AIAA JOURNAL
Vol. 32, No. 6, June 1994

New Successive Approximation Method for Optimum Structural Design

J. A. Snyman* and Nielen Stander*
University of Pretoria, Pretoria 0002, South Africa

A new procedure for solving structural optimization problems, using only first-order explicit information and consisting of the successive solution of approximate subproblems, is proposed. The method differs from established sequential quadratic programming methods by not utilizing the Lagrangian form and not attempting to store full and explicit Hessian information. Instead each constraint is individually approximated by a quadratic function involving only one coefficient in the quadratic term. This coefficient is determined from a two-point collocation. A novel feature of the method is that each quadratic subproblem is solved by a recently proposed interior feasible direction method. An auxiliary problem which is easily integrated into the code is formulated to generate a feasible starting point for each subproblem. Four truss sizing problems were analyzed and results point to competitiveness of the proposed method with regard to robustness, efficiency, accuracy, and generality.

I. Introduction

An important branch in the field of structural optimization is constraint approximations. These are introduced so that approximate but explicit expressions can be found based on local or global information of the constraint functions in the design space. A suitable optimization method is then applied to the approximate problem to find its optimum. This new point in the design space can be used to form a new approximation and the procedure is repeated until the final optimum design is found. Each subproblem optimization can be seen as a direction finding process, which yields a direction and step size to find a better design. Early in the iterative procedure these steps may be excessive and therefore move limits are sometimes used to facilitate convergence. The methods, which may also be designated successive approximation methods, were first suggested for structural optimization by Schmit and Farshi[1] in 1974. Further development followed in the late 1970s[2] and recently a number of review papers[3–5] have appeared in which overviews of approximation concepts are given. Haftka et al.[5] distinguish between global (or multipoint) and local (single-point) approximations. Local approximations are convenient because derivatives of response quantities are required anyway at a design point for the direction seeking procedure in the optimization algorithm. Therefore additional structural analyses are usually not required to construct a local approximation at a design point. Some of the local approximations are first-order Taylor expansions (used in the sequential linear programming method), linear expansion of the reciprocal variables,[2,6,7] and a converse conservative approximation.[8] Multipoint methods (also called midrange approximations[9]) were introduced by Haftka et al.[5] to make use of previous design points in the design sequence. They concluded that their approximations worked well when interpolating but only gave marginal improvement in accuracy when extrapolating. More recently Fadel et al.[10] proposed a two-point exponential approximation that holds significant promise. As a result of this method they managed to relax the move limits significantly while needing few cycles for analysis.

Another method for structural optimization which uses approximations albeit in a different manner, and which has gained respect in recent years, is the sequential quadratic programming method (SQP). This method, in which the Lagrangian function is minimized, seems to feature most of the required properties of a good algorithm, namely robustness, generality, accuracy, ease of use, and efficiency.[11] Although a proven robust method, the SQP method is not a panacea since according to its users[11] it requires enormous effort to store Hessian information and also requires additional function evaluations to ensure convergence.

Seen in the context of the preceding approaches, the method presented here is a generic two-point, second-order successive approximation method. Surprisingly simple and not based on any structural considerations, the method attempts to be efficient in the sense that no explicit Hessian information is calculated or stored. In addition it requires no move limits while promising to be as fast as the SQP method in terms of the number of structural analyses. Basic to the proposed successive approximation method is the adaptation of the authors' recently proposed new feasible direction method (SSOPT),[12] to fully exploit the simple two-point, second-order approximations to the constraint functions. SSOPT is particularly elegant and efficient for the solution of problems with quadratic constraint approximations since the exact step sizes in the line searches can be calculated directly by solving a quadratic equation.

The performance of the new successive approximation method (hereafter referred to as SAM) is illustrated by means of a number of structural sizing problems with displacement and stress constraints.

II. Problem Statement and Method

Consider the typical sizing optimization problem:

$$\text{minimize } f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} \tag{1}$$

subject to the general constraints

$$g_j(\boldsymbol{x}) \le 0, \qquad j = 1, 2, \ldots, m \tag{2}$$

where $\boldsymbol{c}$ and $\boldsymbol{x}$ are column vectors in $\mathcal{R}^n$, f, and g_j scalar functions of $\boldsymbol{x}$, and T denotes the transpose. A trial design $\boldsymbol{x}^{(1)}$ is available. The solution to the problem is denoted by $\boldsymbol{x}^*$. In this paper, a vector component x_i typically represents a member cross-sectional area.

A. Constraint Approximations

A succession of quadratic subproblems, $P[k] : k = 1, 2, \ldots$, is formed at successive design points $\boldsymbol{x}^{(k)}$ by using the constraint ap-

Received Jan. 18, 1993; presented as Paper 93-1354 at the AIAA/ASME/ASCE/AHS/ASC 34th Structures, Structural Dynamics, and Materials Conference, La Jolla, CA, April 19–21, 1993; revision received Oct. 8, 1993; accepted for publication Oct. 8, 1993.

*Professor, Department of Mechanical Engineering, Structural Optimization Research Group.

