A Mixed-Integer Nonlinear Programming Algorithm for Process Systems Synthesis

The problem of synthesizing processing systems via simultaneous structural and parameter optimization is addressed in this paper. Based on a superstructure representation for embedding alternative configurations, a general mixed-integer nonlinear programming (MINLP) framework is presented for the synthesis problem. An efficient outer-approximation algorithm is described for the solution of the underlying optimization problem, which is characterized by linear binary variables and continuous variables that appear in nonlinear functions. The proposed algorithm is based on a bounding sequence that requires the analysis of few system configurations, and the solution of a master problem that identifies new candidate structures. Application of the proposed algorithm is illustrated with the optimal synthesis of gas pipelines.

M. A. Duran and I. E. Grossmann
Department of Chemical Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213

SCOPE

Synthesis is perhaps the cornerstone of the process design activity since it addresses the fundamental problem of structuring a processing scheme so as to satisfy given goals and/or needs. This relatively new research area in chemical engineering has received considerable attention in the literature; see Nishida et al. (1981) for a review. Theoretical as well as applications-related work in this area is currently the subject of major efforts. The main approaches that have emerged for tackling process synthesis problems are the use of heuristics, thermodynamic targets, and algorithmic methods that are based on optimization techniques (Stephanopoulos, 1981). As indicated by Grossmann (1985), the first two approaches have been used quite extensively with some important successes despite their obvious limitations, such as the fact of not being able to assert the quality of the solution, the assumption on dominance of energy costs, and the restricted application to specific subproblems. Algorithmic methods, on the other hand, offer a more general and systematic approach since they explicitly account for the economic trade-offs and interactions in the synthesis of arbitrary processing systems. Furthermore, because of their nature these methods can accommodate the other two approaches and are better suited for automatic synthesis of systems, as has recently been shown in the mixed-integer linear programming (MILP) framework proposed by Papoulias and Grossmann (1983a,b,c). These authors developed MILP formulations for utility systems, heat recovery networks, and integrated total processing systems. An important limitation of these formulations is that nonlinearities cannot be handled explicitly as they require the discretization of those variables that give rise to nonlinear functions. Thus, there is a need to develop efficient optimization procedures that can handle discrete and continuous variables in nonlinear models for the synthesis of process systems.

This paper addresses the problem of developing an efficient solution procedure for algorithmic methods for the synthesis of process systems. Based on the modeling of a superstructure of alternatives as a mixed-integer nonlinear programming (MINLP) program in which the binary variables appear linearly and the continuous variables are involved in nonlinear functions, an

Correspondence concerning this paper should be addressed to I. E. Grossmann.
M. A. Duran's current address is Universidad Autonoma Metropolitana-Iztapalapa, Apdo. Postal 55-534, 09340 Mexico City.

algorithm is presented for the simultaneous structural and parameter optimization of processing systems. This algorithm consists of an alternating sequence of NLP subproblems, and MILP master problems. In the former parameter optimization is performed for fixed configurations, while in the latter configurations that have the potential of being optimal are identified. The application of this algorithm is illustrated with the optimal synthesis of gas pipelines where both the configuration and design parameters must be selected.

CONCLUSIONS AND SIGNIFICANCE

It has been shown that the underlying optimization problem for the systematic synthesis of processing systems that rely on a superstructure representation can be modeled by an MINLP program in which the binary variables appear linearly and the continuous variables are involved in nonlinear functions. To solve this class of problems an outer-approximation algorithm has been proposed that consists of a bounding sequence that is based on NLP subproblems for the detailed analysis of specific configurations, and MILP master problems for generating new candidate configurations. Although the algorithm can be applied to any type of nonlinear functions for the continuous variables, a rigorous guarantee for the global optimal solution can only be provided for the case when the nonlinear functions are convex.

The application of the outer-approximation algorithm has been illustrated with the optimal synthesis of gas pipelines in which both the configuration and parameters must be selected. This has been formulated as an MINLP problem that has the mathematical structure of linear binary variables, and continuous variables that appear in nonlinear convex inequalities. As was shown with a numerical example, due to the good quality of the lower bounds predicted by the master problem, the proposed algorithm requires the detailed analysis of only a few pipeline configurations to find the global optimal solution.

The significance of the proposed algorithm is that it is an efficient method that can handle explicitly nonlinear functions involved in MINLP formulations of synthesis problems. This avoids the need of discretizing continuous variables that give rise to nonlinearities as is done in the MILP approach for process synthesis. Also, the algorithm is quite general in that it can be applied to other problems that involve the optimization of discrete and continuous variables. The application of the outer-approximation algorithm to the synthesis of a large integrated process example will be reported in a future paper.

Introduction

The formulation underlying the algorithmic approach based on mixed-integer programming for process synthesis involves two major steps (Grossmann, 1985). The first step consists in postulating as potential solutions several competitive alternative configurations for the processing system. The basic alternatives are in general the result of preliminary screening based on the use of heuristics, thermodynamic bounds, and/or design experience. The interconnection of these basic alternatives via a superstructure representation usually generates many additional potential configurations that expand the domain of analysis. Examples of superstructures for processing systems can be found in Papoulias and Grossmann (1983a,c) and Duran (1984). The second step in the formulation is the modeling of the postulated superstructure as a mixed-integer programming program that in general has the following mathematical form:

$$
\begin{aligned}
z = \min_{w,y} \quad & C(w,y) \\
\text{s.t.} \quad & r(w,y) = 0 \\
& s(w,y) \le 0 \\
w \in W \, C \, R_{+}^{nw}, \; & y \in D \, C \, R_{+}^{m}
\end{aligned}
\qquad \text{[P0]}
$$

In this formulation w are continuous variables associated with processing parameters such as flow rates, pressures, temperatures, and equipment sizing characteristics. The integer variables y represent discrete decisions in the problem. In particular, a subset of them consists of (0–1) binary variables associated with units in the superstructure. These variables then serve the purpose of distinguishing between alternatives as well as denoting the potential existence of system units in the final configuration. A value of one for a given binary variable implies that the corresponding unit will be included in a particular structure, while a value of zero will exclude it. The vectors of constraints r and s represent performance relationships for system units in the superstructure (e.g., energy and material balance equations, equilibrium relations), design specifications, physical constraints, and logical restrictions. W is a constrained set defined by usually known lower and upper bounds on the continuous variables. $D = \{y{:}y \in [\text{integer set}]\}$, is a finite discrete set structured as explained later in the discussion. The objective function C is typically a cost function involving both investment and operating costs.

The basic idea in this approach for the synthesis problem is therefore to extract from the superstructure the optimal system configuration and its operating parameters by solving the mixed-integer optimization program P0. This problem could in principle be solved with a general purpose branch-and-bound procedure (Beale, 1977; Gupta, 1980), or with the generalized Benders decomposition method of Geoffrion (1972). However, because these methods were intended for the solution of a rather broad class of mixed-integer programs, they cannot in general exploit effectively the special structure of synthesis problems.

MINLP of Special Structure

In its general form program P0 contains many formulations that can be rather complex and usually difficult to solve. That is, features such as nonlinearity in discrete variables and nonseparability of continuous and discrete variables may appear. Although some of these features can be handled using standard procedures such as special ordered sets or "linearization" methods (Beale, 1977; Beale and Tomlin, 1970; Glover, 1975) the actual formulation of synthesis problems very often yields programs with special mathematical structure.

The particular structure that usually arises is that the binary variables appear only linearly, and they are either involved in pure-integer constraints, or else are related to continuous variables through inequalities. The reason for this is as follows. In the objective function, investment cost functions for system units, see Figure 1, are commonly concave in some of the components of w that denote a given activity (e.g., flow rates, equipment sizes). That is, due to economies of scale some activities that cannot profitably be introduced on a small scale can nevertheless be worth introducing on a large scale. Thus, to avoid finding nonrealistic solutions, costs that could only be saved by the activity not taking place must be treated as fixed. Using the binary variables y_j that denote existence of system units, the following fixed-cost charge function approximation can be used for the investment cost of a given unit j,

$$C_j(w, y_j) = c_j y_j + p_j(w) \tag{1}$$

$$p_j(w) = 0 \quad \text{iff} \quad y_j = 0 \tag{2}$$

$$p_j(w) \geq 0 \quad \text{iff} \quad y_j = 1 \tag{3}$$

In this model, where the binary variable y_j appears linearly in Eq. 1, a positive fixed charge c_j is only incurred when the system unit j is present in a particular configuration, or equivalently when the corresponding activity levels in w are greater than zero. Since the variable cost, $p_j(w)$, which can be a linear or nonlinear function, is always nonnegative and monotonic (e.g., see Figure 1), the constraints in Eqs. 2 and 3 can in general be expressed as sets of linear inequalities of the following form,

$$w_i - u_i y_j \leq 0 \tag{4}$$

$$w_i \geq 0 \tag{5}$$

$$y_j \in \{0, 1\} \tag{6}$$

where w_i is a particular component of the vector w, and u_i is a known finite upper bound on w_i. Therefore, for a nonexistent unit j (i.e., $y_j = 0$) the implication is that some of the associated variables w_i must be zero (e.g., flow rates, size parameters). Reciprocally, when $w_i = 0$ holds in Eq. 4, minimization of C_j as given in Eq. 1 will imply that $y_j = 0$. The advantages of this fixed-cost charge model over other type of cost functions are discussed in the example section.

