters in the right-hand side of the constraints. The assumptions on Φ imposed at the beginning imply that $\tilde{\Phi}(\tilde{b}) \in \mathbb{R}$ for any $\tilde{b} \in \mathrm{pos}\, W' = \mathcal{K}$. Furthermore, it is well known from the literature [39] that $\tilde{\Phi}\big|_{\mathcal{K}}$ admits a representation

$$\tilde{\Phi}\big|_{\mathcal{K}}(\tilde{b}) = \max_{i=1,\ldots,\tilde{N}} \tilde{d}_i^T \tilde{b},$$

where $\tilde{d}_i$ $(i = 1, \ldots, \tilde{N})$ are determined by q' and W'.

Thus, the representation (4) says that $\Phi\big|_{\mathcal{B}_i}$ is the pointwise minimum of countably many continuous, piecewise linear functions. Moreover, the above argument proves Lemma 2.3.

LEMMA 2.3. *$\Phi\big|_{\mathcal{B}_i}$ is Lipschitz continuous with a constant $L_o > 0$ not depending on i.*

Let us now turn to some proximity results for the value function Φ and for optimal solutions of (3).

LEMMA 2.4 (see [3, Thm. 8.1]; [7, Thm. 2.1]). *There exist constants $\beta_1 > 0$, $\beta_2 > 0$, $\gamma_1 > 0$, $\gamma_2 > 0$ such that for all $b_1, b_2 \in \mathbb{R}^s$ we have*

(i) $|\Phi(b_1) - \Phi(b_2)| \leq \beta_1 \|b_1 - b_2\| + \gamma_1$,

(ii) *if* $(y_1, y_1') \in \mathbb{Z}^{\bar{s}} \times \mathbb{R}^{s'}$ *is optimal in* (3) *with respect to parameter* b_1, *then there exists a* $(y_2, y_2') \in \mathbb{Z}^{\bar{s}} \times \mathbb{R}^{s'}$ *optimal with respect to* b_2 *such that*

$$\|(y_1, y_1') - (y_2, y_2')\| \leq \beta_2 \|b_1 - b_2\| + \gamma_2.$$

A first consequence of the above lemma which will turn out to be useful later on is the following.

LEMMA 2.5. *There exists a number* $N_1 \in \mathbb{N}$ *such that for any* $b \in \mathbb{R}^s$ *the ball* $\boldsymbol{B}(b, 1)$ *is intersected by at most* N_1 *different continuity regions* $\mathcal{B}_i$ *of* Φ.

Proof. Let $b_o \in \mathbb{R}^s$ be arbitrary and (y_o, y_o') be optimal in (3) with respect to b_o. Let $b \in \boldsymbol{B}(b_o, 1)$. By Lemma 2.4(ii) there exists a $(y, y') \in \mathbb{Z}^{\bar{s}} \times \mathbb{R}^{s'}$ optimal with respect to b such that

$$\|(y_o, y_o') - (y, y')\| \leq \beta_2 \|b_o - b\| + \gamma_2 \leq \beta_2 + \gamma_2 =: r_o;$$

i.e., for any $b \in \boldsymbol{B}(b_o, 1)$ there exists an optimal (y, y') whose y-component belongs to $\boldsymbol{B}(y_o, r_o) \cap \mathbb{Z}^{\bar{s}}$.

Recalling the representation (4) we obtain that for any $b \in \boldsymbol{B}(b_o, 1)$

$$\Phi(b) = \min\{q^T y + \tilde{\Phi}(b - Wy) : y \in \mathrm{pr}_1(M(b)) \cap \boldsymbol{B}(y_o, r_o)\}.$$

Since there are only finitely many different subsets of $\boldsymbol{B}(y_o, r_o) \cap \mathbb{Z}^{\bar{s}}$ the proof is complete. □

A situation deserving special attention is that of pure integer recourse; i.e., Φ is given by

$$\Phi(b) = \min\{q^T y : Wy \geq b, y \in \mathbb{Z}_+^{\bar{s}}\}.$$

Introducing slack variables $y' \in \mathbb{R}^s$ this fits into the above setting, and we have $\mathcal{K} = -\mathbb{R}_+^s$. Furthermore, $q' = 0$, which leads to $\tilde{\Phi}(\tilde{b}) = 0$ for all $\tilde{b} \in \mathcal{K}$, and Lemma 2.3 specifies to $\Phi|_{\mathcal{B}_i}$ being constant for all i.

3. Rates of convergence for expectation functions. The following basic assumptions are imposed throughout. They guarantee that (2) and (3) are well defined.

(A1) For all $b \in \mathbb{R}^s$ there exist $y \in \mathbb{Z}^{\bar{s}}$, $y' \in \mathbb{R}^{s'}$ such that $y \geq 0$, $y' \geq 0$, and $Wy + W'y' = b$.

(A2) There exists a $u \in \mathbb{R}^s$ such that $W^T u \leq q$ and $W'^T u \leq q'$.

(A3) It holds that $\int_{\mathbb{R}^s} \|z\| \mu(\mathrm{d}z) < +\infty$.

Note that these assumptions can be read as natural extensions of counterparts for linear stochastic programs with noninteger recourse [12], [40]. In [33] it is shown that (A1)–(A3) imply that $Q(., \mu)$ is a lower semicontinuous real-valued function on $\mathbb{R}^m$. Let us further remark that (A2) is equivalent to $\Phi(0) = 0$ (which appeared as an assumption in §2) and that (A1) in particular implies that $\mathcal{K} = \mathrm{pos}\, W'$ has a nonempty interior.

We now address the problem of finding upper estimates for $|Q(x, \mu) - Q(x, \nu)|$ in terms of the underlying probability measures $\mu, \nu \in \mathcal{P}(\mathbb{R}^s)$. Let us note that Proposition 3.4 in [33] contains a sufficient condition for the joint continuity of Q in x and μ with respect to the product topology of the usual one on $\mathbb{R}^m$ and that of weak convergence on $\mathcal{P}(\mathbb{R}^s)$.

Example 3.1 in [33] illustrates the difficulties in finding a distance function on $\mathcal{P}(\mathbb{R}^s)$ that fits both to weak convergence and to the discontinuities in Φ. If the

second stage in (1) is a linear program with continuous variables then Φ is convex piecewise linear and globally Lipschitzian. This perfectly fits to the L_1-Wasserstein distance of probability measures (see [25]) that, via the Kantorovich–Rubinstein duality theorem, can be computed as a supremum of (absolute values of) differences of integrals over the set of all Lipschitz functions with modulus less than or equal to 1 (see [29] for utilizing this in stability of stochastic programs). Discontinuities in Φ not only make such an approach technically impossible, but also, as shown by the mentioned example, prevent Hölder rates for $Q(.,\mu)$ with respect to the L_1-Wasserstein distance. This extends to other metrics like the Prokhorov or the β-metrics [9], [25] for which equivalence-type estimates by the L_1-Wasserstein metric exist. Therefore, it is impossible to extend techniques that are known to work in quantitative stability of stochastic programs with (continuous) linear recourse.

In what follows we will show that a certain discrepancy (or variational distance of probability measures) leads to Hölder estimates of $Q(x,.)$. Given a class $\mathcal{B}_o \subset \mathcal{B}(\mathbb{R}^s)$ of Borel sets in $\mathbb{R}^s$, the discrepancy $\alpha_{\mathcal{B}_o}(\mu,\nu)$ is defined by

$$\alpha_{\mathcal{B}_o}(\mu,\nu) := \sup\{|\mu(B) - \nu(B)| : \ B \in \mathcal{B}_o\}.$$

Popular instances of $\mathcal{B}_o$ in the literature are the families of all lower left orthants or of all convex Borel sets in $\mathbb{R}^s$ (cf., e.g., [4]). For us, a class in between these two families will be important: let $\mathcal{B}_\mathcal{K} \subseteq \mathcal{B}(\mathbb{R}^s)$ denote the class of all (closed) bounded polyhedra in $\mathbb{R}^s$ each of whose facets (i.e., $(s-1)$-dimensional faces) parallels a facet of $\mathcal{K} = \operatorname{pos} W'$ or a facet of $\bigtimes_{i=1}^s [0,1]$.

The discrepancy $\alpha_{\mathcal{B}_\mathcal{K}}$ is then even a metric on $\mathcal{P}(\mathbb{R}^s)$ which can be seen as follows: from the definition of a discrepancy $\alpha_{\mathcal{B}_o}$ we deduct the basic properties of a metric, except the one that $\alpha_{\mathcal{B}_o}(\mu,\nu) = 0$ implies $\mu = \nu$. Let $\mathcal{B}_{orth}$ denote the family of all (closed) lower left orthants in $\mathbb{R}^s$ and $\mathcal{B}_{box}$ denote the family of all (closed) bounded polyhedra in $\mathbb{R}^s$ whose facets are parallel to facets of $\bigtimes_{i=1}^s [0,1]$. Then it holds for all $\mu,\nu \in \mathcal{P}(\mathbb{R}^s)$ that

$$\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu) \geq \alpha_{\mathcal{B}_{box}}(\mu,\nu) \geq \alpha_{\mathcal{B}_{orth}}(\mu,\nu),$$

where the last inequality follows from the monotonicity of μ and ν on ascending sequences of sets. Therefore, $\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu) = 0$ implies $\alpha_{\mathcal{B}_{orth}}(\mu,\nu) = 0$. Since $\alpha_{\mathcal{B}_{orth}}$ is just the uniform (or Kolmogorov–Smirnov) distance of distribution functions (which is known to be a metric on $\mathcal{P}(\mathbb{R}^s)$, [25]), this implies $\mu = \nu$.

Let us further introduce

$$\Delta_{p,K}(\mathbb{R}^s) := \Big\{\nu \in \mathcal{P}(\mathbb{R}^s) : \int_{\mathbb{R}^s} \|z\|^p \nu(\mathrm{d}z) \leq K\Big\},$$

where $p > 1$ and $K > 0$ are fixed constants.

Then, the following estimate with respect to the discrepancy $\alpha(\mu,\nu) := \alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu)$ is valid.

PROPOSITION 3.1. *Suppose* (A1) *and* (A2) *and let* $D \subset \mathbb{R}^m$ *be nonempty and bounded. Then there exist constants* $L > 0$ *and* $\delta > 0$ *such that*

$$\sup_{x \in D} |Q(x,\mu) - Q(x,\nu)| \leq L \cdot \alpha(\mu,\nu)^{\frac{p-1}{p(s+1)}}$$

whenever $\mu,\nu \in \Delta_{p,K}(\mathbb{R}^s)$, $\alpha(\mu,\nu) < \delta$.