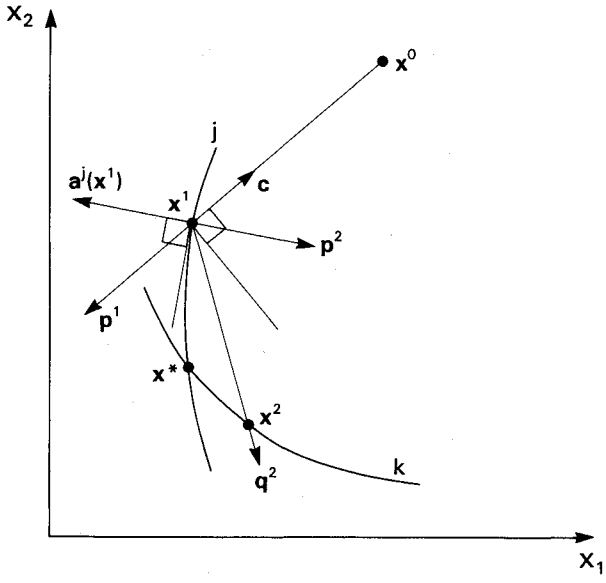

Fig. 1 Initial progress of the interior feasible direction method.

proximations $\tilde{g}_j(\boldsymbol{x})$ to $g_j(\boldsymbol{x})$ given by

$$\tilde{g}_j(\boldsymbol{x}) = g_j(\boldsymbol{x}^{(k)}) + \nabla^T g_j(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)}) + c_j^{(k)} \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \qquad j = 1, 2, \ldots, m \tag{3}$$

where ∇g_j denotes the gradient vector.

The second-order term given here may be seen as a simplification of the full second-order term in the Taylor series expansion

$$g_j(\boldsymbol{x}) = g_j(\boldsymbol{x}^{(k)}) + \nabla^T g_j(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)}) + (\boldsymbol{x} - \boldsymbol{x}^{(k)})^T \boldsymbol{H}_j^{(k)} (\boldsymbol{x} - \boldsymbol{x}^{(k)}) + \ldots \qquad j = 1, 2, \ldots, m \tag{4}$$

in which the Hessian matrix $\boldsymbol{H}$ is represented by the simple approximation

$$\tilde{\boldsymbol{H}}_j^{(k)} = c_j^{(k)} \boldsymbol{I}_n \tag{5}$$

where $\boldsymbol{I}_n$ is an $n \times n$ unit matrix.

These approximated constraints together with the objective function $f(\boldsymbol{x})$ define the subproblem $P[k]$, which is solved to find the Kuhn-Tucker point $\boldsymbol{x}^{(k)*}$. To find this point, the new and robust interior feasible direction method by Stander and Snyman[12] is used. This method simplifies and becomes highly efficient when applied to quadratic subproblems because the line searches inherent in the method can be computed directly. In the next loop a new approximate subproblem is formed by analyzing the design $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)*}$ and the new problem $P[k+1]$ is solved again. This succession is continued until $\boldsymbol{x}^{(k)*}$ converges to the solution $\boldsymbol{x}^*$. A convergence criterion

$$\frac{\left| f(\boldsymbol{x}^{(k+1)*}) - (f\boldsymbol{x}^{(k)*}) \right|}{f(\boldsymbol{x}^{(k+1)*})} < \varepsilon_s \tag{6}$$

based on the cost values of the two most recent subproblem solutions and where ε_s is a specified small positive value, was used. A detailed flow diagram of the algorithm is given in Sec. II.D.

An important question is the calculation of an appropriate $c_j^{(k)}$ for each constraint. In the first iteration $c_j^{(1)}$ can be assumed to be an arbitrary number for all $j = 1, 2, \ldots, m$. A value $c_j^{(1)} = 0$ can be used in which case the first subproblem becomes a linear programming problem. In the numerical experiments that follow, a value $c_j^{(1)} = 1$ was used throughout because it was found that a linear approximation resulted in an excessive size for the first step. Thereafter the coefficients $c_j^{(k)}$, $k = 2, 3, \ldots$, can be approximated from the approximate second-order Taylor expansion

$$g_j(\boldsymbol{x}^{(k-1)}) = g_j(\boldsymbol{x}^{(k)}) + \nabla^T g_j(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)}) + c_j^{(k)} \|\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)}\|^2 \qquad j = 1, 2, \ldots, m \tag{7}$$

along the previous search direction $(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)})$ so that written as the subject of the equation they become

$$c_j^{(k)} = \frac{g_j(\boldsymbol{x}^{(k-1)}) - g_j(\boldsymbol{x}^{(k)}) - \nabla^T g_j(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)})}{\left\| \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)} \right\|^2} \tag{8}$$

The Taylor expansion is written in terms of the reverse direction $\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)}$ so that the most recent gradient vector $\nabla g_j(\boldsymbol{x}^{(k)})$ can be used in the approximation.

The approximation of Eq. (3) can therefore be viewed as a second-order Taylor expansion in the direction $\boldsymbol{x}^{(k-1)} - \boldsymbol{x}^{(k)}$ in which the curvature term $c_j^{(k)}$ is approximated numerically from function values at the points $\boldsymbol{x}^{(k-1)}$ and $\boldsymbol{x}^{(k)}$ and the gradient vector at $\boldsymbol{x}^{(k)}$.