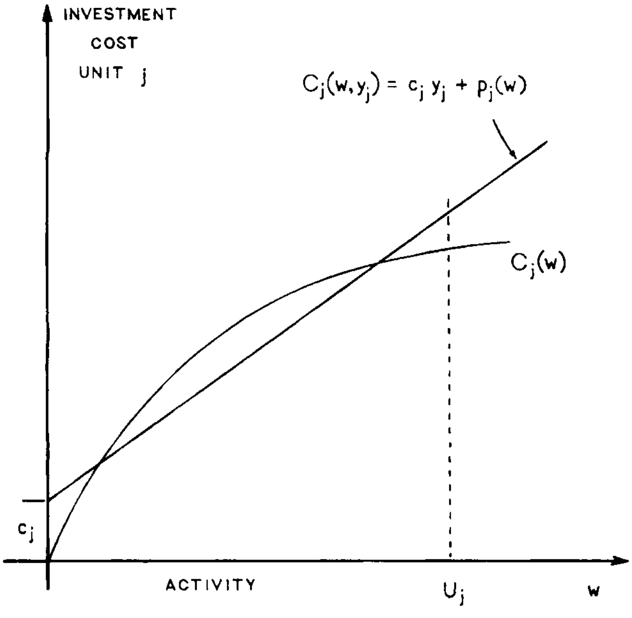

Figure 1. Fixed-cost charge function approximation.

For the mixed-integer formulation of the superstructure, binary variables acting linearly are also used for enforcing logical conditions related to what can be called structural or topological constraints. The basic elements of logical conditions are the fundamental operations "and," "or," and the implication "if . . . then . . . ." A simple example is the case when among a set S of system units only one of them must appear in the final configuration. This logical condition, which is a disjunction ("or"), can be expressed by the multiple choice constraint,

$$\sum_{j \in S} y_j = 1$$
$$y_j \in \{0, 1\} \tag{7}$$

Thus, without great loss of generality, it will be assumed that most synthesis problems can be modeled as mixed-integer nonlinear programming (MINLP) programs where the integer variables are usually binary, always appear in linear form, and are involved in the objective function, and in mixed (continuous-discrete) inequalities and pure-integer constraints as discussed above. Problem P0 reduces then to the following class of MINLP programs as the underlying model for the synthesis problem:

$$\begin{aligned} z = \min_{w,y} \quad & c^T y + p(w) \\ \text{s.t.} \quad & r(w) = 0 \\ & s(w) + By \leq 0 \\ & w \in W, y \in U \end{aligned} \tag{P1}$$

where the functions in the continuous variables (w), p, and those in the vectors r and s, can be of both types, linear and nonlinear. $U = \{y : y \in \{0, 1\}^m, Ay\ [\leq \text{or} =]\ a\}$ is a finite binary set given by pure-integer constraints. Some of the rows in B may be the zero row vector, which then defines constraints in only the continuous variables. It should be noted that the MINLP formulation P1 can be used to model a great variety of synthesis problems such as utility systems and integrated chemical processes (see Papoulias and Grossmann, 1983a,c; Duran, 1984).

Because the synthesis problem is an optimization problem, the vector of equality constraints $r(w) = 0$ in P1, which represent the performance of units in the superstructure, is an underdetermined system of equations, i.e., $\dim(w) > \dim(r)$. The vector of continuous variables w can therefore be partitioned as,

$$w^T = [v^T, x^T] \tag{8}$$

where v the vector of state variables, has the same dimensionality as the vector of equations $r(w) = 0$. The vector x is the set of decision variables, which correspond to degrees of freedom for the continuous variables in the superstructure. Provided the inverse transformation exists for the given partition of variables, the state variables can be expressed as a function of the decision variables x, as given by

$$r(w) = r(v, x) = 0 \quad \Longrightarrow \quad v = r'(x) \tag{9}$$

The transformation r' can be either an explicit or an implicit function of the state variables v. Whenever this transformation corresponds to an explicit relationship, the elimination of state variables is equivalent to algebraic variable substitution. When the transformation is implicit, this requires numerical elimination of the system of equations. Details for the implementation of the implicit form case, as well as comments on the selection of a set of decision variables can be found in Duran (1984).

The transformation in Eq. 9 then allows the elimination of the system of equations $r(v, x) = 0$ and also of the state variables v, so that the functions p and s in program P1 can be reformulated as functions defined in the reduced space of the decision variables x. That is,

$$\begin{aligned} p(w) &= p(v, x) = p(r'(x), x) = f(x) \\ s(w) &= x(v, x) = s(r'(x), x) = g(x) \end{aligned} \tag{10}$$

Therefore, in terms of the decision variables x, problem P1 can be reformulated as,

$$\begin{aligned} z = \min_{x,y} \quad & c^T y + f(x) \\ \text{s.t.} \quad & g(x) + By \leq 0 \\ & x \in X, y \in U \end{aligned} \tag{P2}$$

where $X \, C \, R^n_+$ is the n-dimensional set $X = \{x : [x, r'(x)] \in W, A_1 x \leq a_1\}$, and the functions f and those in the vector function $g: R^n \rightarrow R^p$ are assumed in general to be nonlinear. Thus, elimination of the equality constraints in program P1 allows not only a reduction on the dimensionality of the problem, but perhaps more important, it renders the equivalent formulation P2 where the nonlinear constraints are inequalities.

As will be shown in this paper, the particular mathematical structure in problem P2 can be exploited to derive an efficient algorithm for solving this MINLP program. For convenience in the presentation, unboundedness and infeasibility conditions for program P2 will not be addressed here, and it will therefore be assumed that P2 has a finite mixed-integer optimal solution.

OUTLINE OF THE PROPOSED ALGORITHM

The basic ideas behind the proposed algorithm for solving problems in the class represented by program P2 will be presented in this section in the context of the synthesis problem. Since the complexity (combinatorial nature) of the synthesis problem is due to the selection of structural parameters for a processing system, an intelligent search among alternative configurations has to be performed for an efficient solution.

The key idea in the proposed algorithm is to analyze a sequence of system configurations making use of a bounding scheme to successively reduce the number of candidate alternatives for the optimal solution. To accomplish this objective a master problem is used both to generate specific candidate configurations for detailed analysis, and to identify stopping criteria for the search of the optimal solution. Each detailed analysis corresponds to the optimization of the continuous variables associated with the particular structure predicted by the master problem. The nature of the master problem is such that it represents an approximation to the original MINLP problem, and contains all of the alternatives embedded in the superstructure.

The basic steps involved in the proposed algorithm are shown in Figure 2, and are as follows. An initial system configuration is specified by assigning appropriate fixed values to the binary variables associated with units in the superstructure. This configuration is then analyzed by optimizing its continuous parameters. Since the MINLP program P2 is the model of the superstructure, fixing the binary variables in P2 renders the nonlinear programming (NLP) subproblem associated with a particular configuration. Because every system configuration is a restriction of the superstructure, the optimal objective function value of the associated NLP problem provides a valid upper bound on the cost of the optimal solution to program P2.