Proof. Let $x \in D$ be arbitrary and $\mu, \nu \in \Delta_{p,K}(\mathbb{R}^s)$ such that $\alpha(\mu,\nu) < \delta_o := 1$. The bound δ_o will be further shrunk in the course of the proof. Define the radii $R := \alpha(\mu,\nu)^{-1/p(s+1)}$ and $r := \alpha(\mu,\nu)^{\frac{p-1}{p(s+1)}}$. It holds that

$$(5)\quad |Q(x,\mu) - Q(x,\nu)| = \Big| \int_{\mathbb{R}^s} \Phi(z - Ax)(\mu - \nu)(\mathrm{d}z) \Big|$$
$$\leq \Big| \int_{Ax+B_o} \Phi(z - Ax)(\mu - \nu)(\mathrm{d}z) \Big| + \Big| \int_{\mathbb{R}^s \setminus \{Ax+B_o\}} \Phi(z - Ax)(\mu - \nu)(\mathrm{d}z) \Big|,$$

where $B_o = \boldsymbol{B}(0,R)$.

Now split B_o into $(l^\infty -)$ balls of radius r. The splitting is carried out in a disjunctive way, i.e., B_o is partitioned into disjoint (measurable) sets whose closures are closed balls with radius r. In the splitting, balls of radius r are used as long as possible; possible gaps are filled with maximal balls of radius less than r. Hence, the number of elements in the partition of B_o is bounded above by $(\frac{R}{r} + 1)^s \leq (\frac{2R}{r})^s$.

Each element of the above partition intersects certain continuity regions $\mathcal{B}_i$ of Φ (cf. Lemma 2.2); by Lemma 2.5 it intersects at most N_1 of such regions. From Lemma 2.2(ii) it can be seen that each $\mathcal{B}_i$ splits into disjoint (measurable) subsets whose closures are polyhedra with facets parallel to suitable facets of $\mathcal{K}$. Moreover, the number of such subsets can be bounded above by a constant only depending on N_o (cf. Lemma 2.2(ii)) and the number of facets of $\mathcal{K}$. The constant does not depend on i.

Altogether, B_o is split into disjoint subsets B_j $(j = 1, \ldots, N)$ whose closures are polyhedra with facets parallel to suitable facets of $\mathcal{K}$ or of $\bigtimes_{i=1}^s [0,1]$; i.e., the closures belong to $\mathcal{B}_\mathcal{K}$. Furthermore, there exists a constant $\kappa > 0$ which is independent on R and r such that $N \leq \kappa \cdot (\frac{R}{r})^s$. From each of the sets B_j $(j = 1, \ldots, N)$ we pick an element b_j and continue (5):

$$(6)\quad |Q(x,\mu) - Q(x,\nu)|$$
$$\leq \Big| \sum_{j=1}^N \int_{Ax+B_j} (\Phi(z - Ax) - \Phi(b_j))(\mu - \nu)(\mathrm{d}z) + \sum_{j=1}^N \int_{Ax+B_j} \Phi(b_j)(\mu - \nu)(\mathrm{d}z) \Big|$$
$$+ \Big| \int_{\mathbb{R}^s \setminus \{Ax+B_o\}} \Phi(z - Ax)(\mu - \nu)(\mathrm{d}z) \Big|$$
$$\leq \sum_{j=1}^N \Big(\int_{Ax+B_j} |\Phi(z - Ax) - \Phi(b_j)| \mu(\mathrm{d}z) + \int_{Ax+B_j} |\Phi(z - Ax) - \Phi(b_j)| \nu(\mathrm{d}z) \Big)$$
$$+ \sum_{j=1}^N |\Phi(b_j)| \cdot \Big| \int_{Ax+B_j} (\mu - \nu)(\mathrm{d}z) \Big| + \Big| \int_{\mathbb{R}^s \setminus \{Ax+B_o\}} \Phi(z - Ax)(\mu - \nu)(\mathrm{d}z) \Big|$$
$$\leq L_o \cdot 2r(\mu(Ax + B_o) + \nu(Ax + B_o))$$
$$+ (\beta_1 R + \gamma_1) \cdot N \cdot \alpha(\mu,\nu) + \Big| \int_{\mathbb{R}^s \setminus \{Ax+B_o\}} \Phi(z - Ax)(\mu - \nu)(\mathrm{d}z) \Big|.$$

The first member of the above sum results from Lemma 2.3 and the fact that $\mathrm{diam} B_j \leq 2r$ for all $j = 1, \ldots, N$. Concerning the second member we refer to Lemma 2.4(i) and the fact that $\Phi(0) = 0$. Furthermore for any $j = 1, \ldots, N$

$$|\mu(Ax + B_j) - \nu(Ax + B_j)| \leq \sup\{|\mu(B) - \nu(B)| : B \in \mathcal{B}_\mathcal{K}\},$$

where, if necessary, $Ax + B_j$ is approximated by a sequence that is monotonically increasing with respect to set inclusion and that consists of closed polyhedra from $\mathcal{B}_\mathcal{K}$ which are contained in the interior of $Ax + B_j$. The estimate is then gained by passing to the limit.

Now suppose in addition that $\alpha(\mu,\nu) \le \delta_1$, where δ_1 is selected such that $\beta_1 R \ge \gamma_1$ and

$$\mathbb{R}^s \setminus \{Ax + B_o\} \subset \left\{ z \in \mathbb{R}^s : \|z\| \ge \frac{1}{2}R \right\} \quad \text{for all} \quad x \in D$$

(note that D is bounded).

Then we can continue (6) as follows:

$$\begin{aligned}
(7) \quad & |Q(x,\mu) - Q(x,\nu)| \\
& \le 4L_o r + 2\beta_1 R \cdot N \cdot \alpha(\mu,\nu) + \int_{\|z\| \ge \frac{1}{2}R} |\Phi(z - Ax)| \mu(\mathrm{d}z) \\
& \quad + \int_{\|z\| \ge \frac{1}{2}R} |\Phi(z - Ax)| \nu(\mathrm{d}z) \\
& \le 4L_o r + 2\beta_1 R \cdot N \cdot \alpha(\mu,\nu) + \int_{\|z\| \ge \frac{1}{2}R} (\beta_1 \|z - Ax\| + \gamma_1)(\mu + \nu)(\mathrm{d}z) \\
& \qquad\qquad \text{(by Lemma 2.4(i) and } \Phi(0) = 0) \\
& \le 4L_o \cdot r + 2\beta_1 R \cdot N \cdot \alpha(\mu,\nu) + \int_{\|z\| \ge \frac{1}{2}R} (\beta_1 \|z\| + \beta_1 \|Ax\| + \gamma_1)(\mu + \nu)(\mathrm{d}z).
\end{aligned}$$

Let $\alpha(\mu,\nu) \le \delta_2$, where δ_2 is selected such that $\beta_1 \|z\| \ge \beta_1 \|Ax\| + \gamma_1$ for all $x \in D$ and all $\|z\| \ge \frac{1}{2}R$. Then we can continue:

$$\le 4L_o r + 2\beta_1 R \cdot N \cdot \alpha(\mu,\nu) + 2\beta_1 \int_{\|z\| \ge \frac{1}{2}R} \|z\| (\mu + \nu)(\mathrm{d}z).$$

Now recall that $\mu, \nu \in \Delta_{p,K}(\mathbb{R}^s)$. This implies

$$\begin{aligned}
2K &\ge \int_{\mathbb{R}^s} \|z\|^p (\mu + \nu)(\mathrm{d}z) \ge \int_{\|z\| \ge \frac{1}{2}R} \|z\| \cdot \|z\|^{p-1} (\mu + \nu)(\mathrm{d}z) \\
&\ge \left(\frac{1}{2}R\right)^{p-1} \int_{\|z\| \ge \frac{1}{2}R} \|z\| (\mu + \nu)(\mathrm{d}z).
\end{aligned}$$

Hence we can continue (7):

$$\begin{aligned}
& |Q(x,\mu) - Q(x,\nu)| \\
& \le 4L_o r + 2\beta_1 R \cdot N \cdot \alpha(\mu,\nu) + 2\beta_1 \cdot 2K \cdot 2^{(p-1)} \cdot R^{-(p-1)}.
\end{aligned}$$

The above estimate holds for all $x \in D$ and all $\mu, \nu \in \Delta_{p,K}(\mathbb{R}^s)$ such that $\alpha(\mu,\nu) \le \delta := \min\{\delta_o, \delta_1, \delta_2\}$. Inserting $N \le \kappa(\frac{R}{r})^s$, $R = \alpha(\mu,\nu)^{-1/p(s+1)}$, and $r = \alpha(\mu,\nu)^{\frac{p-1}{p(s+1)}}$ yields the assertion. □

COROLLARY 3.2. *Suppose* (A1) *and* (A2) *and that there exists a bounded set* $S \subset \mathbb{R}^s$ *such that both of the supports of* $\mu, \nu \in \mathcal{P}(\mathbb{R}^s)$ *are contained in* S. *Let*

$D \subseteq \mathbb{R}^m$ be nonempty and bounded. Then there exist constants $L > 0$ and $\delta > 0$ such that

$$\sup_{x \in D} |Q(x,\mu) - Q(x,\nu)| \leq L \cdot \alpha(\mu,\nu)^{\frac{1}{s+1}}$$

whenever $\alpha(\mu,\nu) < \delta$.

Proof. In the same notation as in the proof of Proposition 3.1 we define the radius $r := \alpha(\mu,\nu)^{1/s+1}$ and put $R > 0$ as a constant such that $S \subset Ax + B_o$ for any $x \in D$. Repeating the proof of Proposition 3.1 until (6) and inserting the expressions for r and $N \leq \kappa \cdot R^s \cdot \alpha(\mu,\nu)^{-s/s+1}$ yield the desired estimate. □

A class $\mathcal{B}$ of Borel sets in $\mathbb{R}^s$ is called a μ-uniformity class if $\sup\{|\mu(B) - \mu_n(B)| : B \in \mathcal{B}\} \to 0$ holds for every sequence $\{\mu_n\}$ in $\mathcal{P}(\mathbb{R}^s)$ converging weakly to $\mu \in \mathcal{P}(\mathbb{R}^s)$ (cf. [4]). Theorem 2.11 in [4] says that the family $\mathcal{B}_c$ of all convex Borel sets in $\mathbb{R}^s$ is a μ-uniformity class if μ is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}^s$. Since $\mathcal{B}_\mathcal{K} \subset \mathcal{B}_c$, this implies that $\alpha(\mu_n,\mu) \to 0$ provided that $\{\mu_n\}$ converges weakly to μ and μ is absolutely continuous. In [33], Proposition 3.4, it is stated that Q as a function from $\mathbb{R}^m \times \Delta_{p,K}(\mathbb{R}^s)$ to $\mathbb{R}$ is continuous on $\mathbb{R}^m \times \{\mu\}$, provided that μ is absolutely continuous and $\Delta_{p,K}(\mathbb{R}^s)$ is equipped with weak convergence of probability measures. By the above remark this result is now obtained as a conclusion of Proposition 3.1 which, therefore, can be read as a quantification of the continuity result in [33].

For the situation of pure integer recourse, i.e., the case where Φ in (3) is given by

$$\Phi(b) = \min\{q^T y : Wy \geq b,\ y \in \mathbb{Z}_+^{\bar{s}}\},$$

some substantial improvements of Proposition 3.1 are possible.