The question arises as to the validity of the assumption when the new search direction emanating from the current design $\boldsymbol{x}^{(k)}$ differs from $\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$. Because of the simplicity of the assumption, $c_j^{(k)}$ become less representative when there is significant deviation of this new search direction. This loss of accuracy is however not crucial since the first-order information remains intact and the spherical nature of the approximation potentially acts as an inherent move limit. Also, in the neighborhood of the optimal design the importance of the second-order information (which may be deficient) diminishes.

Since the solution of the quadratic subproblem is an important part of the successive solution procedure outlined, it is necessary to give some details of the method used to solve the subproblem.

B. Quadratic Subproblem

The quadratic problem is solved by using the interior feasible direction method SSOPT described in great detail in Ref. 12. A

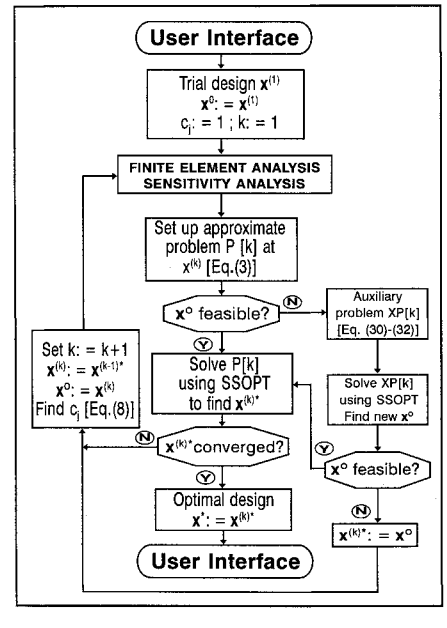

Fig. 2 SAM algorithm.

simple introduction to the method is presented here. The initial progress of the method is illustrated in Fig. 1.

Starting from $\boldsymbol{x}^0$ move in the direction of the steepest descent until the first (nearest) constraint hypersurface, say $g_j(\boldsymbol{x}) = 0$, is met at $\boldsymbol{x}^1$. Denote the gradient of $g_j(\boldsymbol{x})$ at $\boldsymbol{x}^1$ by $\boldsymbol{a}^j(\boldsymbol{x}^1)$ and for convenience let $\cdot$ denote the scalar product that follows. Feasibility as well as descent may now be ensured by computing a direction for the next step which lies within the wedge defined by the constraint tangent plane

$$\boldsymbol{a}^{j,1} \cdot (\boldsymbol{x} - \boldsymbol{x}^1) = 0 \tag{9}$$

where $\boldsymbol{a}^{j,1} = \boldsymbol{a}^j(\boldsymbol{x}^1)$, and the hyperplane of constant objective function value

$$\boldsymbol{c} \cdot (\boldsymbol{x} - \boldsymbol{x}^1) = 0 \tag{10}$$

through the point $\boldsymbol{x}^1$. A suitable choice would be a direction which points away equally from the respective planes. This direction, denoted by $\boldsymbol{q}^2$, may be computed as follows.

The unit vector orthogonal to the constant objective function plane and pointing in the direction of descent is given by $\boldsymbol{p}^1 = -\boldsymbol{c}/\|\boldsymbol{c}\|$ and the unit vector orthogonal to the constraint plane and pointing in the direction of the feasibility is $\boldsymbol{p}^2 = -\boldsymbol{a}^{j,1}/\|\boldsymbol{a}^{j,1}\|$. The required direction $\boldsymbol{q}^2$ is then given by a linear combination of $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$:

$$\boldsymbol{q}^2 = \alpha_1 \boldsymbol{p}^1 + \alpha_2 \boldsymbol{p}^2 \tag{11}$$

such that

$$\boldsymbol{q}^2 \cdot \boldsymbol{p}^1 = \boldsymbol{q}^2 \cdot \boldsymbol{p}^2 = \delta \tag{12}$$

for some arbitrarily chosen $\delta > 0$. The choice $\delta = 1$ may be made. The latter condition (12) specifies that the direction $\boldsymbol{q}^2$ points away equally from the respective two planes. This condition together with Eq. (11) reduces to

$$\begin{bmatrix} (\boldsymbol{p}^1 \cdot \boldsymbol{p}^1) & (\boldsymbol{p}^1 \cdot \boldsymbol{p}^2) \\ (\boldsymbol{p}^2 \cdot \boldsymbol{p}^1) & (\boldsymbol{p}^2 \cdot \boldsymbol{p}^2) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{13}$$

Substitution of the solution of Eq. (13) into Eq. (11) gives the required feasible descent direction (in the simple initial step, the various α are equal, $\alpha_1 = \alpha_2$, but this is not generally the case for more than two bounding planes).

Table 1 Problem size data

Problem	References	Bars	n	m	$x_i^{(1)}$
1	15,16	3	3	33	10;5;5
2	16	10	10	46	2
3	17	25	7	179	2.22
4	16	200	29	2129	See Ref.