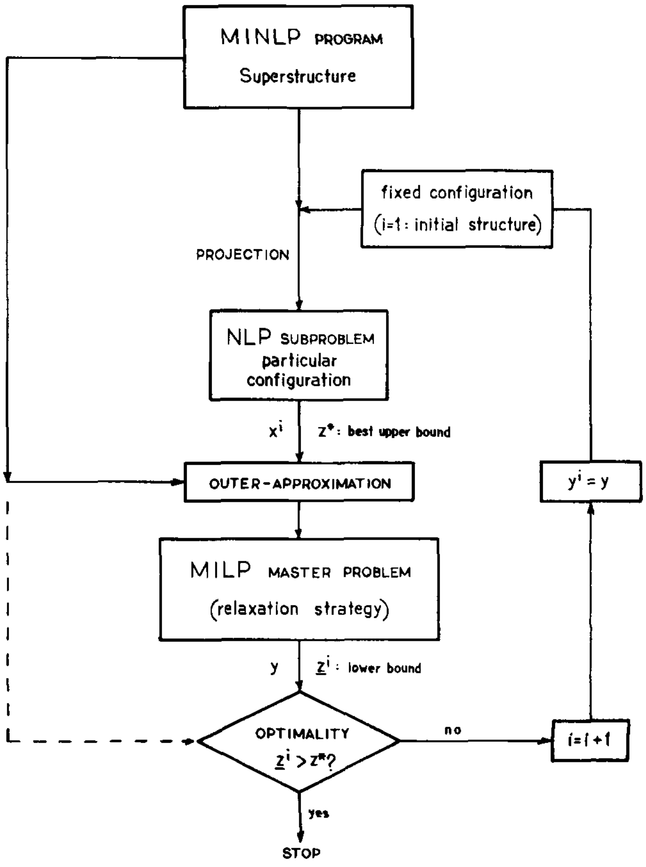

Figure 2. MINLP algorithm for system synthesis.

The solution of the corresponding NLP subproblem is then used as a base point to construct a master program, which is the representation of an approximation to the original synthesis problem. Outer approximations at that continuous point, which correspond to a linearization of the continuous part of the superstructure, are used to construct the master program. Since outer approximations will be selected so as to overestimate the continuous feasible region of the superstructure and to underestimate its objective function, the master program corresponds to a relaxation of the synthesis problem. Thus, the optimal objective function value of the master program provides a lower bound on the solution of P2. Because the approximating functions will be chosen as linear expressions in the continuous variables, the master program corresponds to a mixed-integer linear programming (MILP) problem. Additional constraints can be incorporated in this problem so as to exclude from consideration the system configuration just analyzed, and so as to require that the objective function of the master program lie below the upper bound that has been found. There are then two possibilities when solving the master program at the first iteration:

1. If no feasible solution is found, it means that there are no other system configurations with a lower bound smaller than the upper bound given by the optimal cost of the configuration just analyzed. Hence, this processing structure is the optimal solution to the synthesis problem and the search can be stopped.

2. If a feasible solution is found, the new set of values determined for the binary variables will correspond to the configuration of the processing scheme that is the optimal solution of the current approximation to the superstructure. Since the cost of the approximation is a lower bound that lies below the cost of the structure just analyzed, the new configuration is a candidate for the optimal solution of program P2. To determine whether or not the new configuration is the solution for the synthesis problem, the optimization of the continuous parameters has to be performed by solving the corresponding NLP subproblem in the second iteration. New outer approximations are determined, and from this point on the procedure is repeated as shown in Figure 2 until no feasible solution is found in the master program.

It should be pointed out that particular system configurations have different feasible spaces, and therefore in general they are independent of each other. Thus in the iterative procedure outlined above, the cost of successive structures that are generated does not follow any particular trend, and the upper bound to be considered at any iteration must be given by the lowest cost among the configurations that have been analyzed up to that point. Therefore, the optimal processing configuration will be given by the one associated with the current best upper bound when the stopping condition occurs.

In order for the master program to provide an increasingly better approximation to the original problem P2 as iterations proceed, all of the outer approximations determined for the configurations analyzed previously are considered in the master program at any given iteration. Under this strategy, the sequence of master problems is equivalent to a sequence of nested approximations of the superstructure. Since these approximations become increasingly constrained, the associated solutions of successive master problems will define a monotone nondecreasing sequence of lower bounds on the cost of the optimal configuration.

In summary, the proposed algorithm for solving the synthesis problem consists of the solution of an alternate sequence of nonlinear programming subproblems and mixed-integer linear master programs as seen in Figure 2. The former correspond to the detailed analysis of particular system configurations, while the lattter correspond to the solution of linear approximations of the superstructure that will either generate a new structure to be tested, or identify the stopping criterion in the search procedure. The main advantage with this procedure is that usually only a few iterations are required to solve mixed-integer nonlinear programming programs with the mathematical structure given by P2 (Duran and Grossmann, 1984, 1985a).

MASTER PROGRAM

Since the master problem is the major component in the proposed algorithm for solving the MINLP program P2, this section will present in a somewhat informal manner the basic ideas behind the mathematical formulation of this problem. The detailed derivation of the outer-approximation algorithm and its theoretical properties are given in Duran and Grossmann (1984, 1985a).

First, consider that a given combination y^i of binary variables is selected. The MINLP program P2 for fixed y^i then renders the following NLP subproblem for the particular system configuration associated with y^i,

$$\begin{aligned} z(y^i) = c^T y^i + \min_{x} f(x) \qquad & [S(y^i)] \\ \text{s.t. } g(x) \le -By^i & \\ x \in X & \end{aligned}$$

Assuming that the continuous optimization problem $S(y^i)$ has an optimal solution given by $[z(y^i), x^i]$, its optimal objective function value $z(y^i)$ provides a valid upper bound on the optimal objective z for program P2. Furthermore, based on the optimum value x^i for the continuous variables x, an approximation to problem P2 can be constructed as follows.

Suppose that the nonlinear functions f and those in the vector g are represented by linear outer approximations derived at the point x^i. That is, linear expressions are derived for which the following relations will hold,

$$\left.\begin{aligned} (a^i)^T x - b^i \le f(x) \\ D^i x - d^i \le g(x) \end{aligned}\right\} \quad \text{all } x \in X \qquad (11)$$

where b^i, a^i, d^i, and D^i, are respectively a scalar, vectors, and a matrix of comformable dimensions evaluated at given $x^i \in X$. If the nonlinear functions are continuously differentiable and convex, a judicious choice for the outer approximations in Eq. 11 is clearly the tangential approximation at x^i, that is,

$$\left.\begin{aligned} (a^i)^T x - b^i \equiv f(x^i) + \nabla f(x^i)^T (x - x^i) \\ D^i x - d^i \equiv g(x^i) + \nabla g(x^i)^T (x - x^i) \end{aligned}\right\} \quad \text{all } x \in X \qquad (12)$$

where $\nabla f(x^i)$ is the n-gradient vector and $\nabla g(x^i)$ the $n \times p$ Jacobian matrix evaluated at given $x^i \in X$. If the functions are nonconvex, the derivation of the outer approximations in Eq. 11 is in general a nontrivial problem. However, as discussed in Duran (1984), good linear underestimators that can be used as

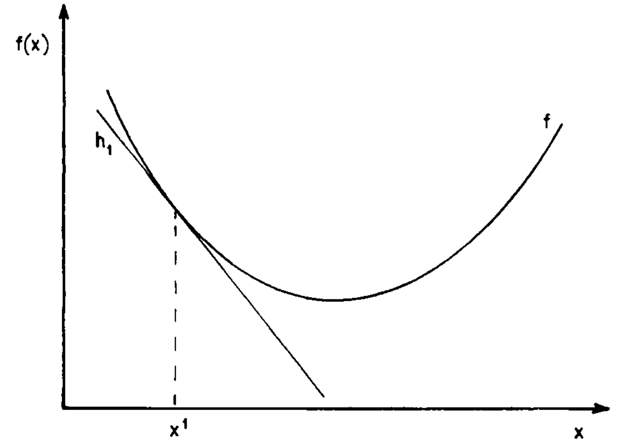

Figure 3. Outer approximation at one point for a convex function in R^1.

outer approximations can be derived for special classes of nonconvex functions, e.g., concave, bilinear.

As given by Eq. 11, the basic property that an outer approximation has to satisfy is that it must underestimate the value of the function for all points in the domain. Further, for a continuously differentiable convex function the best linear outer approximation is given by the tangential approximation at the given point, as shown in Figure 3. Because of the linearity of the binary variables, the underestimating effect carries over to the objective function in program P2 since from Eq. 11,

$$c^Ty + (a^i)^Tx - b^i \leq c^Ty + f(x) \tag{13}$$

On the other hand, the intersection of the spaces defined by the outer approximations to the functions g has the global effect of overestimating the original continuous feasible region, as illustrated in Figure 4. Hence, from Eq. 11 and due to the linearity of the binary variables it follows that,

$$D^ix - d^i + By \leq g(x) + By \leq 0 \tag{14}$$

Assuming that the outer approximations in Eq. 11 can be obtained, the following mixed-integer linear programming (MILP) formulation provides an approximation to the MINLP program P2,

$$\begin{aligned} z^i = \min_{x,y} \; & c^Ty + (a^i)^Tx - b^i \\ \text{s.t.} \; & D^ix + By \leq d^i \\ & x \in X, y \in U \end{aligned} \tag{15}$$

This problem underestimates the objective function and overestimates the feasible region of program P2. Therefore, the optimal objective function value z^i in Eq. 15 provides a lower bound on the objective function of program P2. Note that the MILP problem in Eq. 15 represents a linear approximation to the superstructure at the point x^i, which corresponds to the optimal continuous variables of the system configuration associated with y^i.