First recall (cf. §2) that here $\mathcal{K} = -\mathbb{R}_+^s$, hence $\mathcal{B}_\mathcal{K}$ consists of all bounded polyhedra whose facets parallel those of $\bigtimes_{i=1}^s [0,1]$, i.e., of all boxes in $\mathbb{R}^s$. The discrepancy $\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu)$ can now be bounded above by the Kolmogorov–Smirnov distance

$$\|F_\mu - F_\nu\|_\infty := \sup\{|F_\mu(t) - F_\nu(t)| : t \in \mathbb{R}^s\}$$

of the distribution functions F_μ and F_ν.

LEMMA 3.3. *If $\mathcal{K} = -\mathbb{R}_+^s$ then $\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu) \leq 2^s \cdot \|F_\mu - F_\nu\|_\infty$.*

Proof. Let $B \in \mathcal{B}_\mathcal{K}$ and assume a representation $B = \bigtimes_{i=1}^s [\underline{b}_i, \bar{b}_i]$. Then B has exactly 2^s vertices b_j $(j = 1,\ldots,2^s)$. Let $\varepsilon > 0$ and consider $B_\varepsilon = \bigtimes_{i=1}^s (\underline{b}_i - \varepsilon, \bar{b}_i]$ with vertices $b_{j,\varepsilon}$ $(j = 1,\ldots,2^s)$. According to a well-known formula there exist $n_{j,\varepsilon} \in \{0,1\}$ $(j = 1,\ldots,2^s)$ such that $\mu(B_\varepsilon) = \sum_{j=1}^{2^s} (-1)^{n_{j,\varepsilon}} F_\mu(b_{j,\varepsilon})$.

Therefore

$$|\mu(B_\varepsilon) - \nu(B_\varepsilon)| \leq 2^s \cdot \|F_\mu - F_\nu\|_\infty$$

for all $\varepsilon > 0$ and all $B \in \mathcal{B}_\mathcal{K}$.

Let $\varepsilon \downarrow 0$; the continuity of μ and ν on monotone sequences of sets then implies

$$|\mu(B) - \nu(B)| \leq 2^s \cdot \|F_\mu - F_\nu\|_\infty \quad \text{for all} \quad B \in \mathcal{B}_\mathcal{K}. \qquad \square$$

Another peculiarity we can benefit from is that here Lemma 2.3 holds with $L_o = 0$; i.e., Φ is constant on each of the continuity sets $\mathcal{B}_i$.

PROPOSITION 3.4. *Let* (1) *have pure integer recourse. Suppose* (A1) *and* (A2) *and let $D \subset \mathbb{R}^m$ be nonempty and bounded. Then there exist constants $L > 0$ and $\delta > 0$ such that*

$$\sup_{x \in D} |Q(x,\mu) - Q(x,\nu)| \leq L \cdot \|F_\mu - F_\nu\|_\infty^{\frac{p-1}{p+s}}$$

whenever $\mu, \nu \in \Delta_{p,K}(\mathbb{R}^s)$, $\|F_\mu - F_\nu\|_\infty < \delta$.

Proof. Let $x \in D$ be arbitrary and $\mu, \nu \in \Delta_{p,K}(\mathbb{R}^s)$ such that $\|F_\mu - F_\nu\|_\infty < \delta_o := 1$. Define a radius $R := \|F_\mu - F_\nu\|_\infty^{-1/s+p}$. Let $B_o = \boldsymbol{B}(0, R)$. In view of Lemma 2.5 and Lemma 2.2(ii) there exists a constant $\kappa > 0$, independent on R, such that B_o is intersected by at most $\kappa \cdot R^s$ rectangular continuity regions of Φ. Now it holds in analogy to the proof of Proposition 3.1 that

$$|Q(x,\mu) - Q(x,\nu)| \leq 2\beta_1 R \cdot \kappa R^s \cdot 2^s \|F_\mu - F_\nu\|_\infty + 2\beta_1 \cdot 2K \cdot 2^{-(p-1)} \cdot R^{-(p-1)} \tag{8}$$

provided that $\mu, \nu \in \Delta_{p,K}(\mathbb{R}^s)$, $\|F_\mu - F_\nu\|_\infty \leq \delta$ with some properly chosen $\delta > 0$. Inserting $R := \|F_\mu - F_\nu\|_\infty^{-1/s+p}$ completes the proof. □

COROLLARY 3.5. *Let* (1) *have pure integer recourse. Suppose* (A1) *and* (A2) *and that there exists a bounded set* $S \subset \mathbb{R}^s$ *such that both of the supports of* $\mu, \nu \in \mathcal{P}(\mathbb{R}^s)$ *are contained in* S. *Let* $D \subset \mathbb{R}^m$ *be nonempty and bounded. Then there exist constants* $L > 0$ *and* $\delta > 0$ *such that*

$$\sup_{x \in D} |Q(x,\mu) - Q(x,\nu)| \leq L \cdot \|F_\mu - F_\nu\|_\infty$$

whenever $\|F_\mu - F_\nu\|_\infty < \delta$.

Proof. Adopt the notation from the above proof and put $R > 0$ as a constant such that $S \subset Ax + B_o$ for any $x \in D$. We obtain the same estimate as in (8) with the difference that the second member of the sum on the right does not appear. □

4. Rates of convergence for optimal values. In contrast to stochastic programs with noninteger recourse, where $Q(.,\mu)$ in (2) is always convex, integer recourse models obey local minimizers which are not necessarily global. The subsequent analysis is directed to convergence rates for optimal values if the underlying probability measure in (1) is subjected to perturbations. From the literature [16], [27] it is well known that already qualitative investigations on the convergence of optimal values necessitate an exclusion of pathological types of local minimizers in the unperturbed problem (cf. also [33] and [32], which contains an illustrative example). This leads to the concept of a complete local minimizing set (CLM set) (cf. [16], [27]) which we will introduce next.

Let $V \subset \mathbb{R}^m$ be arbitrary and let $\operatorname{cl} V$ denote the closure of V. Recall the shape of (1) and consider the following localized optimal value and set of optimal solutions:

$$\varphi_V(\mu) := \inf\{f(x) + Q(x,\mu) : x \in C \cap \operatorname{cl} V\},$$
$$\psi_V(\mu) := \{x \in C \cap \operatorname{cl} V : f(x) + Q(x,\mu) = \varphi_V(\mu)\}.$$

Given $\mu \in \mathcal{P}(\mathbb{R}^s)$, a nonempty set $M \subset \mathbb{R}^m$ is called a CLM set for (1) with respect to an open set $V \subset \mathbb{R}^m$ if $M \subset V$ and $M = \psi_V(\mu)$.

Obvious examples for CLM sets are the set of global minimizers and isolated local minimizers (cf. [16], [27] for further details).

THEOREM 4.1. *Suppose* (A1) *and* (A2), *let* $\mu \in \mathcal{P}(\mathbb{R}^s)$ *be absolutely continuous with respect to the Lebesgue measure on* $\mathbb{R}^s$, *and let there exist constants* $p > 1$ *and* $K > 0$ *such that* $\mu \in \Delta_{p,K}(\mathbb{R}^s)$. *Assume further that* $M \subset \mathbb{R}^m$ *is a CLM set for* (1) *with respect to some bounded open set* $V \subset \mathbb{R}^m$. *Then there exist constants* $L > 0$ *and* $\delta > 0$ *such that*

$$|\varphi_V(\mu) - \varphi_V(\nu)| \leq L \cdot \alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu)^{\frac{p-1}{p(s+1)}}$$

whenever $\nu \in \Delta_{p,K}(\mathbb{R}^s)$, $\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu) < \delta$.

Proof. Since μ is absolutely continuous, $\mathcal{B}_\mathcal{K}$ is a μ-uniformity class. In view of Proposition 4.1(ii), (iii) in [33] there then exists a $\delta > 0$ such that $\psi_V(\nu)$ is a CLM set for (1) satisfying $\emptyset \neq \psi_V(\nu) \subset V$ for all $\nu \in \Delta_{p,K}(\mathbb{R}^s)$, $\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu) < \delta$.

Let $\nu \in \Delta_{p,K}(\mathbb{R}^s)$, $\alpha_{\mathcal{B}_\mathcal{K}}(\mu,\nu) < \delta$, and $x_\nu \in \psi_V(\nu)$, $x_\mu \in \psi_V(\mu) = M$. Then it holds that

$$\varphi_V(\mu) \leq f(x_\nu) + Q(x_\nu,\mu) \leq \varphi_V(\nu) + |Q(x_\nu,\mu) - Q(x_\nu,\nu)|$$

and

$$\varphi_V(\nu) \leq f(x_\mu) + Q(x_\mu,\nu) \leq \varphi_V(\mu) + |Q(x_\mu,\nu) - Q(x_\mu,\mu)|.$$

Hence

$$|\varphi_V(\mu) - \varphi_V(\nu)| \leq \sup_{x \in C \cap \operatorname{cl} V} |Q(x,\mu) - Q(x,\nu)|. \tag{9}$$

Since $C \cap \operatorname{cl} V$ is a bounded subset, we can apply Proposition 3.1, which completes the proof. □

Let us point out that in the above proof we have also shown that the sets $\psi_V(\nu)$ are CLM sets and, therefore, sets of local minimizers to (1) (i.e., under minimization over C!). Nonemptiness of $\psi_V(\nu)$ alone is not sufficient for the latter, since the minimization is restricted to $C \cap \operatorname{cl} V$. Improved versions of Theorem 4.1 are straightforward when adopting the more specific setting as in Corollary 3.2, Proposition 3.4, and Corollary 3.5. The rates obtained there directly extend to the convergence of local optimal values.

5. Asymptotic convergence. This section is devoted to studying some implications of our previous results for the asymptotic convergence of (local) optimal values and optimal solutions when estimating the underlying measure μ in (1) by empirical measures. Given a sequence $\xi_1, \xi_2, \ldots, \xi_n, \ldots$ of independent $\mathbb{R}^s$-valued random variables on some probability space $(\Omega, \mathcal{A}, P)$ with joint distribution μ, the empirical measures $\mu_n(\omega)$ ($\omega \in \Omega$, $n \in \mathbb{N}$) are defined by

$$\mu_n(\omega) = \frac{1}{n} \sum_{i=1}^{n} \delta_{\xi_i(\omega)},$$

where $\delta_{\xi_i(\omega)}$ denotes the measure with unit mass at $\xi_i(\omega)$ (cf. [9],[11],[23],[35]). A classical result in probability theory (Glivenko–Cantelli theorem) states that

$$\|F_{\mu_n(\omega)} - F_\mu\|_\infty \underset{n\to\infty}{\longrightarrow} 0 \quad \text{for } P\text{-almost all } \omega \in \Omega.$$

Recalling that weak convergence of probability measures in $\mathcal{P}(\mathbb{R}^s)$ is equivalent to pointwise convergence of the distribution functions at continuity points of the limit function, the Glivenko–Cantelli theorem asserts some uniformity of weak convergence. In contrast to the uniformity of weak convergence expressed by μ-uniformity classes, we do not need that the limit measure μ is absolutely continuous here. Proposition 3.4 and Corollary 3.5 now allow immediate consequences for the asymptotic convergence of the objective function and local optimal values when estimating μ in (1) via empirical measures. Instead of elaborating this issue we take a more general stand and show how to extend the uniformity argument from lower left orthants (distribution functions) to the class $\mathcal{B}_\mathcal{K}$ introduced in §3. More specifically, we are interested in convergence properties of

$$\sup\{|\mu_n(\omega)(B) - \mu(B)| : B \in \mathcal{B}_\mathcal{K}\} \tag{10}$$

as n tends to infinity. This will allow us to prove asymptotic convergence of local optimal values for the general case of complete mixed-integer recourse. The proper tools from probability theory in this respect are Vapnik–Červonenkis classes of Borel sets in $\mathbb{R}^s$ [23], [37].