Table 2 Algorithms used for comparison

Algorithm	Brief description
PLBA	SQP by using a modified Pschenichny's linearization method.[14]
IDESIGN3	SQP based on Pschenichny's descent function. Potential constraint strategy used.[15]
NPSOL	SQP.
NAG-E04UCF	SQP, Numerical Analysis Group, Mark 14B, (default parameter settings).[18]
SSOPT	Interior feasible direction method based on an equidistant search rule.[12]
SAM	Current algorithm.

Table 3 Results, minimum weight f^*

	3-bar truss		10-bar truss		25-bar truss		200-bar truss	
	NFE	f^*, lb	NFE	f^*, lb	NFE	f^*, lb	NFE	f^*, lb
PLBA	—	—	51	5061.0	17	545.00	31	25,450
IDESIGN3	—	—	39	5060.9	17	545.03	34	25,449
NPSOL	—	—	50	5060.9	12	545.03	15	25,447
NAG	8	20.544	41	5060.9	26	545.04	36	25,449
SSOPT	70	20.544	574	5060.9	281	545.04	1728	25,447
SAM	8	20.544	38	5060.9	27	545.04	26	25,449

Now moving away from $\boldsymbol{x}^1$ in the new descent direction $\boldsymbol{q}^2$, a second constraint will be encountered at $\boldsymbol{x}^2$, say $g_k(\boldsymbol{x}) = 0$, unless the problem is unbounded. If a nonzero step is taken to $\boldsymbol{x}^2$ it will result in a decrease in the objective function as well as moving away from the first encountered constraint tangent plane. Denote the gradient of $g_k(\boldsymbol{x})$ at $\boldsymbol{x}^2$ by $\boldsymbol{a}^{k,2}$. At $\boldsymbol{x}^2$ the next feasible descent direction may be determined as follows. The tangent plane of the first encountered constraint at $\boldsymbol{x}^1$ is translated to $\boldsymbol{x}^2$ and it is required that the new direction $\boldsymbol{q}^3$ for the next feasible step at $\boldsymbol{x}^2$ be computed to lie within the feasible cone (difficult to depict in a two-dimensional figure), defined by the tangent plane of the currently encountered constraint at $\boldsymbol{x}^2$

$$\boldsymbol{a}^{k,2} \cdot (\boldsymbol{x} - \boldsymbol{x}^2) = 0 \tag{14}$$

the tangent plane of the first constraint moved to $\boldsymbol{x}^2$ and the plane of constant objective function at $\boldsymbol{x}^2$. Again it is required that $\boldsymbol{q}^3$ point away equally from the three planes defining the cone, i.e., $\boldsymbol{q}^3$ is given by the linear combination

$$\boldsymbol{q}^3 = \alpha_1 \boldsymbol{p}^1 + \alpha_2 \boldsymbol{p}^2 + \alpha_3 \boldsymbol{p}^3 \tag{15}$$

where $\boldsymbol{p}^3 = -\boldsymbol{a}^{k,2}/\|\boldsymbol{a}^{k,2}\|$ and the various α chosen such that

$$\begin{bmatrix} (\boldsymbol{p}^1 \cdot \boldsymbol{p}^1) & (\boldsymbol{p}^1 \cdot \boldsymbol{p}^2) & (\boldsymbol{p}^1 \cdot \boldsymbol{p}^3) \\ (\boldsymbol{p}^2 \cdot \boldsymbol{p}^1) & (\boldsymbol{p}^2 \cdot \boldsymbol{p}^2) & (\boldsymbol{p}^2 \cdot \boldsymbol{p}^3) \\ (\boldsymbol{p}^3 \cdot \boldsymbol{p}^1) & (\boldsymbol{p}^3 \cdot \boldsymbol{p}^2) & (\boldsymbol{p}^3 \cdot \boldsymbol{p}^3) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{16}$$

which may be written more concisely as

$$\boldsymbol{P}^3 \alpha^3 = \boldsymbol{e}^3 \tag{17}$$

where $\boldsymbol{e}^3 = [1, 1, 1]^T$. This procedure ensures not only a descent direction but also avoids, in the case of smooth constraints, that the last encountered constraint be immediately met again in the new descent direction. Continuing in this way and having identified $\boldsymbol{x}^1$, in the initial descent step, a sequence of n boundary points $\boldsymbol{x}^1$, $\boldsymbol{x}^2, \ldots, \boldsymbol{x}^n$ on the bounding constraints may be identified by successively solving

$$\boldsymbol{P}^r \alpha^r = \boldsymbol{e}^r, \quad r = 2, 3, \ldots, n \tag{18}$$

and determining the nearest boundary point $\boldsymbol{x}^r$ by moving from $\boldsymbol{x}^{r-1}$ in the direction

$$\boldsymbol{q}^r = \sum_{j=1}^{r} \alpha_j^r \boldsymbol{p}^j \tag{19}$$

The normalized search directions are denoted by

$$\boldsymbol{s}^r = \frac{\boldsymbol{q}^r}{\|\boldsymbol{q}^r\|} \tag{20}$$

The whole procedure starting with $\boldsymbol{x}^0$ and ending with $\boldsymbol{x}^n$ is referred to as a cycle. Note that, since $\boldsymbol{P}^r$ is obtained from $\boldsymbol{P}^{r-1}$ by

simply adding the symmetrically related bordering r row and r column vectors, the solutions of the systems (18), in terms of the number of computations required, effectively reduce to the solution of a single system

$$P^n\alpha^n = e^n \tag{21}$$

once per cycle. On completion of a cycle, the overall procedure may be terminated if no significant progress was made during the previous cycle, i.e., if

$$\frac{f(x^0) - f(x^n)}{f(x^n)} < \varepsilon_b \tag{22}$$

for an arbitrarily prescribed small positive number ε_b. If the termination criteria (22) are not met, a new cycle is started with $x^0 := x^n$, and the overall iterative procedure is continued.