The formulation in Eq. 15 can be generalized for the case when k alternative system configurations have been analyzed through the solution of the corresponding NLP subproblems $S(y^i)$. That is, if at the kth iteration of the algorithm one considers the outer approximations for the functions f and g at the optimal points $x^i : i = 1, \ldots, k$, the MILP problem in terms of these k sets of outer approximations can then be formulated as,

$$\begin{aligned} z^k = \min_{x,\mu,y} \; & c^Ty + \mu \\ \text{s.t.} \; & \left.\begin{aligned} (a^i)^Tx - \mu \leq b^i \\ D^ix + By \leq d^i \end{aligned}\right\} \quad i = 1, \ldots k \\ & x \in X, \mu \in R^1, y \in U \end{aligned} \tag{16}$$

where the outer approximations for the objective function have been written as a set of k inequality constraints with the upper bound scalar variable μ. This is the standard manipulation for handling the pointwise maximum of a set of functions, which in this case correspond to the objective function outer approximations. This ensures that for every point in the domain the largest underestimation available will be selected for the objective function, Figure 5.

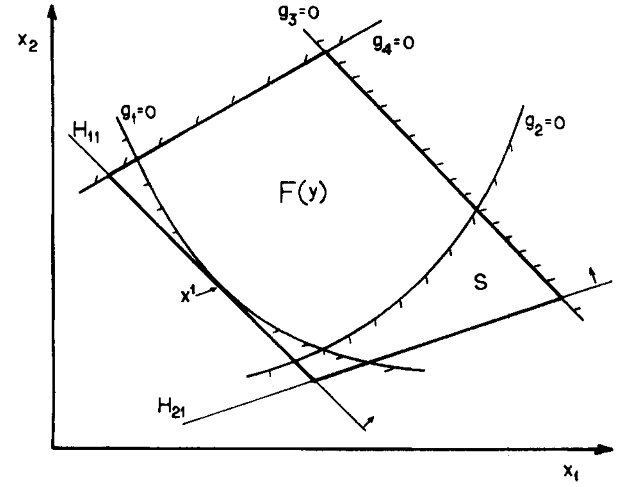

Figure 4. Outer approximation at one point for a convex set in R^2.

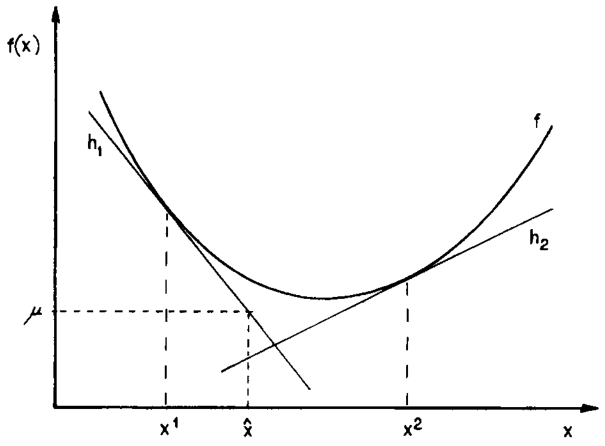

Figure 5. Underestimation of a convex function in R^1.

As shown in Figures 5 and 6, as an increasing number of outer approximations is considered in problem 16, an increasingly tighter representation for both the original objective function and the feasible space is obtained. Because of this fact, as iterations proceed in the algorithm, the optimal objective function values z^k of the corresponding master problems in 16 determine a monotone nondecreasing sequence of lower bounds on the optimal objective z of the MINLP program P2; that is,

$$z^1 \leq \cdots \leq z^{k-1} \leq z^k \leq z^{k+1} \leq \cdots \leq z \qquad (17)$$

If integer cuts are included in problem 16, so as to make infeasible the selection of the binary combinations associated with the k system configurations that have been analyzed up to that point, it is clear that the bounding property in Eq. 17 will still hold. In the context of the proposed algorithm in Figure 2, the monotonicity of the sequence of lower bounds $z^i: i = 1, \ldots, k$, can then be used to identify the stopping condition for the procedure.

It should be recalled that the current best upper bound z^* is selected as the smallest optimal cost among the k configurations that have been analyzed, i.e., $z^* = \min \{z(y^i): i = 1, \ldots, k\}$. Therefore, if the inequality $z^k > z^*$ holds when solving the master problem in 16 at a given iteration k, this indicates that there is no other configuration in the superstructure, different from the previously analyzed structures, that can have a smaller optimal cost than the current upper bound. Hence, this implies that the search can be stopped whenever the condition $z^k > z^*$ is satisfied.

The stopping criterion in the proposed algorithm can then be detected if a violation to the constraint $z^k \leq z^*$ occurs. From problem 16 this constraint can also be expressed as $c^T y + \mu \leq z^*$. Incorporating this constraint in problem 16 together with integer cuts for excluding the k configurations that have been analyzed, the master program at a given iteration k can be written in final form as the following MILP problem,

$$\begin{aligned} z^k = \min_{x, \mu, y} \quad & c^T y + \mu \\ \text{s.t.} \quad & \left.\begin{aligned} (a^i)^T x - \mu &\leq b^i \\ D^i x + By &\leq d^i \end{aligned}\right\} \quad i = 1, \ldots k \qquad [\mathrm{M}^k] \\ & z^{k-1} \leq c^T y + \mu \leq z^* \\ & y \in \{\text{integer cuts}\} \\ & x \in X, \mu \in R^1, y \in U \end{aligned}$$

Although according to Eq. 17 the constraint $z^{k-1} \leq c^T y + \mu$ is redundant, it has been introduced because it may expedite the enumeration procedure when solving the MILP problem. The integer cuts to eliminate previously analyzed configurations are linear inequalities, and their exact definition is given in the Appendix.

For the kth iteration, problem M^k then provides the required master program in the algorithm outlined in Figure 2. If this program has an optimal solution, an improvement in the lower bound will then be obtained, and a binary combination will be determined for a new system structure that has the potential of becoming the optimal configuration. If on the other hand, problem M^k has no feasible mixed-integer solution, this will imply that the constraint $z^k \leq z^*$ has been violated, which indicates that the search procedure can be stopped. The optimal solution $[z, x^*, y^*]$ of the MINLP program P2 will then be given by the current best upper bound $z = z^*$ obtained from the optimal solution x^* of the y^*-parameterized subproblem $S(y^*)$.

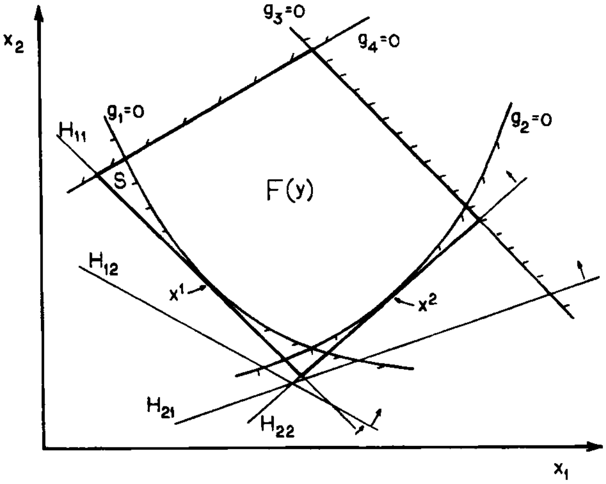

Figure 6. Overestimation of a convex set in R^2.

Algorithm

Having presented the basic ideas behind the outer-approximation algorithm, the main steps shown in Figure 2 can now be stated. For presentation purposes it is assumed that every NLP subproblem has a finite optimal solution. The algorithm is then as follows.

Step 1. Set lower bound $z^0 = -\infty$, upper bound $z^* = +\infty$, iteration $k = 1$.

Select a binary combination $y^1 \in U$.