Given a family $\mathcal{B}_o \subset \mathcal{B}(\mathbb{R}^s)$, let $\mathcal{V}(\mathcal{B}_o)$ be the smallest $k \in \mathbb{N}$ such that for every finite set $E \subset \mathbb{R}^s$ with k elements not every subset of E is of the form $E \cap B$, $B \in \mathcal{B}_o$. $\mathcal{B}_o$ is called a *Vapnik–Červonenkis class (VČ class)* if $\mathcal{V}(\mathcal{B}_o) < +\infty$.

It is well known (e.g., [11], [23], [35]) that the family of all closed halfspaces in $\mathbb{R}^s$ forms a VČ class. Moreover, if $\mathcal{B}', \mathcal{B}'' \subseteq \mathcal{B}(\mathbb{R}^s)$ are VČ classes then this is also true for $\mathcal{B}' \cap \mathcal{B}''$ and $\mathcal{B}' \cup \mathcal{B}''$ where $\mathcal{B}' \uplus\!\!\!\cap\ \mathcal{B}'' := \{B' \uplus\!\!\!\cap\ B'' : B' \in \mathcal{B}',\ B'' \in \mathcal{B}''\}$ [23], [35]. Now, observe that for each polyhedron in $\mathcal{B}_\mathcal{K}$ the number of facets is bounded above by a constant K equal to $2s$ plus twice the number of facets of $\mathcal{K}$. Together with the facts just mentioned this implies that $\mathcal{B}_\mathcal{K}$ is a VČ class.

Next, we have to make sure that (10) defines a measurable function with respect to ω. A priori, this is not clear since the supremum is taken over a quite general uncountable set. We refer the reader to the discussion in Appendix C of [23] where the measurability problem is addressed from a comprehensive point of view. In the present paper, the structure of $\mathcal{B}_\mathcal{K}$ also allows us to proceed in a direct way. We will index $\mathcal{B}_\mathcal{K}$ using a finite dimensional vector space, and then well-known results on the measurability of optimal values in nonlinear programs apply. Indeed, with a suitable matrix $H \in L(\mathbb{R}^s, \mathbb{R}^K)$ whose rows correspond to the (at most) K different facets of elements of $\mathcal{B}_\mathcal{K}$ we obtain

$$\mathcal{B}_\mathcal{K} = \{B(\tau) : \tau \in \mathbb{R}^K\} \text{ with } B(\tau) = \{t \in \mathbb{R}^s : Ht \leq \tau\}.$$

Now

$$\begin{aligned}
&\sup\{|\mu_n(\omega)(B) - \mu(B)| : B \in \mathcal{B}_\mathcal{K}\} \\
&\quad = \sup\{|\mu_n(\omega)(B(\tau)) - \mu(B(\tau))| : \tau \in \mathbb{R}^K\} \\
&\quad = \sup\left\{\left|\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{B(\tau)}(\xi_i(\omega)) - \mu(B(\tau))\right| : \tau \in \mathbb{R}^K\right\},
\end{aligned}$$

where $\mathbb{1}_{B(\tau)}$ denotes the indicator function of $B(\tau)$.

As a multifunction, $B : \mathbb{R}^K \longrightarrow \mathbb{R}^s$ is upper semicontinuous; i.e., for any $\tau \in \mathbb{R}^K$ and any open set $V \supset B(\tau)$ there exists a neighbourhood $\mathcal{N}(\tau)$ such that $B(\tau') \subset V$ for all $\tau' \in \mathcal{N}(\tau)$ [2].

Therefore, the (single-valued) mappings $(\tau, \xi_i) \mapsto \mathbb{1}_{B(\tau)}(\xi_i)$ and $\tau \mapsto \mu(B(\tau))$ are upper semicontinuous and hence measurable. This yields the measurability of

$$\omega \mapsto \sup\{|\mu_n(\omega)(B) - \mu(B)| : B \in \mathcal{B}_\mathcal{K}\}$$

via the above identity and Lemma III.39 in [8] or Theorem 2K in [28].

Having settled both the VČ property of $\mathcal{B}_\mathcal{K}$ and the measurability in (10) we are ready to apply general results from the literature to our setting.

LEMMA 5.1 (see [11], [23], [37]). *It holds that*

$$\sup\{|\mu_n(\omega)(B) - \mu(B)| : B \in \mathcal{B}_\mathcal{K}\} \underset{n\to\infty}{\longrightarrow} 0 \quad \textit{for } P - \textit{almost all } \omega \in \Omega.$$

LEMMA 5.2 (see [18]). *It holds that*

$$\limsup_{n\to\infty} \left(\frac{n}{2\log\log n}\right)^{\frac{1}{2}} \cdot \sup\{|\mu_n(\omega)(B) - \mu(B)| : B \in \mathcal{B}_\mathcal{K}\} \leq \frac{1}{2}$$

for P-almost all $\omega \in \Omega$.

Lemma 5.1 was specified from Theorem 14 in [23, p. 18] and Lemma 5.2 from Corollary 2.4 in [18].

The mentioned sources being rather abstract we enclose some comments for the benefit of the reader. The measurability assumptions in both sources are to ensure measurability of the suprema in the assertions. For our situation this is settled above. The VČ property among the above assumptions says that, given measurability, convergence of the suprema is essentially ruled by combinatorial properties (consult [23] for a thorough discussion). Being a VČ class, roughly speaking, means for a family $\mathcal{B}_o \subseteq \mathcal{B}(\mathbb{R}^s)$ that, for some cardinality $k \in \mathbb{N}$ and for any finite set $E \subset \mathbb{R}^s$ with exactly k elements, $\mathcal{B}_o$ is unable to pick out all the possible 2^k subsets of E. Given $n \geq k$, it then can be shown that $\mathcal{B}_o$ picks out at most

$$\binom{n}{0} + \cdots + \binom{n}{k-1} \tag{11}$$

subsets from any finite set with exactly n elements (cf. [23, Thm. 16, p. 19]). The number of different partitions of E of the form $B \cap E, B^c \cap E, B \in \mathcal{B}_o$ (with the superscript denoting complementation) is thus polynomial in n. This polynomiality is crucial from a technical viewpoint and explains the term "class with polynomial discrimination" arising in Theorem 14 in [23, p. 18] .

Employing the above preliminaries we obtain the following results.

PROPOSITION 5.3. *Suppose* (A1) *and* (A2); *let* $\mu \in \Delta_{p,K}(\mathbb{R}^s)$ *for some* $p > 1$, $K > 0$; *and assume that* $M \subset \mathbb{R}^m$ *is a CLM set for* (1) *with respect to some bounded open set* $V \in \mathbb{R}^m$. *Then it holds that*

(i) $\varphi_V(\mu_n(\omega)) \underset{n\to\infty}{\longrightarrow} \varphi_V(\mu)$ *for P-almost all* $\omega \in \Omega$,

(ii) *for any open set* $\tilde{V} \subset \mathbb{R}^m$ *such that* $\psi_V(\mu) \subset \tilde{V}$ *and P-almost all* $\omega \in \Omega$ *there exists an* $n_o(\omega) \in \mathbb{N}$ *such that* $\psi_V(\mu_n(\omega)) \subset \tilde{V}$ *for all* $n \geq n_o(\omega)$ *(upper semicontinuity of* ψ_V*),*

(iii) *for P-almost all* $\omega \in \Omega$ *there exists an* $n_1(\omega) \in \mathbb{N}$ *such that for all* $n \geq n_1(\omega)$ *the set* $\psi_V(\mu_n(\omega))$ *is a CLM set for* (1) *with respect to* V.

Proof. To simplify the notation we often write μ_n instead of $\mu_n(\omega)$. Since $C \cap \mathrm{cl} V$ is compact and $Q(.,\mu_n)$, $Q(.,\mu)$ are both lower semicontinuous on $\mathbb{R}^s$ (Proposition 3.1 in [33]), the sets $\psi_V(\mu_n)$ and $\psi_V(\mu)$ are both nonempty. Let $x_n \in \psi_V(\mu_n)$ and $x \in \psi_V(\mu)$. As in (9) we obtain

$$|\varphi_V(\mu_n) - \varphi_V(\mu)| \leq \sup_{x \in C \cap \mathrm{cl}\, V} |Q(x,\mu_n) - Q(x,\mu)|. \tag{12}$$

It holds that (cf., e.g., [24, Chap. 4.1, Cor. 2])

$$\int_{\mathbb{R}^s} \|z\|^p \mu_n(\mathrm{d}z) \underset{n\to\infty}{\longrightarrow} \int_{\mathbb{R}^s} \|z\|^p \mu(\mathrm{d}z) \quad \text{for } P\text{-almost all } \omega \in \Omega. \tag{13}$$

Hence, for P-almost all $\omega \in \Omega$ there exists an $n_3(\omega) \in \mathbb{N}$ such that $\mu_n(\omega) \in \Delta_{p,2K}(\mathbb{R}^s)$ for all $n \geq n_3(\omega)$, and Proposition 3.1 works in the present setting (with $\Delta_{p,2K}(\mathbb{R}^s)$ instead of $\Delta_{p,K}(\mathbb{R}^s)$). $\mathcal{B}_\mathcal{K}$ being a VČ class, Lemma 5.1 and (12) together establish (i).

Since $C \cap \mathrm{cl}\, V$ is compact, upper semicontinuity of ψ_V is equivalent to closedness of ψ_V (cf. [2]); i.e., given $x_n \in \psi_V(\mu_n)$ such that $x_n \underset{n\to\infty}{\longrightarrow} \bar{x}$ we have to show that $\bar{x} \in \psi_V(\mu)$.

Let $\omega \in \Omega$ be such that (i), (13), and the assertion of Lemma 5.1 hold. Let $\varepsilon > 0$ be arbitrary and $n_4(\omega) \in \mathbb{N}$ such that for $n \geq n_4(\omega)$

$$f(\bar{x}) + Q(\bar{x}, \mu) \leq f(x_n) + Q(x_n, \mu) + \varepsilon/3, \tag{14}$$

$$\sup_{x \in C \cap \operatorname{cl} V} |Q(x, \mu_n) - Q(x, \mu)| \leq \varepsilon/3, \tag{15}$$

$$|\varphi_V(\mu_n) - \varphi_V(\mu)| \leq \varepsilon/3. \tag{16}$$

The validity of (14) follows from the lower semicontinuity of $Q(., \mu)$.