Throughout this introduction it was assumed that the matrix P^r in Eq. (18) never becomes singular. In many cases, such as in the application of the method to the structural optimization problems presented here, it may occur. Should P^r become singular, the cycle is terminated and a new cycle started with $x^0 := x^{r-1}$. The latter strategy is justified by the following informal argument: If P^r becomes singular at x^{r-1} then either c is a linear combination of the gradients of the encountered constraints at the boundary points x^1, $x^2, \ldots, x^{r-1}$ (case i) or the gradients of the encountered constraints are mutually linear dependent (case ii).

If in case i $x^{r-1} = x^*$ then the feasible region has obviously shrunk to zero and no further descent is possible during the next cycle. In the absence of degeneracy P^r for the next cycle $k+1$ would be equal to the P^r of the previous cycle, i.e. $P^{(k+1)r} = P^{(k)r}$ and $f(x^{(k+1)r}) = f(x^{(k)r})$, which may be taken as the termination condition.

If in case i $x^{r-1} \neq x^*$, and assuming that in the sequence x^0 to x^{r-1} a nonzero step was taken, then on restarting a cycle it is impossible to have $P^{(k+1)r} \equiv P^{(k)r}$. Thus the possibility of constructing an identical singular matrix P^r on restart is avoided.

Also in case ii, provided a nonzero step was taken during the sequence x^0 to x^{r-1}, then again, in the next cycle, $P^{(k+1)r} \not\equiv P^{(k)r}$.

The only remaining complication that may occur is the highly unlikely event of x^0 coinciding with an exact degenerate point such that $x^{r-1} \equiv x^0$, P^r singular and x^0 nonoptimal. In this case the problem may be avoided in the usual way by solving for the Lagrange multipliers of the active constraints at x^0 and dropping the constraint associated with the most negative value. Not surprisingly this unlikely geometrical situation was never encountered in practice, and the special Lagrange multiplier routine was therefore never required. This completes the argument for restarting a cycle whenever P^r becomes singular.

An important question is the assessment of the conditioning of P^r to decide whether a new cycle should be started or not. A tolerance on the pivots formed during the factorization of P^r can be used, but this approach is unsatisfactory due to the difficulty of choosing a suitable value for the tolerance. Instead the feasibility and usability criteria

$$q^r \cdot p^r > 0 \tag{23}$$

$$q^r \cdot p^1 > 0 \tag{24}$$

can be used where q^r has been obtained from Eq. (19). In the current implementation a new cycle is started whenever a pivot is exactly zero [in which case Eqs. (18) cannot be solved] or when the

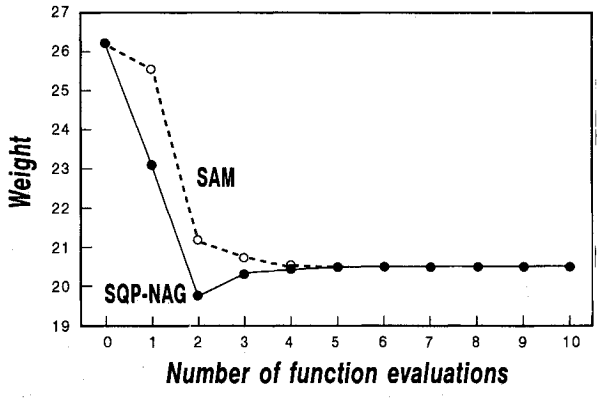

Fig. 3 Optimization history for the 3-bar truss.

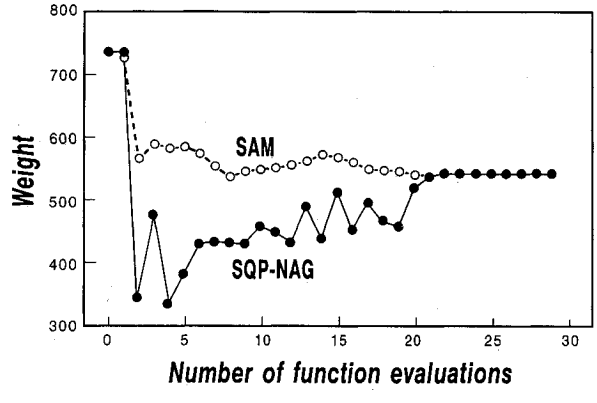

Fig. 5 Optimization history for the 25-bar truss.