Step 2. Solve the y^k-parameterized NLP subproblem $S(y^k)$.

$S(y^k)$ has the optimal solution $[x^k, z(y^k)]$, where $z(y^k)$ is a valid upper bound on the optimal value z of the MINLP program P2.

Update the current upper bound estimate: $z^* = \min \{z^*, z(y^k)\}$.

If $z^* = z(y^k)$ set $y^* = y^k$, $x^* = x^k$.

At the point x^k, derive the outer approximations in Eq. 11.

Derive integer cut (see Appendix A) to eliminate y^k from further consideration.

Step 3. Using the sets of outer approximations that have been determined up to the current iteration k, construct and solve the corresponding MILP master program M^k.

One of the following cases must occur:

(I). Problem M^k does not have a mixed-integer feasible solution. STOP.

The optimal solution to the MINLP program P2 is given by the current upper bound z^* and the variable vectors (x^*, y^*). That corresponds to the optimal solution of the y^*-parameterized NLP subproblem as defined in step 2.

(II). Problem M^k has a finite optimal solution (z^k, x, y); z^k is an element in the monotonic sequence of lower bounds on the optimal value z of the MINLP program P2; y is a new integer combination to be tested in the algorithm.

Derive integer cuts (see Appendix A) to eliminate binary combinations that were found to be infeasible in the solution of problem M^k.

Set $y^{k+1} = y$, and $k = k + 1$ to indicate a new iteration. Return to step 2.

In Duran and Grossmann (1985a), it has been shown that this algorithm converges in a finite number of iterations to the optimal solution of program P2. For the case when the nonlinear functions are convex, the algorithm is guaranteed to converge to the global optimum solution. This follows from the fact that for convex functions the nonlinear programming subproblems have a unique solution, and that the bounding properties for the master problem will rigorously hold by using the linearizations in Eq. 12.

In the case when some of the nonlinear functions are nonconvex, there are two possibilities in applying this algorithm. If the nonlinear functions are of special structure, variable transformations may be used to obtain convex functions, or else valid linear underestimators may be used to ensure that the bounding properties of the master problem will hold. This will in general guarantee a global optimum solution. For nonlinear functions with unknown structure the algorithm can simply be applied directly with the linearizations in Eq. 12. In this case, however, there is no rigorous guarantee that the global optimal solution will be obtained since the NLP subproblems may exhibit local solutions, and/or the bounding properties of the master problem may not be preserved.

It should also be noted that it has been proved in Duran and Grossmann (1984), that when convex functions are involved in the class of problems P2, this algorithm will converge in fewer iterations than the generalized Benders decomposition method of Geoffrion (1972). Qualitatively, this is due to the good approximations to P2 that are provided by the master problem, which will tend to predict very quickly tight bounds for the optimal solution.

In some applications there is the possibility that in step 2 the NLP subproblem may have no feasible solution for the selected binary variable y^k. When this case arises, step 2 in the above algorithm can simply be modified by setting $z(y^k) = +\infty$, and using the infeasible point x^k found for that particular NLP subproblem as the point for deriving the corresponding outer approximation. The theoretical justification for this procedure can be found in Duran and Grossmann (1985a).

It is also worth noting that the number of constraints in the master problem M^k in step 3 will increase as iterations proceed, due to the successive addition of outer approximations. However, it is clear that those individual constraints that are linear are exactly equivalent to their corresponding outer approximations. Hence, linear constraints do not require the successive addition of approximations in the master problem. Therefore, the increase of the size of the master problem in successive iterations is dependent only on the actual number of nonlinear constraints present in problem P2.

As a final remark, it must be pointed out that if linear equality constraints are present in the MINLP formulation, these can be handled directly by the algorithm and do not require elimination. This follows from the fact that linear functions have an exact representation in the master problem, and therefore by including them into this problem they will not contribute to the gap of the predicted lower bound.

Example Problem

The test problem selected for this paper is the optimal synthesis of gas transmission networks. For this problem it will be assumed that the two-dimensional location of a set of gas wells and demand sites is given with their corresponding specifications for pressure and gas flow.

In general there are many possible transportation paths for interconnecting the gas wells with the demand sites. In this paper it will be assumed that the basic form of the transportation paths is postulated with a maximum number of potential branches in the pipeline, each having specified gas flow rates, e.g., Figure 7a. Furthermore, the two-dimensional location for the mixing and splitting points in the pipeline will be treated as continuous variables to be selected. In this way, different selections for these mixing and splitting points within the proposed form of the transportation paths can give rise to different pipeline structures as illustrated in Figure 7. Note in this figure that the length of one of the branches is reduced to zero for the alternatives in Figures 7b–d. Hence, if a maximum number of potential compressors is postulated in each branch of the proposed transportation paths (Figure 7a), a superstructure representation can be obtained for the gas pipeline problem. It should be noted, however, that this superstructure is restricted to having fixed flows in each of the potential branches.

Given then the superstructure described above, and specifica-

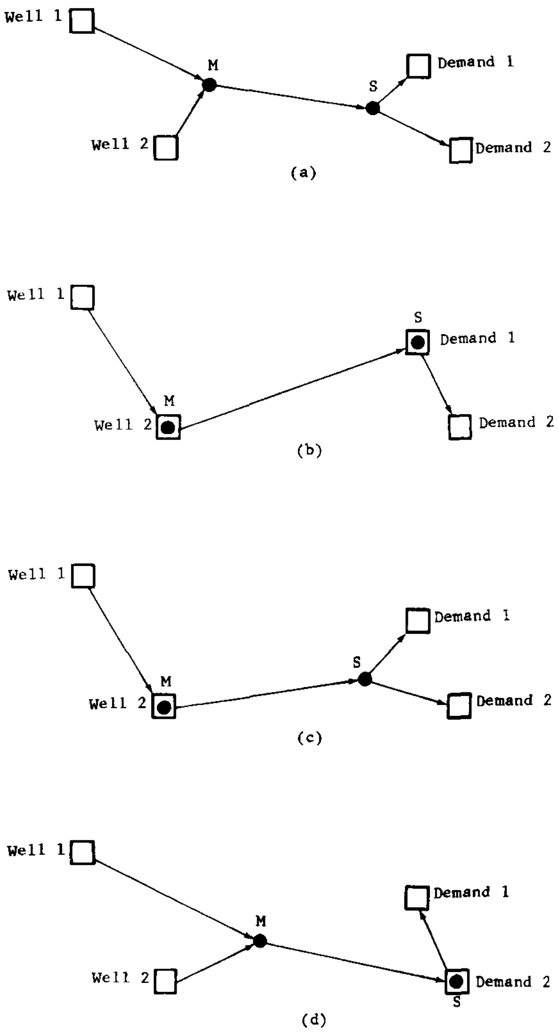

Figure 7. Alternative transportation paths for a gas pipeline.

tions for the gas wells and demand sites, the optimal synthesis of the gas pipeline considered in this paper involves the selection of the following items so as to minimize the total annual cost of the system:

a. Number and location of gas compressors
b. Two-dimensional location of mixing and splitting points
c. Length and diameter for the different pipe segments
d. Power of compressors
e. Pressure profile of the pipeline

Note that items a and b define the configuration of the pipeline, while items c–e define sizes and operating parameters. In this paper the decision on the number of compressors will be modeled with binary variables, while all the other decisions will be modeled with continuous variables. It is clear that pipe diameters in practice are only available in discrete sizes. However, in order to simplify the formulation, pipe diameters will be treated as continuous variables.

The above problem has been addressed previously by different authors with somewhat different assumptions and using other solution procedures. In the work by Bickel et al. (1978) the lengths of the branches along each path were specified without accounting for coordinates in the location of gas wellheads and demand points. Therefore, the optimal location of branch points was not considered. These authors tackled this problem as both a nonlinear optimization problem and as a mixed-integer program in which integer variables were related to the number of compressor stations. The latter formulation was solved with a special branch-and-bound enumeration scheme. Since the nonlinear functions in these formulations are nonconvex, globality of the solution and bounding properties in the branch and bound solution method may not always hold. Dunn (1980) extended the work of Bickel et al by specifying the two-dimensional location of gas wells and demand points, and considering the optimal location of branch points. Models were derived for the case of one gas well and two demands, and for the case of two gas wells and two demands. This author solved these models with two nonlinear programming formulations. In the first, linear costs were assumed for the compressors; in the second, fixed cost charges were included. For the latter formulation continuous approximations in terms of compression ratios were used to replace the binary variables. Although computer times with the latter approach were substantially lower than with the branch-and-bound procedure of Bickel et al., different solutions were found by using different starting points. Finally, Soliman (1982) has solved a restricted version of the gas pipeline problem in which the lengths of all pipe segments were fixed. In this work the problem was formulated as a nonlinear programming program.