Now it holds for $n \geq n_4(\omega)$ that

$$\begin{aligned} f(\bar{x}) + Q(\bar{x}, \mu) &\leq f(x_n) + Q(x_n, \mu) + \varepsilon/3 \\ &= f(x_n) + Q(x_n, \mu_n) - Q(x_n, \mu_n) + Q(x_n, \mu) + \varepsilon/3 \\ &\leq \varphi_V(\mu_n) + 2\varepsilon/3 \ \leq \ \varphi_V(\mu) + \varepsilon. \end{aligned}$$

Since $\varepsilon > 0$ was arbitrary this implies $\bar{x} \in \psi_V(\mu)$, and (ii) is shown.

To establish (iii) recall that $\psi_V(\mu_n(\omega))$ is nonempty due to the lower semicontinuity of $Q(., \mu_n)$ and the compactness of $C \cap \operatorname{cl} V$. The remaining part of the CLM property is a direct consequence of (ii). □

Under the additional assumption that μ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^s$, a version of Proposition 5.3 working for arbitrary perturbations in the topology of weak convergence of probability measures has been proved in [33]. The crucial point of Proposition 5.3 is that in the context of estimation via empirical measures one can dispense with the smoothness assumption on μ.

PROPOSITION 5.4. *Adopt the setting of Proposition* 5.3. *Then there exists a constant* $c > 0$ *such that for* P*-almost all* $\omega \in \Omega$

$$\limsup_{n \to \infty} \left(\frac{n}{2 \log \log n} \right)^{\frac{1}{2} \cdot \frac{p-1}{p(s+1)}} |\varphi_V(\mu_n(\omega)) - \varphi_V(\mu)| \leq c.$$

Proof. By (13) we again have that $\mu_n(\omega) \in \Delta_{p,2K}(\mathbb{R}^s)$ for P-almost all $\omega \in \Omega$ and $n \in \mathbb{N}$ sufficiently large. Since $\mathcal{B}_K$ is a VČ class, Lemma 5.1 implies that for P-almost all $\omega \in \Omega$ and $n \in \mathbb{N}$ sufficiently large $\alpha_{\mathcal{B}_K}(\mu_n(\omega), \mu) < \delta$ where δ is taken according to Proposition 3.1 (with $\Delta_{p,2K}(\mathbb{R}^s)$ instead of $\Delta_{p,K}(\mathbb{R}^s)$). Hence, (12) and Proposition 3.1 yield

$$\left(\frac{n}{2 \log \log n} \right)^{\frac{1}{2} \cdot \frac{p-1}{p(s+1)}} |\varphi_V(\mu_n(\omega)) - \varphi_V(\mu)| \leq L \cdot \left[\left(\frac{n}{2 \log \log n} \right)^{1/2} \alpha(\mu_n(\omega), \mu) \right]^{\frac{p-1}{p(s+1)}}$$

for P-almost all $\omega \in \Omega$ and $n \in \mathbb{N}$ sufficiently large. Taking the lim sup and employing Lemma 5.2 yields the assertion with $c := L \cdot (\frac{1}{2})^{\frac{p-1}{p(s+1)}}$. □

If $\mathcal{B}_o \subset \mathcal{B}(\mathbb{R}^s)$ is a VČ class, then there exists a function $\pi : \mathbb{R}_+ \to \mathbb{R}_+$ growing at most polynomially such that for any $\varepsilon > 0$ and all sufficiently large $n \in \mathbb{N}$

$$P(\{\omega \in \Omega : \alpha_{\mathcal{B}_o}(\mu_n(\omega), \mu) \geq \varepsilon\}) \leq 4\pi(2n) \exp(-n\varepsilon^2/8)$$

(cf. [35, p. 829]). Combining this result with Theorem 4.1 yields an upper estimate for $P(\{\omega \in \Omega : |\varphi_V(\mu_n(\omega)) - \varphi_V(\mu)| \geq \varepsilon\})$. Similar estimates for noninteger recourse models are addressed in [14], [15], [30], [38], for instance.

6. Conclusions. We have elaborated a variational distance of probability measures based on a suitable Vapnik–Červonenkis class of Borel sets in $\mathbb{R}^s$ that allows quantitative statements on the stability of stochastic programs with mixed-integer recourse. The distance metrizes weak convergence of probability measures under mild additional assumptions. Objective-function values for different underlying probability measures are related to each other by a Hölder estimate in terms of the mentioned variational distance (Proposition 3.1). The Hölder exponent improves if the supports of the probability measures are uniformly bounded or if the stochastic program has pure integer recourse (Corollaries 3.2 and 3.5, Proposition 3.4). Under mild assumptions, convergence rates for objective functions extend to local optimal values (Theorem 4.1). In the context of estimation via empirical measures the Vapnik–Červonenkis property is the key to concatenate our convergence results for local optimal values with qualitative and quantitative results on the asymptotic convergence of empirical measures (Propositions 5.3 and 5.4).

Acknowledgments. I wish to thank S. T. Rachev (University of California, Santa Barbara), W. Römisch (Humboldt-Universität Berlin), and W. Stute (Justus-Liebig-Universität Gießen) for beneficial discussions. Moreover, suggestions by two anonymous referees are gratefully acknowledged.

REFERENCES

[1] Z. Artstein and R. J-B. Wets, *Stability results for stochastic programs and sensors, allowing for discontinuous objective functions*, SIAM J. Optim., 4 (1994), pp. 537–550.
[2] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Nonlinear Parametric Optimization*, Akademie-Verlag, Berlin, 1982.
[3] B. Bank and R. Mandel, *Parametric Integer Optimization*, Akademie-Verlag, Berlin, 1988.
[4] R. N. Bhattacharya and R. Ranga Rao, *Normal Approximation and Asymptotic Expansions*, John Wiley, New York, 1976.
[5] P. Billingsley, *Convergence of Probability Measures*, John Wiley, New York, 1968.
[6] J. R. Birge and R. J-B. Wets, *Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse*, Math. Programming Stud., 27 (1986), pp. 54–102.
[7] C. E. Blair and R. G. Jeroslow, *The value function of a mixed integer program:* I, Discrete Math., 19 (1977), pp. 121–138.
[8] C. Castaing and M. Valadier, *Convex Analysis and Measurable Multifunctions*, in Lecture Notes in Mathematics, 580, Springer-Verlag, Berlin, 1977.
[9] R. M. Dudley, *Real Analysis and Probability*, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1989.
[10] J. Dupačová, *Stochastic programming with incomplete information: A survey of results on postoptimization and sensitivity analysis*, Optimization, 18 (1987), pp. 507–532.
[11] P. Gaenssler and W. Stute, *Empirical processes: A survey of results for independent and identically distributed random variables*, Ann. Probab., 7 (1979), pp. 193–243.
[12] P. Kall, *Stochastic Linear Programming*, Springer-Verlag, Berlin, 1976.
[13] P. Kall, A. Ruszczyński, and K. Frauendorfer, *Approximation techniques in stochastic programming*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J-B. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 33–64.
[14] Y. M. Kaniovski, A. J. King, and R. J-B. Wets, *Probabilistic bounds (via large deviations) for the solutions of stochastic programming problems*, Ann. Oper. Res., 56 (1995), pp. 189–208.
[15] V. Kaňková, *An approximative solution of a stochastic optimization problem*, in Trans. Eighth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, Academia, Prague, 1978, pp. 349–353.
[16] D. Klatte, *A note on quantitative stability results in nonlinear optimization*, in Proc. 19. Jahrestagung Mathematische Optimierung, K. Lommatzsch, ed., Seminarbericht Nr. 90, Humboldt-Universität Berlin, Sektion Mathematik, 1987, pp. 77–86.
[17] W. K. Klein Haneveld and M. H. van der Vlerk, *On the expected value function of a simple integer recourse problem with random technology matrix*, J. Comput. Appl. Math.,

56 (1994), pp. 45–54.
[18] J. KUELBS AND R. M. DUDLEY, *Log log laws for empirical measures*, Ann. Probab., 8 (1980), pp. 405–418.
[19] H. LANGEN, *Convergence of dynamic programming models*, Math. Oper. Res., 6 (1981), pp. 493–512.
[20] G. LAPORTE AND F. V. LOUVEAUX, *The integer L-shaped method for stochastic integer programs with complete recourse*, Oper. Res. Lett., 13 (1993), pp. 133–142.
[21] F. V. LOUVEAUX AND M. H. VAN DER VLERK, *Stochastic programs with simple integer recourse*, Math. Programming, 61 (1993), pp. 301–326.
[22] G. L. NEMHAUSER AND L. A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.
[23] D. POLLARD, *Convergence of Stochastic Processes*, Springer-Verlag, New York, 1984.
[24] S. T. RACHEV, *The Monge-Kantorovich mass transference problem and its stochastic applications*, Theory Probab. Appl., 29 (1984), pp. 647–676.
[25] ———, *Probability Metrics and the Stability of Stochastic Models*, John Wiley, New York, 1991.
[26] A. H. G. RINNOOY KAN AND L. STOUGIE, *Stochastic integer programming*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J-B. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 201–213.
[27] S. M. ROBINSON, *Local epi-continuity and local optimization*, Math. Programming, 37 (1987), pp. 208–222.
[28] R. T. ROCKAFELLAR, *Integral functionals, normal integrands and measurable selections*, in Nonlinear Operators and the Calculus of Variations, Lecture Notes in Mathematics, 543, G. P. Gossez et al., eds., Springer-Verlag, Berlin, 1976, pp. 157–207.
[29] W. RÖMISCH AND R. SCHULTZ, *Stability analysis for stochastic programs*, Ann. Oper. Res., 30 (1991), pp. 241–266.
[30] ———, *Lipschitz stability of stochastic programs with complete recourse*, SIAM J. Optim., 6 (1996), pp. 531–547.
[31] R. SCHULTZ, *Continuity and stability in two-stage stochastic integer programming*, in Stochastic Optimization, Numerical Methods and Technical Applications, Lecture Notes in Economics and Mathematical Systems Series 379, K. Marti, ed., Springer-Verlag, Berlin, 1992, pp. 81–92.
[32] ———, *Continuity properties of expectation functions in stochastic integer programming*, Math. Oper. Res. 18 (1993), pp. 578–589.
[33] ———, *On structure and stability in stochastic programs with random technology matrix and complete integer recourse*, Math. Programming, 70 (1995), pp. 73–89.
[34] S. SEN, R. D. DOVERSPIKE, AND S. COSARES, *Network planning with random demand*, Telecommunication Systems 3 (1994), pp. 11–30.
[35] G. R. SHORACK AND J. A. WELLNER, *Empirical Processes with Applications to Statistics*, John Wiley, New York, 1986.
[36] L. STOUGIE, *Design and analysis of algorithms for stochastic integer programming*, CWI Tract 37, Center for Mathematics and Computer Science, Amsterdam, 1987.
[37] V. N. VAPNIK AND A. V. ČERVONENKIS, *On the uniform convergence of relative frequencies of events to their probabilities*, Theory Probab. Appl., 16 (1971), pp. 264–280.
[38] S. VOGEL, *Stability results for stochastic programming problems*, Optimization, 19 (1988), pp. 269–288.
[39] D. W. WALKUP AND R. J-B. WETS, *Lifting projections of convex polyhedra*, Pac. J. Math. 28 (1969), pp. 465–475.
[40] R. J.-B. WETS, *Stochastic programs with fixed recourse: The equivalent deterministic program*, SIAM Rev., 16 (1974), pp. 309–339.
[41] R. M. WOLLMER, *Two-stage linear programming under uncertainty with* 0-1 *integer first stage variables*, Math. Programming, 19 (1980), pp. 279–288.

SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1153–1162, November 1996
© 1996 Society for Industrial and Applied Mathematics
015

OPTIMALITY OF NESTED PARTITIONS AND ITS APPLICATION TO CLUSTER ANALYSIS*

E. BOROS† AND F. K. HWANG‡

Abstract. A partition of a set N of n distinct numbers is called nested if four numbers $a < b < c < d$ in N such that a and c are in one part while b and d in another do not exist. A partition is called a p-partition if the number of parts is specified at p and a shape-partition if the sizes of the p parts are also specified. There are exponentially many p-partitions but only polynomially many nested p-partitions. In this paper we consider these notions in d-dimensional Euclidean spaces and give a general condition on the cost structure for which an optimal shape-partition is always nested. We illustrate applications of our results to some clustering problems, generalize some known results in this way, and propose some open problems.

Key words. clustering, nested partitions

AMS subject classifications. 62H30, 05A18

1. Introduction.

Consider the problem of partitioning a set N of n distinct numbers into nonempty disjoint parts. The partition is called an *open-partition* if the number of parts is not prespecified and called a *p-partition* if the number is specified to be p. If, furthermore, a set $\{n_1, \ldots, n_p\}$ with $\sum_{i=1}^{p} n_i = n$ is prespecified to be the set of sizes of the p parts, then the partition is called a *shape-partition*, shape referring to the set $\{n_i\}$.

Often, one encounters the problem of finding an optimal partition given a cost (of partition) function. However, the brute force approach of comparing the costs of all partitions is too time-consuming due to the large number of partitions. For example, using the principle of inclusion–exclusion, the number of p-partitions can be shown to be

$$\#(n,p) = \frac{1}{p!}\sum_{k=0}^{p-1}(-1)^k \binom{p}{k}(p-k)^n. \tag{1}$$

The number of open-partitions

$$\#(n) = \sum_{p=1}^{n} \#(n,p) \tag{2}$$

is represented by the *Bell numbers* whose first 10 terms are 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975. Even for the shape-partition, the number is

$$\#(n_1, \ldots, n_p) = \frac{n!}{\prod_{i=1}^{p} n_i! \prod_{j=1}^{n-p+1} p_j!}, \tag{3}$$

where p_j is the number of parts of size j. This number is easily seen to be exponential in n even for $p = 2$.

* Received by the editors August 4, 1994; accepted for publication (in revised form) April 20, 1995. This paper is a revised version of RUTCOR Research Report #7, 1993.

† RUTCOR, Rutgers University, New Brunswick, NJ 08904 (boros@rutcor.rutgers.edu). The research of this author was supported in part by Office of Naval Research grants N00014-92-J-1375 and N00014-92-J-4083 and by Air Force Office of Scientific Research grant F49620-95-1-0233.

‡ Department of Applied Mathematics, Chiao-Tung University, Hsin-Chu, Taiwan 30050 ROC (fhwang@math.nctu.edu.tw).

One way to deal with the combinatorial problem of huge partition spaces is to look for small subspaces which, nevertheless, also contain optimal partitions. One well-studied subspace consists of *consecutive partitions* [3,9] which are characterized by the requirement that each part of a consecutive partition consists of numbers consecutive in N. In this subspace each p-partition corresponds to a way of inserting $p-1$ bars into the $n-1$ spaces between the n numbers. The number of p-partitions is thus

$$\#_C(n,p) = \binom{n-1}{p-1}, \tag{4}$$

a polynomial function of n for fixed p. For shape-partitions, the number is easily seen to be

$$\#_C(n_1,\ldots,n_p) = \frac{p!}{\prod_{j=1}^{n-p+1} p_j!}. \tag{5}$$

When the "consecutive" subspace is not known to contain an optimal partition, one has to search other subspaces. Boros and Hammer [2] raised the notion of *nested partitions*, which is defined by the nonexistence of four numbers $a<b<c<d$ in N such that a and c belong to one part, while b and d belong to another. Note that a consecutive partition is always nested, but not vice versa. They demonstrated some cost functions which guarantee that an optimal p-partition is nested. Hwang and Mallows [10] showed that the number of nested p-partitions is

$$\#_N(n,p) = \frac{\binom{n}{p-1}\binom{n}{p}}{n}, \tag{6}$$

again, a polynomial function of n for fixed p.

The notions of "consecutiveness" and "nestedness" have been extended to vectors (points in d-dimensional spaces). We now extend them further to d-dimensional multisets. Let $X=\{x_1,x_2,\ldots,x_n\}\subset\mathbb{R}^d$ be a *multiset* of d-dimensional points, i.e., elements of X may coincide. Furthermore, let $\operatorname{conv}(X)$ denote the *convex hull* of X, and let $\operatorname{conv}^*(X)$ denote the relative interior of $\operatorname{conv}(X)$. A partition $\pi=(\pi_1,\ldots,\pi_p)$ of the multiset X (identical points are treated as separated entities in a partition) is called *consecutive* (see [1]) if $\operatorname{conv}^*(\pi_i)\cap\operatorname{conv}^*(\pi_j)=\emptyset$ for all $1\le i,j\le p$. It is called *nested* (see [2]) if for all $1\le i,j\le p$, either $\pi_i\cap\operatorname{conv}^*(\pi_j)=\emptyset$ or $\pi_j\cap\operatorname{conv}^*(\pi_i)=\emptyset$. Again, consecutiveness implies nestedness. In this paper we give a sufficient condition on the cost function such that an optimal shape-partition is always nested. A by-product is a sufficient condition for the existence of a consecutive optimal shape-partition. In particular, they lead to an extension of Fisher's result [6] on a clustering problem from one dimension to d dimensions which he long desired.

2. The main results. In this section we derive a general condition which guarantees that every optimal shape-partition is nested. We first consider a shape with only two parts, and then we extend the result to general p.

Consider a multiset X of d-dimensional points, $d\ge 1$, and a partition of it into two parts, $\pi=(\pi_1,\pi_2)$. Let, furthermore, $F(\pi)$ denote the cost of partition π.

Let π' be a partition obtained from π by interchanging two points, $x\in\pi_1$ and $y\in\pi_2$. Clearly, π' has the same shape as π. We will consider $F(\pi')-F(\pi)$ as a function $\Delta_F(x,y)$ of x and y, i.e., $\Delta_F:\pi_1\times\pi_2\longrightarrow\mathbb{R}$. More precisely, let us consider a continuous, real-valued mapping Δ_F^* over the space $\mathbb{R}^d\times\mathbb{R}^d$ satisfying the following

conditions:

$$\begin{array}{ll} \Delta_F^*(x,y) = \Delta_F(x,y) & \text{for all } x \in \pi_1 \text{ and } y \in \pi_2, \\ \Delta_F^*(z,z) = 0 & \text{for all } z \in \mathbb{R}^d. \end{array} \tag{7}$$

Such a mapping Δ_F^* exists and can naturally be considered as a continuous extension of Δ_F, for if $x \in \pi_1$ and $y \in \pi_2$ happen to coincide (X is a multiset), then $\Delta_F(x,y) = 0$ since the switch of identical elements does not change the partition. Let us remark that in most cases, when F is given in an algebraic form, the formula for Δ_F will automatically define such an extension.

For a fixed vector $x \in \pi_1$ let us introduce the notation $g_x^*(y) = \Delta_F^*(x,y)$ and, analogously, let $g_y^*(x) = \Delta_F^*(x,y)$, if we want to emphasize that $y \in \pi_2$ is fixed now. Let, furthermore, $X^+(g_y^*) = \{x \in \mathbb{R}^d | g_y^*(x) \geq 0\}$, and let $Y^+(g_x^*) = \{y \in \mathbb{R}^d | g_x^*(y) \geq 0\}$.

We are ready now to state a sufficient condition for a shape-partition to be nested.

THEOREM 2.1. *For a shape-partition problem let $\pi = (\pi_1, \pi_2)$ be an optimal partition. Further let us suppose that either for every $x \in \pi_1$, the set $Y^+(g_x^*)$ is a convex set with x being a boundary point, or for every $y \in \pi_2$, the set $X^+(g_y^*)$ is a convex set with y being a boundary point. Then $\pi = (\pi_1, \pi_2)$ is nested.*

Proof. Let us assume that for every $y \in \pi_2$, the set $X^+(g_y^*)$ is a convex set with y being a boundary point. Since π is optimal, i.e., its cost $F(\pi)$ is minimal among all partitions of the same shape, $g_y^*(x) \geq 0$ for all $x \in \pi_1$ and all $y \in \pi_2$, implying

$$\pi_1 \subset X^+(g_y^*) \tag{8}$$

for all $y \in \pi_2$. Since for each $y \in \pi_2$ the set $X^+(g_y^*)$ is convex with y being a boundary point, the intersection of all these sets

$$X^+ = \bigcap_{y \in \pi_2} X^+(g_y^*)$$

is also convex and no point of π_2 belongs to its interior. Since $\pi_1 \subset X^+$ by (8), $\pi_2 \cap \text{conv}^*(\pi_1) = \emptyset$ follows, which proves that π is nested. The other case is analogous. □

Sometimes, it is easier to use Theorem 2.1 when the conditions are specified on the functions g_y^* and g_x^*.

A real-valued function $f(x)$ is called *quasi concave* if over any interval $[a,b] = \{\alpha a + (1-\alpha)b | 0 \leq \alpha \leq 1\}$ it always attains its minimum over $[a,b]$ at one of the endpoints. The function f is called *strictly quasi concave* if no internal point of an interval can be a minimum (over that interval). It is well known that a (strictly) concave function is (strictly) quasi concave.

COROLLARY 2.2. *Let X be a given multiset with F being the cost function of its partitions, as before. If either g_x^* for any $x \in X$ or g_y^* for any $y \in X$ is strictly quasi concave, then every optimal shape-partition is nested.*

Proof. Let us assume that g_y^* for any $y \in X$ is strictly quasi concave. The other case can be treated analogously.

Let us consider an optimal shape-partition $\pi = (\pi_1, \pi_2)$. According to the previous theorem, if $X^+(g_y^*)$ is convex having y on its boundary for every $y \in \pi_2$, then π is necessarily nested.