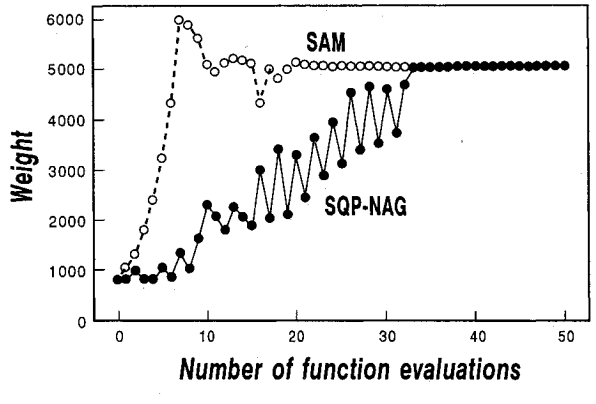

Fig. 4 Optimization history for the 10-bar truss.

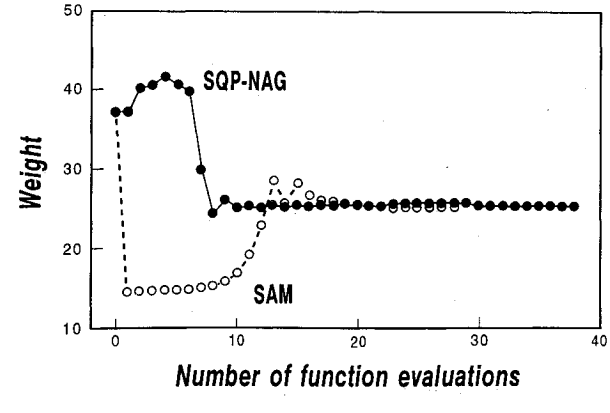

Fig. 6 Optimization history for the 200-bar truss.

Table 4 10-bar truss; different starting points (see also Fig. 7)

Symbol	Starting point $x^{(1)}$	NFE	f^*
A	2, 2, ..., 2	38	5060.9
B	20, 20, ..., 20	50	5060.9
C	1, 2, 3, ..., 10	50	5060.9
D	0.101, 0.101, ..., 0.101	42	5060.9
E	1000, 1000, ..., 1000	35	5060.9

given criteria are violated as a result of the inexact computation of the values of the various α_j due to the near singularity of $\boldsymbol{P}^r$.

As the subproblem to which SSOPT is applied is quadratic, the line searches may be conducted by using simple formulas. If λ is the scalar step size in the direction $\boldsymbol{x} - \boldsymbol{x}^{r-1}$ so that

$$\lambda \boldsymbol{s}^r = \boldsymbol{x} - \boldsymbol{x}^{r-1} \tag{25}$$

where $\boldsymbol{s}^r$ is the unit search vector and $\boldsymbol{x}^{r-1}$ is the design point from which the search emanates, then the constraint approximation in terms of λ can be given as

$$\tilde{g}(\boldsymbol{x}) = \tilde{g}(\boldsymbol{x}^{r-1}) + \nabla^T \tilde{g}(\boldsymbol{x}^{r-1})\, \boldsymbol{s}^r \lambda + (1/2)\, c\lambda^2 \tag{26}$$

where the j subscripts denoting constraint number have been omitted for simplicity. Further simplified, Eq. (26) can also be written as

$$G(\lambda) = g(\boldsymbol{x}^{r-1} + \lambda \boldsymbol{s}^r) = G(0) + G'(0)\, \lambda + (1/2)\, c\lambda^2 \tag{27}$$

where

$$G'(0) = \nabla^T \tilde{g}(\boldsymbol{x}^{r-1}) \boldsymbol{s} \tag{28}$$

Setting $G(\lambda) = 0$ for the line search results in

$$\lambda = \begin{cases} \dfrac{-G'(0) \pm \sqrt{G'(0)^2 - 2G(0)\, c}}{c}; & c \neq 0 \\ \dfrac{-G(0)}{G'(0)}; & c = 0 \end{cases} \tag{29}$$

Because the current design is feasible in terms of the subproblem, only the larger root of $G(\lambda) = 0$ is required.

A difficulty with the subproblem solution is that a feasible starting point is required for the subproblem. It can readily be found by setting up an auxiliary problem .

C. Auxiliary Problem: *XP*

With reference to Eqs. (2) and (3), the auxiliary subproblem[13] is defined as

$$\text{minimize } y \tag{30}$$

subject to the $m + 1$ constraints

$$\tilde{g}_j(\boldsymbol{x}) - y \leq 0, \quad j = 1, 2, \ldots, m \tag{31}$$

$$-y \leq 0, \quad \text{constraint } m + 1 \tag{32}$$

where y is an $(n + 1)$th (auxiliary) variable. The feasible starting point for the auxiliary problem is $\boldsymbol{x}^{00} \in \mathcal{R}^n$ and y^0 where y^0 is selected so that $y^0 = \max[0, \tilde{g}_j(\boldsymbol{x}^{00}); j = 1, 2, \ldots, m]$ and $\boldsymbol{x}^{00}$ is the (infeasible) design. The solution procedure SSOPT is used to compute the solution $\boldsymbol{x}^0$. Ideally, convergence is obtained when constraint $m + 1$ is encountered as bounding constraint making $y = 0$. Such a solution to the auxiliary problem clearly constitutes a feasible design $\boldsymbol{x}^0$ for the subproblem.