It should be pointed out that the main reason a purely nonlinear programming model may not represent the gas transmission network problem adequately is the effect of economies of scale. If strictly linear cost functions are used, then a gradient-based technique cannot guarantee to find a realistic solution. That is, in this case since the effect of economies of scale is disregarded, the resulting nonlinear optimization program will have the tendency to select many compressors that in addition may have very small compression ratios. On the other hand, if economies of scale are modeled with the use of nonlinear concave cost functions, or with continuous approximations for the fixed-cost charges, the resulting nonlinear program will be nonconvex. Hence, multiple local solutions can arise, possibly for each different configuration. A way of circumventing both difficulties is by assigning binary variables to the compressors to identify individual alternatives, and by including fixed-cost charges in the capital cost as indicated in Eq. 1 (see also Figure 1). This then leads to an underlying MINLP model for the gas pipeline problem.

A generalized network formulation for the definition of the gas pipeline problem addressed in this paper is given in Appendix B. This formulation is applicable to any number of gas wells and demand points for which a given superstructure is specified. The MINLP formulation presented in Appendix B has the characteristic that the binary variables appear linearly and that the nonlinear functions in the objective function and constraints are convex. Also, since the nonlinear equations can be eliminated by variable substitution, the nonlinear constraints are inequalities. Thus, the formulation belongs to the general class defined by program P2. Note that due to the convexity of the nonlinear functions the proposed algorithm will actually determine the global optimum solution for this problem. Furthermore, due to the convexity and differentiability of the functions the outer approximations in Eq. 11 can be obtained by the function linearizations indicated in Eq. 12.

The example problem selected to illustrate the performance of the outer-approximation algorithm, and the application of the proposed MINLP formulation for the gas pipeline problem is given in Figure 8. In this superstructure the circles represent potential compressors, while the pairs (x_j, z_j) are variable coordinates that define their location. The data for the example in this paper are given in Table 1. Each compressor, when present in a particular pipeline configuration, is assumed to lose 0.5% of the gas transmitted, and has an upper bound of 7,457 kW, which is the maximum power that a single-stage centrifugal compressor can handle (Bickel et al., 1978). The associated MINLP program for this particular instance can readily be derived based on the general model given in Appendix B. The particular formulation involves 10 binary variables, 51 continuous variables (41 acting nonlinearly), and 77 constraints partitioned as 10 nonlinear inequalities, 42 mixed-linear, and 25 linear.

With respect to the actual implementation of the proposed outer-approximation algorithm, the NLP subproblems were solved on a DEC-20 computer system with the computer code MINOS/AUGMENTED (Murtagh and Saunders, 1980), while the MILP master problems were solved with LINDO (Schrage, 1981). It should be noted that the problem was solved in English units.

The starting point selected was the binary combination $y = \{y_j : j = 1, \ldots, 10\} = (0, 1, 0, 0, 1, 0, 0, 0, 0, 0)$, which corresponds to the configuration shown in Figure 9. According to the gas pressure conditions at the well and demand points, this seemed to be a good initial configuration since it involved the existence of compressors in the well branch and in the high-pressure branch. The optimal continuous parameters for this configuration are reported in Table 2. A minimum total cost of $8,586,756/yr was found, and the solution of the associated NLP problem required 23.97 s of CPU time.

With this starting point for the first iteration, the proposed outer-approximation algorithm solved the pipeline problem given in Figure 8 and Table 1 in only five iterations and a total of 178.18 s of CPU time. Of this total time, 135.20 s were used for the solution of the five NLP subproblems. The actual optimal configuration, $y = \{y_j : j = 1, \ldots, 10\} = (1, 1, 1, 0, 0, 0,$

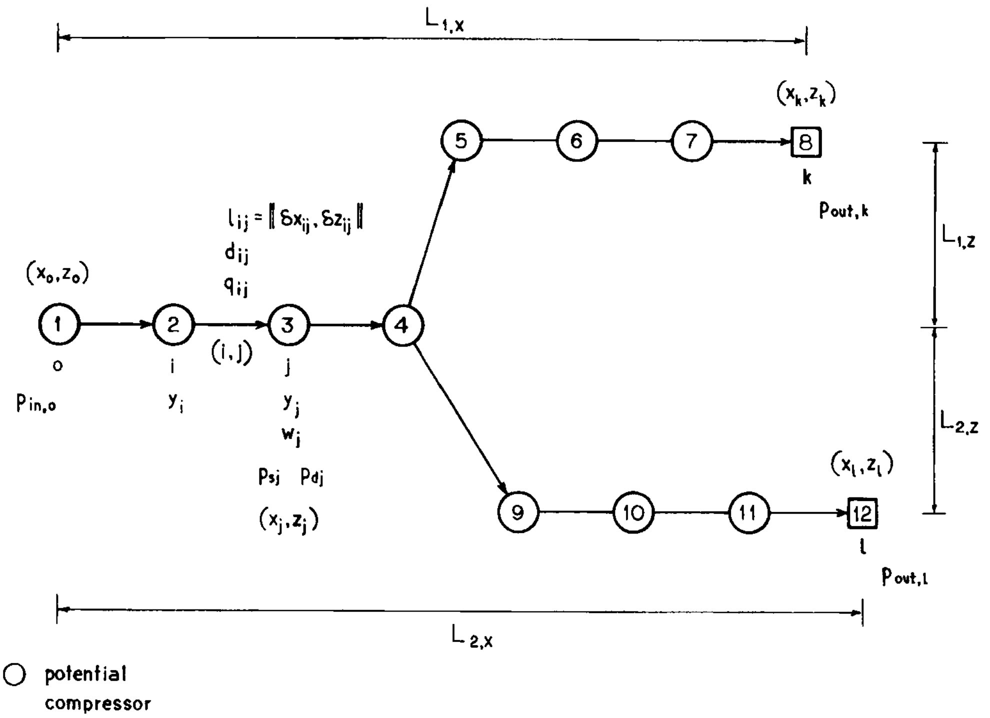

Figure 8. Gas transmission network: superstructure for example problem.

0, 0, 0, 0), was found by the master problem at the second iteration, while the optimal continuous parameters for this configuration were determined by the NLP subproblem at the third iteration. This configuration, which is shown in Figure 10, represents the global minimum, and has a total cost of $7,837,827/yr. Its detailed results are presented in Table 3. The progress of the outer-approximation algorithm as given by the sequence of lower bounds predicted by the master program, and the cost of the five alternative configurations that were analyzed is reported in Figure 11. This figure clearly shows that the master problem provides very quickly good lower bounds, and consequently good approximations to the original problem. The condition to terminate the search for candidate configurations is shown at iteration 5 of this figure.

Table 1. Data for Example Problem, Figure 8

Cost Coefficients	Constant Parameters
C_f = \$10,000/yr	$a = 3/16, z = 1, \eta = 1$
C_o = \$103.93/kW · yr	$\gamma = 1.26, s_g = 0.76$
C_p = \$21,283.77/km · m · yr	p_o = 0.1013 MPa, T_o = 288.9 K T = 288.9 K
Capacity Bounds	**Problem Specifications**
U_c^U = 7,457 kW	$(x_1, z_1) = (0,0)$ $(x_8, z_8) = (281.63,80.46)$
U_x^U = 281.63 km	$(x_{12}, z_{12}) = (321.86,0)$
U_x^L = −281.63 km	$p_{in,1}$ = 3.445 MPa
U_z^U = 80.46 km	$p_{out,8}$ = 4.137 MPa, $p_{out,12}$ = 2.068 MPa
U_z^L = −80.46 km	$q_{01} = q_1$ = 16.992 MMm^3/d $q_{45} = q_{49} = (q_{34})/2$

Graph $G = (V, E)$ Index Sets

$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
$E = \{(1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (4,9), (9,10), (10,11), (11,12)\}$
$S_1 = \{(1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8)\}$
$S_2 = \{(1,2), (2,3), (3,4), (4,9), (9,10), (10,11), (11,12)\}$
$V_W = \{1\}$, $V_I = \{2, 3, 4, 5, 6, 7, 9, 10, 11\}$, $V_D = \{8,12\}$, $V_B = \{4\}$
$V_d(1) = \{2\}$, $V_u(1) = 0$
$V_d(2) = \{3\}$, $V_u(2) = \{1\}$
etc.