Let us observe first that for every $y \in \pi_2$, the point y must be a boundary point of $X^+(g_y^*)$, since g_y^* is strictly quasi concave. This implies that if π is not nested,

then, by Theorem 2.1, there is a vector $y \in \pi_2$ for which the set $X^+(g_y^*)$ is not convex. Then there must exist points $u, v \in X^+(g_y^*)$ and $w = \alpha u + (1-\alpha)v \notin X^+(g_y^*)$ for some $0 < \alpha < 1$, i.e., for which $g_y^*(u) \geq 0$, $g_y^*(v) \geq 0$ while $g_y^*(w) < 0$. Since g_y^* is continuous, the interval $[u, v]$ has an internal minimum, contradicting the strict quasi concavity of g_y^*. □

THEOREM 2.3. *Suppose that the cost function has the structure* $F(\pi) = \sum_{i=1}^{p} f(\pi_i)$, *i.e.,* $F(\pi)$ *is the sum of independent values associated with each of the parts. In this case, if every optimal shape-partition is nested holds for* $p = 2$, *then it holds for* $p > 2$.

Proof. Let π be an optimal shape-partition. By Theorem 2.1, any two parts of π must be a nested partition of their elements or we would be able to reduce $F(\pi)$ by making them nested, which contradicts the assumption that π is optimal. By the definition of a nested partition, π is nested if any two parts of π are pairwise nested. □

Interestingly, the arguments used to establish nested optimal partitions are also applicable for consecutive optimal partitions, for which more efficient algorithms exist.

THEOREM 2.4. *Consider a shape-partition problem and let* π *be an optimal partition. Suppose that for every pair* (π_i, π_j) *and for every* $x \in \pi_j$, $Y^+(g_x^*)$ *is a convex set with* x *on its boundary, and for every* $y \in \pi_j$, $X^+(g_y^*)$ *is a convex set having* y *as a boundary point. Then every optimal partition is consecutive.*

Proof. First consider the case of two parts. Let π be an optimal partition. We have argued in the proof of Theorem 2.1 that $X^+(g_y^*)$ being a convex set having y on its boundary implies that no $y \in \pi_2$ is in $\text{conv}^*(\pi_1)$. Similarly, $Y^+(g_x^*)$ being a convex set with x being the boundary implies that no $x \in \pi_1$ is in $\text{conv}^*(\pi_2)$. Hence π is consecutive. The result is then extended to general p parts by an argument analogous to the proof of Theorem 2.3. □

COROLLARY 2.5. *Suppose that both* g_x^* *and* g_y^* *are strictly quasi concave for every* x *and* y, *respectively. Then every optimal shape-partition is consecutive.*

Since an open-partition must be a p-partition for some p, and a p-partition must be a shape-partition for some shape, results in this section also apply to p-partitions and open-partitions.

3. Applications to clustering. In a clustering problem, one partitions a given set of points into clusters usually with points in the same cluster close to each other, though closeness can be defined in various ways. It is very rare for a clustering problem to have a polynomial-time algorithm for exact optimal clustering, due to the usually large number of possible clusterings. One of the few exceptions is due to Fisher who was one of the first to use consecutive partitions. Fisher [6] considered a one-dimensional clustering problem where the goal is to minimize the sum of squares, i.e., the cost of a partition $\pi = (\pi_1, \ldots, \pi_p)$ is

$$F(\pi) = \sum_{i=1}^{p} \sum_{x_j \in \pi_i} (x_j - \overline{x}_i)^2, \tag{9}$$

where $\overline{x}_i$ is the average of the numbers in π_i. He proved that there exists a consecutive optimal p-partition, even when there is a weight w_j associated to each number x_j. Since every open-partition must be a p-partition for some p, this also implies the existence of a consecutive optimal open-partition. Fisher wrote [6, pp. 796–797]: "It would of course be most desirable to develop, both theoretically and computationally, a distance criterion that is defined in more than one dimension. An example of the need for such a formulation is shown in a multivariate stratification problem

encountered in a sample survey by Hagood and Bernert [8]. Of course involved in any such approach is a relevant system of weighing the different dimensions to reflect their relative importance in determining distance." Gower [7] studied three criteria commonly adopted in the literature of cluster analysis for multivariate data. One of which, attributed to Edwards and Cavalli-Sforza [5], is to divide the data into two disjoint subsets with a minimum sum of squares, a special case of Fisher's d-dimensional problem with $p = 2$.

Unfortunately, Fisher's proof technique of the one-dimensional case cannot handle a weight function associated with the dimensions. Gower proved the existence of consecutive optimal partitions for $p = 2$ and without dimension weight. We now consider the general case. Suppose that $x_j = (x_{j1}, \ldots, x_{jd})$ and u_k is the positive weight of dimension k, $k = 1, \ldots, d$. Consider the cost function

$$
\begin{aligned}
F(\pi) &= \sum_{i=1}^{p} \sum_{x_j \in \pi_i} \sum_{k=1}^{d} u_k (x_{jk} - \overline{x}_{ik})^2 \\
&= \sum_{i=1}^{p} \sum_{x_j \in \pi_i} (y_j - \overline{y}_i)^2,
\end{aligned} \tag{10}
$$

where $y_j = (\sqrt{u_1} x_{j1}, \sqrt{u_2} x_{j2}, \ldots, \sqrt{u_d} x_{jd})$ and $\overline{y}_i = \sum_{x_j \in \pi_i} y_j / |\pi_i|$ is the mean (centroid) of the vectors y_j for $x_j \in \pi_i$ (and where the product of the vectors denotes their inner product).

In the following theorem we shall replace y_j in (10) by x_j for uniformity and also generalize it by introducing a weight function w_i associated to part i.

THEOREM 3.1. *Suppose that*

$$
F(\pi) = \sum_{i=1}^{p} w_i \sum_{x_j \in \pi_i} (x_j - \overline{x}_i)^2, \tag{11}
$$

where $w_i > 0$ and $\overline{x}_i$ is the centroid (mean, in this case) of the d-dimensional points in π_i. Then an optimal shape-partition must be nested.

Proof. By Theorem 2.3 it is enough to prove the above statement for the case of $p = 2$.

Let π be an optimal shape-partition and let π' be the partition obtained from π by interchanging $y \in \pi_1$ and $z \in \pi_2$. Let $\overline{x}'_1$ and $\overline{x}'_2$ denote the centroids of π'_1 and π'_2, respectively. Then

$$
\begin{aligned}
0 \le \Delta_F(y, z) &= w_1 \left[\sum_{x_j \in \pi'_1} (x_j - \overline{x}'_1)^2 - \sum_{x_j \in \pi_1} (x_j - \overline{x}_1)^2 \right] \\
&\quad + w_2 \left[\sum_{x_j \in \pi'_2} (x_j - \overline{x}'_2)^2 - \sum_{x_j \in \pi_2} (x_j - \overline{x}_2)^2 \right] \\
&= w_1 \left[\sum_{x_j \in \pi'_1} x_j^2 - \frac{\left(\sum_{x_j \in \pi'_1} x_j \right)^2}{n_1} - \sum_{x_j \in \pi_1} x_j^2 + \frac{\left(\sum_{x_j \in \pi_1} x_j \right)^2}{n_1} \right]
\end{aligned}
$$

$$
\begin{aligned}
&+ w_2 \left[\sum_{x_j \in \pi_2'} x_j^2 - \frac{\left(\sum_{x_j \in \pi_2'} x_j\right)^2}{n_2} - \sum_{x_j \in \pi_2} x_j^2 - \frac{\left(\sum_{x_j \in \pi_2} x_j\right)^2}{n_2} \right] \\
&= w_1 \left[z^2 - y^2 - \frac{(z-y)^2 + 2(z-y)\sum_{x_j \in \pi_1} x_j}{n_1} \right] \\
&\quad + w_2 \left[y^2 - z^2 - \frac{(y-z)^2 + 2(y-z)\sum_{x_j \in \pi_2} x_j}{n_2} \right] \\
&= w_1 \left[z^2 - y^2 - \frac{(z-y)^2}{n_1} - 2(z-y)\overline{x}_1 \right] + w_2 \left[y^2 - z^2 - \frac{(y-z)^2}{n_2} - 2(y-z)\overline{x}_2 \right] \\
&= \left(w_1 - w_2 - \frac{w_1}{n_1} - \frac{w_2}{n_2} \right) z^2 + \left(w_2 - w_1 - \frac{w_1}{n_1} - \frac{w_2}{n_2} \right) y^2 \\
&\quad + 2 \left(\frac{w_1}{n_1} + \frac{w_2}{n_2} \right) yz + 2(w_2 \overline{x}_2 - w_1 \overline{x}_1) z + 2(w_1 \overline{x}_1 - w_2 \overline{x}_2) y.
\end{aligned}
$$

View the above expression as a function of real y and z (vectors) with the given coefficient ($\overline{x}_1$ and $\overline{x}_2$ are treated as fixed) and define $g_z^*(y)$ and $g_y^*(z)$ accordingly. Since the sum of the coefficients of the z^2 term and the y^2 term is negative, at least one of them is negative, say, the coefficient of the z^2 term. Since $g_z^*(y)$ is separable in the dimension of y, it is easily verified that the negative coefficient of the z^2 term implies that the Hessian is negative-definite. Hence g_z^* is strictly concave. Since the coefficients of y^2 and z^2 are independent of the particular selection of y and z, we can conclude that g_z^* is strictly concave for all $z \in \pi_2$. By Corollary 2.2 an optimal shape-partition thus must be nested. □

COROLLARY 3.2. *If*

$$
|w_1 - w_2| \leq \frac{w_1}{n_1} + \frac{w_2}{n_2}, \tag{12}
$$

then every optimal shape-partition is consecutive.

Proof. The proof of Corollary 3.2 follows immediately from Corollary 2.5. □

In particular, if $w_i = 1$ for all i, then the condition of Corollary 3.2 is satisfied. Thus we have extended Fisher's sum-of-squares result to d-dimensional points.

By setting $w_i = 0$ for $n_i = 1$ and $w_i = 1/(n_i - 1)$ for $n_i \geq 2$, $F(\pi)$ in Theorem 3.1 represents the sum of variances (for multidimensional points, each variance is a weighted sum over the d dimensions). Therefore all shape-partitions to minimize the sum of variances are nested. It is also easily verified that (12) holds if $|n_1 - n_2| \leq 1$. Hence we have the following result.

COROLLARY 3.3. *Consider a partitioning problem where the part-sizes can differ by at most* 1. *Then every optimal partition minimizing the sum of variances is consecutive.*

One may feel that perhaps for an arbitrary shape there exists a consecutive optimal partition. We now give a one-dimensional example to show that Corollary 3.3 is tight, i.e., if the part-size can differ by 2, then no optimal partition is consecutive.

Let $N = \{0, 13, 14, 14, 15, 28\}$ and the shape be $\{2, 4\}$. Then $\pi_1 = \{14, 14\}$, $\pi_2 = \{0, 13, 15, 28\}$ is the optimal shape-partition minimizing the sum of variances. But $\{\pi_1, \pi_2\}$ is not a consecutive partition.