It may happen that a feasible solution cannot be found because the solution to the auxiliary problem $\boldsymbol{x}^0$ corresponds to the $(m+1)$th constraint being inactive and therefore $y > 0$. The auxiliary problem solution with reduced value of y then represents the most feasible design which can be used to construct a new subproblem. In all such cases which occurred in the test problems, the resultant sequence of auxiliary problems culminated in an eventual feasible starting design for a subproblem.

To limit the sizing variables x_i to the positive range of the design space, Eqs. (31) and (32) were modified to

$$\tilde{g}_j(\boldsymbol{x}) - y \leq 0, \quad j = 1, 2, \ldots, m - n \tag{33}$$

$$-x_k + \underline{x}_k \leq 0, \quad k = 1, \ldots, n \tag{34}$$

$$-y \leq 0, \quad \text{constraint } m + 1 \tag{35}$$

where $\tilde{g}_j$ approximates the behavioral constraints and $\underline{x}_k$ represents lower limits on the design variables.

The auxiliary problem formulation is easily incorporated into the SSOPT algorithm by simply augmenting the matrices $\tilde{\boldsymbol{g}}$ and $\nabla\tilde{\boldsymbol{g}}$ which contain m constraints and $n \times m$ gradient terms, respectively. The auxiliary problem matrices are

$$\tilde{\boldsymbol{g}}^A = \begin{bmatrix} \tilde{\boldsymbol{g}} - y\mathbf{1} \\ \hline -\boldsymbol{x} + \underline{\boldsymbol{x}} \\ \hline -y \end{bmatrix}; \quad \nabla\tilde{\boldsymbol{g}}^A = \left[\begin{array}{c|c|c} \nabla\tilde{\boldsymbol{g}} & -\boldsymbol{I} & 0 \\ \hline -\mathbf{1}^T & \mathbf{0}^T & -1 \end{array}\right] \tag{36}$$

where $\tilde{\boldsymbol{g}}^A$ has dimensions $[(m - n) + (n) + (1)] \times [1]$ and $\nabla\tilde{\boldsymbol{g}}^A$ has dimensions $[(n) + (1)] \times [(m - n) + (n) + (1)]$. $\mathbf{1}$ is a vector with unit components and of dimension $m - n$. As the auxiliary problem is also quadratic, line searches are conducted in a manner similar to that when the true subproblem is optimized as described here.

D. Successive Approximation Method Algorithm

The preceding formulations are embodied in a SAM algorithm of which the logic is as shown in Fig. 2.

III. Test Problems and Results

Four sizing test problems were chosen to demonstrate the method. The problems are all well-documented standard truss structures subjected to both displacement and stress constraints. The problem size data are summarized in Table 1. Further details concerning the exact data can be obtained from the literature sources also referred to in Table 1.

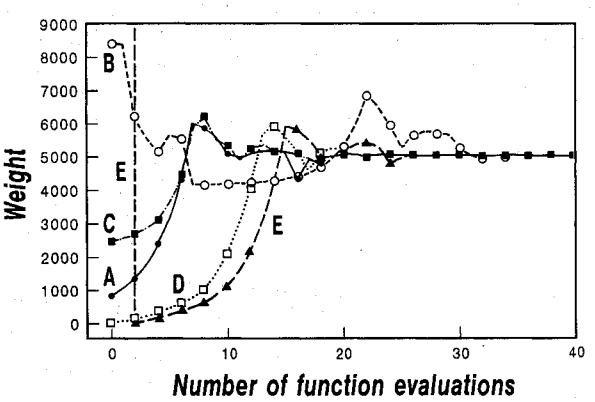

Fig. 7 Optimization history for the 10-bar truss, different starting points (see also Table 4).

A number of recently published algorithms which also use approximations are listed in Table 2. Their performance, with regard to accuracy and efficiency, is compared with that of the SAM algorithm in Table 3. In addition to the methods using approximations, the results of applying SSOPT without approximations and with exact line searches are also included in Table 3. A value of $\varepsilon_s = 10^{-5}$ was used throughout.

The norm used to compare efficiency is number of function evaluations (NFE) which corresponds to the number of structural analyses. This is not necessarily a good basis for comparison because in all algorithms mentioned significant computational effort can also be attributed to other operations such as, for example, the computations required to solve any subproblem. A truer picture is given by Figs. 3–6 in which the NAG-SQP method (algorithm E04UCF; see Ref. 18) and the SAM convergence behavior are compared. It can be seen that overall (in terms of the number of structural analyses) SAM converges as fast as the SQP method. Since all of the function evaluations required to set up auxiliary subproblems are also included in the plots, these can be easily recognized as forming the ascending curves (e.g., Fig. 6).