Analysis of the results reported in Tables 2 and 3 shows that with respect to the initial configuration, the parameters defining the optimal transportation paths, Figures 9 and 10, did not change substantially. However, the major element for the reduction in the cost of the gas pipeline network was the selection of smaller pipe diameters, which was possible because of the selection of three compressors. Note that this actually led to total power requirements that are approximately twice as large as in the initial configuration.

An interesting feature in the solution for the optimal configuration in Figure 10, was the fact that the pipeline segments between selected compressors had zero length (see Table 3). The interpretation of this result is that compressors with zero pipe segment length between them can actually be regarded as just

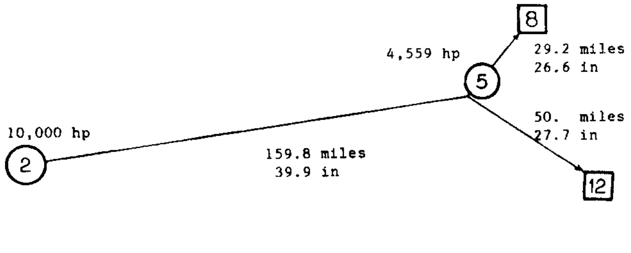

Figure 9. Gas pipeline: initial configuration.

Table 2. Results for Initial Configuration, Figure 9

(i,j)*	$p_{\text{in},i}$ MPa	$p_{\text{out},j}$ MPa	q_{ij} MMm^3/day	w_j kW	ℓ_{ij} km	d_{ij} m
1, 1	3.447	3.447	16.992	0.0	—	—
1, 2	3.447	3.447	16.992	—	0.000	0.000
2, 2	3.447	4.950	16.992	7457.0	—	—
2, 3	4.950	3.382	16.907	—	257.189	1.015
3, 3	3.382	3.382	16.907	0.0	—	—
3, 4	3.382	3.382	16.907	—	0.000	0.000
4, 4	3.382	3.382	16.907	0.0	—	—
4, 5	3.382	3.382	8.453	—	0.000	0.000
5, 5	3.382	4.716	8.453	3399.5	—	—
5, 6	4.716	4.137	8.411	—	46.952	0.677
6, 6	4.137	4.137	8.411	0.0	—	—
6, 7	4.137	4.137	8.411	—	0.000	0.000
7, 7	4.137	4.137	8.411	0.0	—	—
7, 8	4.137	4.137	8.411	—	0.000	0.000
4, 9	3.382	2.068	8.453	—	80.543	0.705
9, 9	2.068	2.068	8.453	0.0	—	—
9,10	2.068	2.068	8.453	—	0.000	0.000
10,10	2.068	2.068	8.453	0.0	—	—
10,11	2.068	2.068	8.453	—	0.000	0.000
11,11	2.068	2.068	8.453	0.0	—	—
11,12	2.068	2.068	8.453	—	0.000	0.000

(ij)	1–2	2–3	3–4	4–5	5–6	6–7	7–8	4–9	9–10	10–11	11–12
δx_{ij}	0.0	253.6	0.0	0.0	28.0	0.0	0.0	68.2	0.0	0.0	0.0
δz_{ij}	0.0	42.8	0.0	0.0	37.7	0.0	0.0	−42.8	0.0	0.0	0.0

Total Cost: $8,586,755.96/yr

*$i = j$: compressor ⟨node j⟩.
$i \neq j$: pipeline segment ⟨arc ij⟩.

Table 3. Results for Optimal Configuration, Figure 10

(i,j)*	$p_{\text{in},i}$ MPa	$p_{\text{out},j}$ MPa	q_{ij} MMm^3/day	w_j kW	ℓ_{ij} km	d_{ij} m
1, 1	3.447	4.950	16.992	7457.0	—	—
1, 2	4.950	4.950	16.907	—	0.000	0.000
2, 2	4.950	7.121	16.907	7457.0	—	—
2, 3	7.121	7.121	16.823	—	0.000	0.000
3, 3	7.121	10.262	16.823	7457.0	—	—
3, 4	10.262	5.831	16.738	—	262.398	0.738
4, 4	5.831	5.831	16.738	0.0	—	—
4, 5	5.831	4.137	8.369	—	41.898	0.529
5, 5	4.137	4.137	8.369	0.0	—	—
5, 6	4.137	4.137	8.369	—	0.000	0.000
6, 6	4.137	4.137	8.369	0.0	—	—
6, 7	4.137	4.137	8.369	—	0.000	0.000
7, 7	4.137	4.137	8.369	0.0	—	—
7, 8	4.137	4.137	8.369	—	0.000	0.000
4, 9	5.831	2.068	8.369	—	78.128	0.535
9, 9	2.068	2.068	8.369	0.0	—	—
9,10	2.068	2.068	8.369	—	0.000	0.000
10,10	2.068	2.068	8.369	0.0	—	—
10,11	2.068	2.068	8.369	—	0.000	0.000
11,11	2.068	2.068	8.369	0.0	—	—
11,12	2.068	2.068	8.369	—	0.000	0.000

(ij)	1–2	2–3	3–4	4–5	5–6	6–7	7–8	4–9	9–10	10–11	11–12
δx_{ij}	0.0	0.0	258.4	23.2	0.0	0.0	0.0	63.4	0.0	0.0	0.0
δz_{ij}	0.0	0.0	45.5	34.9	0.0	0.0	0.0	−45.5	0.0	0.0	0.0

Total Cost: $7,837,827.20/yr

*$i = j$: compressor ⟨node j⟩.
$i \neq j$: pipeline segment ⟨arc ij⟩.

one multistage compressor: a compressor with as many stages as compressors involved in a pipeline segment of zero length. For the present problem, the solution is a three-stage compressor as seen in Figure 10. Notice in Table 3 that the compression ratios for the individual compressors are the same and equal to 1.44, which falls within practical limits. This result agrees with the fact that work minimization is obtained when compression ratios are equal in each compression stage. The significance of the results of this example is then that they show that the proposed MINLP framework can actually accommodate a rather broad spectrum of gas pipeline alternative configurations from which realistic solutions can be obtained.

It should also be pointed out that according to the MINLP formulation presented in Appendix B, the minimum number of possible different configurations for the gas pipeline example problem, as determined by the combinations of number of possible compressors (five for wellhead branch, four for each of the two other branches), is 80; the maximum number of possible configurations, as determined by all individual compressor combinations, is 1,024 (2^{10}). The proposed algorithm can therefore be regarded as an efficient solution procedure since only five different pipeline configurations had to be analyzed in order to find the optimal solution. This is particularly relevant since the bottleneck in the computations is usually the detailed analysis of the configurations with the NLP subproblems, as was the case in this example problem.

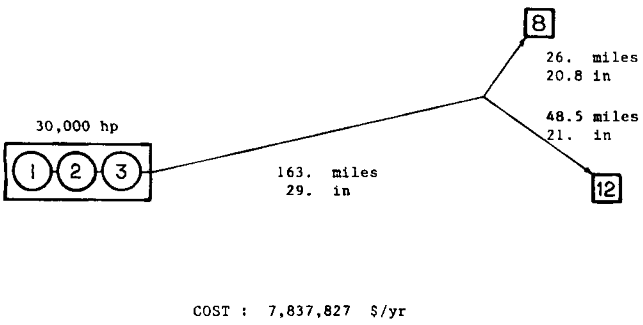

Figure 10. Optimal gas pipeline configuration for synthesis problem.

Finally, in order to test the robustness of the outer approximation algorithm, the problem was solved with two additional starting points: $y = (1, 1, 0, 1, 0, 0, 0, 1, 0, 0)$, and $y = (1, 1, 1, 0, 0, 0, 0, 1, 0, 0)$. The optimal design of the first configuration has a cost of $8,010,419/yr; the second configuration has a cost of $7,857,890/yr. As expected, both starting points converged to the global optimum solution ($7,837,827/yr). The former required only three iterations and 95.92 s of CPU time (79.69 for NLP subproblems), while the latter converged in four iterations and 141.37 s (108.26 for NLP subproblems). Thus, the efficiency of the algorithm in terms of number of required iterations would suggest that the proposed procedure is a promising tool for tacking the optimal design of gas transmission networks problem in its general form.