Another consequence of Corollary 2.2 is a strengthening of Theorem 1.2 of [2].

THEOREM 3.4. *Suppose that*

$$F(\pi) = \sum_{i=1}^{p} w_i \sum_{x_j, x_k \in \pi_i} (x_j - x_k)^2, \tag{13}$$

where $w_i > 0$. *Then every optimal shape-partition is nested.*

Proof. By Theorem 2 again, it is enough to consider $p = 2$. Let $y \in \pi_1$ and $z \in \pi_2$. Then

$$\begin{aligned}\Delta_F(y,z) &= w_1 \sum_{x_j \in \pi_1} \left((z - x_j)^2 - (y - x_j)^2\right) \\ &\quad + w_2 \sum_{x_j \in \pi_2} \left((y - x_j)^2 - (z - x_j)^2\right) \\ &= y^2[w_2(n_2 - 1) - w_1(n_1 + 1)] + z^2[w_1(n_1 - 1) - w_2(n_2 + 1)] \\ &\quad + 2(w_1 + w_2)yz + 2y\left[w_1 \sum_{x_j \in \pi_1} x_j - w_2 \sum_{x_j \in \pi_2} x_j\right] \\ &\quad + 2z\left[w_2 \sum_{x_j \in \pi_2} x_j - w_1 \sum_{x_j \in \pi_1} x_j\right],\end{aligned} \tag{14}$$

where n_1 and n_2 denote the cardinalities of π_1 and π_2, respectively. Since the sum of the coefficients of y^2 and z^2 is $-2(w_1 + w_2) < 0$, at least one of them is negative, implying that at least one of g_z^* or g_y^* is strictly concave. Thus, by Corollary 2.2, we can conclude that an optimal shape-partition must be nested. □

If both y^2 and z^2 have nonpositive coefficients in the above proof, i.e., if

$$|w_1 n_1 - w_2 n_2| \le w_1 + w_2, \tag{15}$$

then by Corollary 2.5 an optimal shape-partition must be consecutive. This observation yields the following interesting consequence.

COROLLARY 3.5. *Consider a shape-partitioning problem where*

$$F(\pi) = \sum_{i=1}^{p} \sum_{x_j, x_k \in \pi_i} (x_j - x_k)^2 \tag{16}$$

and in which the part-sizes can differ by at most 2. *Then every optimal shape-partition is consecutive.*

Boros and Hammer studied a one-dimensional clustering problem with

$$F(\pi) = \sum_{i=1}^{p} \sum_{x_j, x_k \in \pi_i} |x_j - x_k| \tag{17}$$

and proved that every optimal p-partition is nested. This, however, may not be true for shape-partitions. In this paper, instead of the absolute difference between two numbers, we consider the absolute difference between a number and the centroid of the part, with a part-weight.

THEOREM 3.6. *Suppose that*

$$F(\pi) = \sum_{i=1}^{p} w_i \sum_{x_j \in \pi_i} |x_j - m_i|, \tag{18}$$

$$z...z \;\; y...y \;\; |_{m_1} \;\; y...y \;\; z...z \;\; |_{m_2} \;\; y...y \; z...z$$

FIG. 1. *The ordering of y's and z's when* $w_1 \geq w_2$.

where $w_i > 0$ *and* m_i *is the median of the set* π_i. *Then every optimal shape-partition is nested.*

Proof. By Theorem 2.3 it is sufficient to consider $p = 2$. Let $y \in \pi_1$ and $z \in \pi_2$. Let $\pi' = (\pi'_1, \pi'_2)$ be obtained from π by interchanging y and z, and let m'_1 and m'_2 be the medians of π'_1 and π'_2. Without loss of generality, assume $m_1 \leq m_2$.

Case i. $y, z \geq m_2$. Then $m'_1 = m_1$ and $m'_2 = m_2$.

$$\begin{aligned} 0 \leq \Delta_F(y,z) &= w_1(z - m_1) + w_2(y - m_2) - w_1(y - m_1) - w_2(z - m_2) \\ &= (w_1 - w_2)(z - y). \end{aligned} \tag{19}$$

So $z - y$ has the same sign as $w_1 - w_2$.

Case ii. $m_1 \leq y, z \leq m_2$. Then $m'_1 = m_1$ and $m'_2 = m_2$.

$$\begin{aligned} 0 \leq \Delta_F(y,z) &= w_1(z - m_1) + w_2(m_2 - y) - w_1(y - m_1) - w_2(m_2 - z) \\ &= (w_1 + w_2)(z - y). \end{aligned} \tag{20}$$

So $z \geq y$.

Case iii. $y, z \leq m_1$. Then $m'_1 = m_1$ and $m'_2 = m_2$.

$$\begin{aligned} 0 \leq \Delta_F(y,z) &= w_1(m_1 - z) + w_2(m_2 - y) - w_1(m_1 - y) - w_2(m_2 - z) \\ &= (w_1 - w_2)(y - z). \end{aligned} \tag{21}$$

So $y - z$ has the same sign as $w_1 - w_2$.

First consider $w_1 \geq w_2$, then the ordering of $y \in \pi_1$ and $z \in \pi_2$ in the three intervals separated by m_1 and m_2 is shown in Figure 1.

We will show that a $y \in \pi_1$, $y \geq m_2$ and a $z \in \pi_2$, $m_1 \leq z < m_2$ cannot coexist. This will imply that the partition is nested. Suppose to the contrary that such a pair (y, z) exists. Then

$$\begin{aligned} 0 \leq \Delta_F(y,z) &= w_1 \sum_{x_j \in \pi'_1} |x_j - m'_1| + w_2 \sum_{x_j \in \pi'_2} |x_j - m'_2| \\ &\quad - w_1 \sum_{x_j \in \pi_1} |x_j - m_1| - w_2 \sum_{x_j \in \pi_2} |x_j - m_2| \\ &\leq w_1 \sum_{x_j \in \pi'_1} |x_j - m_1| + w_2 \sum_{x_j \in \pi'_2} |x_j - m_2| \\ &\quad - w_1 \sum_{x_j \in \pi_1} |x_j - m_1| - w_2 \sum_{x_j \in \pi_2} |x_j - m_2| \\ &= w_1(z - m_1) + w_2(y - m_2) - w_1(y - m_1) - w_2(m_2 - z) \\ &= w_1(z - y) + w_2(y + z - 2m_2) \\ &< w_2(2z - 2m_2) < 0, \end{aligned}$$

an absurdity.

Next consider $w_1 \leq w_2$; then the possible ordering of the y's and z's is as shown in Figure 2.

$$y...y \quad z...z \quad |_{m_1} \quad y...y \quad z...z \quad |_{m_2} \quad z...z \quad y...y$$

FIG. 2. *The ordering of y's and z's when* $w_1 \leq w_2$.

We can show in a similar fashion that $z \leq m_1$ and $m_1 < y \leq m_2$ cannot coexist. Thus the partition is nested. □

One can also observe that if $w_1 = w_2$, then the only order of the elements of π_1 and π_2 satisfying all conditions in the above proof is $y \mid y \quad z \mid z$, implying hence the following corollary (for a different proof, see also [9]).

COROLLARY 3.7. *If*

$$F(\pi) = \sum_{i=1}^{p} \sum_{x_j \in \pi_i} |x_j - m_i|, \tag{22}$$

where m_i *denotes the median of the set* π_i, *then every optimal shape-partition is consecutive.*

4. Some concluding remarks. Boros and Hammer proved that if $F(\pi) = \sum_{i=1}^{p} \sum_{x,y \in \pi_i} |x - y|$ for one-dimensional points or if $F(\pi) = \sum_{i=1}^{p} \sum_{x,y \in \pi_i} (x - y)^2$ for d-dimensional points, then an optimal partition is nested. We generalized their result by giving a broad sufficient condition derived from a novel geometric argument. We applied this condition to obtain the d-dimensional version of Fisher's clustering problem which he proposed but couldn't prove.

Since the concept of nested partition is fairly recent, there are still many unexplored issues. We raise the following questions.

(i) Clearly, a partition $\pi = (\pi_1, \ldots, \pi_p)$ is nested if and only if for any i and j, (π_i, π_j) is a nested partition of the elements in $\pi_i \cup \pi_j$. Does the existence of a nested optimal 2-partition guarantee the existence of a nested optimal p-partition for general p? An affirmative answer was recently given by Hwang, Rothblum, and Yao [11] for one-dimensional points but the problem for general dimension remains open.

(ii) For the subspace of consecutive partitions (in one dimension), there exist an $O(n^2)$-time dynamic programming algorithm to find an optimal open-partition [9] and an $O(pn^2)$-time algorithm to find an optimal p-partition. From (6), there exists an $O(n^{2p-2})$-time algorithm to find an optimal nested p-partition. Does there exist a better dynamic programming algorithm for the subspace of nested p-partitions? Note that since any subset can be a part in a nested open partition, for a general cost function, one must inspect at least 2^n cost terms to find an optimal nested open-partition.

For the one-dimensional case, it seems quite plausible to conjecture that there always exists a consecutive optimal partition. However, a counterexample was recently given by Chang and Hwang [4] in which the optimal partition is nested but is not consecutive.

REFERENCES

[1] E. R. Barnes, A. J. Hoffman, and U. G. Rothblum, *On optimal partitions having disjoint convex and conic hulls*, Math. Programming, 54 (1992), pp. 69–86.

[2] E. Boros and P. L. Hammer, *On clustering problems with connected optima in Euclidean spaces*, Discrete Math., 75 (1989), pp. 81–88.

[3] A. K. Chakravarty, J. B. Orlin, and U. G. Rothblum, *A partitioning problem with additive objective with an application to optimal inventory grouping for joint replenishment*, Oper. Res., 30 (1982), pp. 1018–1022.

[4] G. J. Chang and F. K. Hwang, *Optimality of consecutive and nested tree partitions*, to appear.

[5] A. W. F. Edwards and L. L. Cavalli-Sforza, *A method for cluster analysis*, Biometrics, 21 (1965), pp. 362–375.

[6] W. D. Fisher, *On grouping for maximum homogeneity*, J. Amer. Statist. Assoc., 53 (1958), pp. 789–798.

[7] J. C. Gower, *Some distance properties of latent roots and vector methods used in multivariate analysis*, Biometrika, 53 (1966), pp. 325–338.

[8] M. J. Hagood and E. H. Bernert, *Component indexes as a basis for stratification in sampling*, J. Amer. Statist. Assoc., 40 (1945), pp. 330–341.

[9] F. K. Hwang, *Optimal partitions*, J. Optim. Theory Appl., 34 (1981), pp. 1–10.

[10] F. K. Hwang and C. L. Mallows, *The numbers of nested partitions and inner-consecutive partitions*, J. Combin. Theory Ser. A, 70 (1995), pp. 323–333.

[11] F. K. Hwang, U. G. Rothblum, and Y. C. Yao, *Localizing combinatorial properties of partitions*, Discrete Math., to appear.