Starting point sensitivity was also tested, and Table 4 (see illustration in Fig. 7) gives results for the 10-bar truss using a variety of both feasible and infeasible starting points. Curves *A*, *C*, and *D* of Fig. 7 also show the typical ascending curves associated with severely infeasible starting designs. Although the initial design *E* is overly conservative it plunges to a very low weight represented by an infeasible design early in the optimization process. This is probably because the initial design is too far from optimal and therefore the approximations are severely inaccurate. It is not apparent why design curve *B* lingers before converging although it is clear that the ascending curve which it negotiates after becoming infeasible is relatively extended. This variation in behavior seems to be entirely dependent on the randomness of positions of intermediate designs in the design space. In spite of variations in performance, all of the trial designs finally led to convergence within roughly the same total number of finite element analyses.

IV. Conclusion

An optimization method for problems with linear objective functions and nonlinear constraint functions is presented. The method is based on a generic two-point constraint approximation of each constraint function in which the second-order Taylor series term is approximated using a diagonal Hessian with identical elements.

The successive approximation method (SAM) has been shown to efficiently yield accurate results for a number of standard sizing problems with stress and displacement constraints. One of the main advantages of the method is that it is easy to use because, except for the convergence tolerance, it does not need any user adjustable parameter. The method seems to be robust because accurate results were obtained from any feasible or infeasible starting point without applying a move limit. It is also efficient because construction of a Hessian matrix is not required and the constraint approximations are simple enough to allow direct and exact line searches when a feasible directions method is used for optimization of the subproblem. For large problems this aspect becomes more important, as one may expect that the subproblems of the current method may be solved much more efficiently than the subproblems occurring in the established SQP methods, which require the manipulation of Hessian information. As was demonstrated previously, the method is accurate because it succeeded in obtaining known local minima to all of the problems considered here.

To generalize the algorithm, it must be modified to accommodate a nonlinear objective function (e.g., for shape optimization). This work is currently in progress.

References

[1]Schmit, L. A., and Farshi B., "Some Approximation Concepts for Structural Synthesis," *AIAA Journal*, Vol. 12, No. 5, 1974, pp. 692–699.

[2]Schmit, L. A., and Miura, H. "Approximation Concepts for Efficient Structural Synthesis," NASA CR-2552, March 1976.

[3]Prasad, B., "Explicit Constraint Approximation Forms in Structural Optimization. Part 1: Analyses and Projections," *Computer Methods in Applied Mechanics and Engineering*, Vol. 40, No. 1, 1983, pp. 1–26.

[4]Prasad, B., "Explicit Constraint Approximation Forms in Structural Optimization. Part 2: Numerical Experiences," *Computer Methods in Applied Mechanics and Engineering*, Vol. 46, No. 1, 1984, pp. 15–38.

[5]Haftka, R. T., Nachlas, J. A., Watson, L. T., Rizzo, T., and Desai, R., "Two-Point Constraint Approximation in Structural Optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 60, No. 3, 1987, pp. 289–301.

[6]Storaasli, O. O., and Sobieszcanski, J. O., "On the Accuracy of the Taylor Approximation for Structure Resizing," *AIAA Journal*, Vol. 12, No. 2, 1974, pp. 231–233.

[7]Noor, A. K., and Lowder, H. E., "Structural Reanalysis via a Mixed Method," *Computers & Structures*, Vol. 5, No. 1, 1975, pp. 9–12.

[8]Starnes, J. H., and Haftka, R. T., "Preliminary Design of Composite Wings for Buckling Stress and Displacement Constraints," *Journal of Aircraft*, Vol. 16, No. 5, 1979, pp. 564–570.

[9]Haftka, R. T., and Gürdal, Z., *Elements of Structural Optimization*, Kluwer Academic Publishers, Boston, 1992, p. 221.

[10]Fadel, G. M., Riley, M. F., Barthelemy, J. M., "Two-Point Exponential Approximation Method for Structural Optimization," *Structural Optimization*, Vol. 2, No. 2, 1990, pp. 117–124.

[11]Thanedar, P. B., Arora, J. S., Li, G. Y., and Lin, T. C., "Robustness, Generality and Efficiency of Optimization Algorithms for Practical Applications," *Structural Optimization*, Vol. 2, No. 4, 1990, pp. 203–212.

[12]Stander, N., and Snyman, J. A., "A New First Order Interior Feasible Direction Method for Structural Optimization," *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 23, No. 12, 1993, p. 17.

[13]Chvatal, V., *Linear Programming*, W. H. Freeman, New York, 1983, p. 39.

[14]Thanedar, P .B., Arora, J. S., Tseng, C. H., Lim, O. K., and Park, J. J., "Performance of Some SQP Algorithms on Structural Design Problems," *International Journal for Numerical Methods in Engineering*, Vol. 23, No. 12, 1986, pp. 2187–2203.

[15]Arora, J. S., *Introduction to Optimum Design*, 1st ed., McGraw-Hill, 1989, pp. 489–494.

[16]Sunar, M., and Belegundu, A. D., "Trust Region Methods for Structural Optimization Using Exact Second Order Sensitivity," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 2, 1991, pp. 275–293.

[17]Haug, A., and Arora, J. S., *Applied Optimal Design*, Wiley, New York, 1979, pp. 245–250.

[18]Anon., *The NAG Fortran Library Manual, Mark 14*, Numerical Algorithms Group, Oxford, England, UK, 1990, pp. E04UCF1-E04UCF41.