Discussion

The outer approximation algorithm outlined in this paper was the result of a recent study by the authors (Duran and Gross-

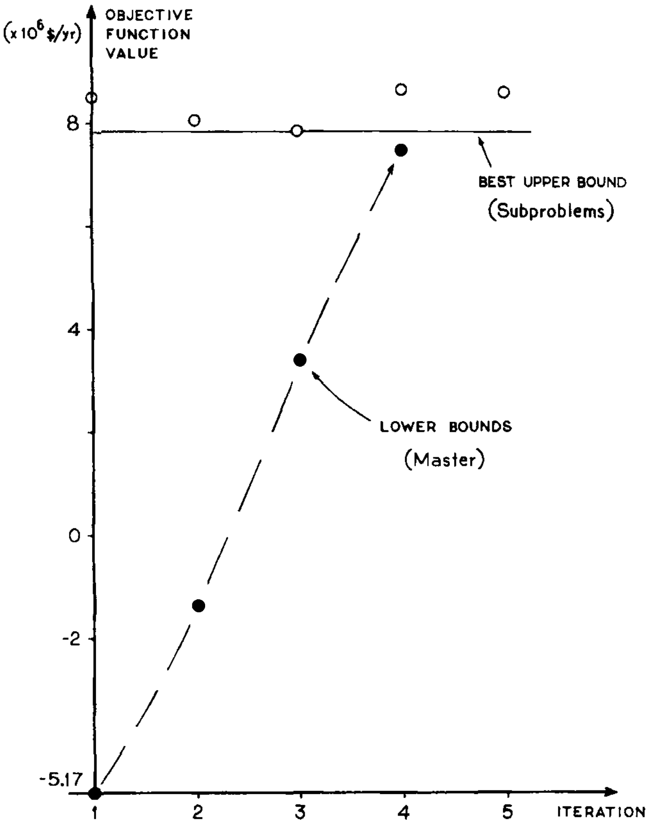

Figure 11. Example: progress of the outer-approximation algorithm.

mann, 1984, 1985a) to solve MINLP problems with the particular mathematical structure given by program P2. The motivation for studying this problem arose from the need to develop an efficient MINLP algorithm for process synthesis that can handle explicitly the nonlinearities involved in the modeling of processing systems. Therefore, the significance of the algorithm proposed in this paper is that it can avoid not only the discretization of nonlinear continuous variables (e.g., pressures, temperatures, reactor conversions, purge rates) as in the case of the MILP approach by Papoulias and Grossmann (1983a, b, c), but it can also reduce the number of binary variables needed to model synthesis problems with mixed-integer programming formulations. Therefore, the work that has been presented here allows the direct solution of synthesis problems that are formulated as MINLP programs.

As for the scope and limitations of the outer-approximation algorithm to large process synthesis problems, the following observations can be made. First, the algorithm requires the elimination of the nonlinear equations so as to render a problem in terms of inequalities. In general this can be a nontrivial problem within an equation-oriented environment. However, if the algorithm is applied within a sequential process simulator, this problem is largely removed since in this type of simulator the system of equations has already been eliminated, so that the optimization problem is given mainly in terms of inequalities. Second, in most large process synthesis applications it is not possible to determine whether the nonlinear functions that are involved are convex. This implies that under these circumstances there is no rigorous guarantee that the solution obtained will indeed correspond to the global optimum. Nevertheless, the algorithm will still produce a local optimum solution in which the interactions of both the structure and continuous variables are accounted for.

It should also be pointed out that the general formulation defined by program P2 is the natural model for problems in several other research areas (Duran and Grossmann, 1985a). Among the applications that have been solved using the proposed algorithm are the marketing problem of determining the optimal positioning of a new product in a multiattribute space, and small process synthesis problems where only optimal flowsheet configurations and stream flow rates were considered. Recently, Vaselenak (1985) has applied the outer-approximation algorithm in the optimal retrofit design of multiproduct batch processes. The performance of the outer-approximation algorithm has been quite satisfactory in all of the above-cited examples since few iterations were required to find the optimal solution (maximum of six). This was due mainly to the good quality of the lower bounds predicted by the master problem.

The ultimate objective of this work, however, is to be able to perform the systematic synthesis of integrated processing systems using the basic ideas behind the MINLP approach presented in this paper. As steps toward this goal, the main tools have already been developed. These tools are the outer-approximation algorithm presented in this paper, and a method for the simultaneous optimization and heat integration in process flowsheets. The latter method (Duran and Grossmann, 1986) has the capability of handling variable flow rates and temperatures of the process streams for nonlinear optimization. Preliminary experience with the application of this approach to a large integrated process example reported by Duran (1984) indicates that efficient schemes and computer implementations are required for the numerical elimination of large systems of nonlinear equations (see Eq. 9). Results on the application of this algorithm to this example will be published in a forthcoming paper.

Acknowledgment

This work was supported by the National Science Foundation under Grant No. CPE-8351237. Partial support from CONACYT and Universidad Autonoma Metropolitana-I, Mexico, is also gratefully acknowledged.

Notation

a^i = vector of variable coefficients for objective linearization
b^i = constant coefficient for objective linearization
B = matrix of coefficients for binary variables
c = vector of fixed-cost charges
C = cost function
d^i = vector of constant coefficients for constraint linearization
D^i = matrix of variable coefficients for constraint linearization
f = nonlinear term in objective function
g = vector of nonlinear functions for inequalities
p = nonlinear cost function
r = vector of nonlinear equalities
s = vector of continuous variable functions for inequalities
u_i = upper bound for variable w_i
U = set of pure-integer constraints
v = vector of state variables
w = vector of continuous variables
W = compact set for continuous variables
x = vector of continuous decision variables
X = compact set for continuous decision variables
y = vector of 0–1 variables

z = objective function of MINLP problem
z = lower bound of the objective function
μ = scalar variable for highest underestimate of nonlinear objective

Appendix A: Integer Cuts

Integer cuts have been derived to help in reducing the enumeration effort in the search for promising system configurations. There are at least two aspects in the proposed solution procedure where integer cuts can be used to enhance the efficiency of the search procedure. The first is related to the elimination of previously analyzed system structures (step 2 of the algorithm), while the second is the efficient solution of the sequence of MILP master problems. For the latter purpose, integer cuts can be used to express the infeasibility condition of certain binary combinations [step 3(II) of the algorithm]. That is, if a binary combination is found to be infeasible when solving the master problem in a given iteration, it will also be infeasible in subsequent iterations. Since in general an enumeration scheme (e.g., branch-and-bound) has to be used to solve each instance of the master program, in the associated search tree infeasible binary combinations can be identified as given by nodes where the bounding subproblem either does not have a solution or else has an objective value greater than the current best upper bound in the proposed algorithm. The elimination of particular binary combinations associated with the two aspects described above can be accomplished with the integer cut (pure-integer constraint) (Duran, 1984), $\Sigma_{j\in Q^i} y_j - \Sigma_{j\in NQ^i} y_j \le |Q^i| - 1$, where for the binary combination $y^i = \{y_j^i\}$ to be eliminated the index sets are given by $Q^i = \{j: y_j^i = 1\}$, $NQ^i = \{j: y_j^i = 0\}$.

Appendix B: MINLP Model for the Gas Transmission Network Problem

In order to determine the actual number of compressor stations required in the postulated transportation path for the gas pipeline, a maximum number of potential stations per branch in the network can be embedded within a simple superstructure, as in Figure 8. A binary (0–1) variable y_j can then be associated with each potential station to denote its existence or nonexistence in a particular pipeline configuration. Without loss of generality, it is assumed that in the postulated superstructure the potential compressors per branch are allocated such that a candidate compressor is assigned to every mixing or splitting point, to every gas wellhead, and to intermediate points of each branch. Also, no compressor is located at any gas demand point, Figure 8.

If nodes $j \in V = \{j: j = 1, \ldots, N\}$ are associated to potential compressors and gas supply sites, and arcs $(i, j) \in E$ to pipe segments, the natural representation of the gas pipeline problem is that of a directed graph $G = (V, E)$ (Figure 8). The underlying model for this graph is a generalized network formulation that corresponds to a MINLP program as given next. Each compressor j (nodes) has the following continuous variables associated with it: horsepower (w_j), suction ($p_{s,j}$) and discharge ($p_{d,j}$) pressures, and coordinates (x_j, z_j) for determining its location. Gas flow rates, pressure specifications, and coordinates are given for both production and demand nodes. The variables characterizing each pipeline segment (i, j), (arcs), are length (ℓ_{ij}), gas flow rate (q_{ij}), and internal diameter (d_{ij}). The transportation paths k connecting gas well-demand point pairs are assumed to be horizontal.

As a standard manipulation for network structures, the nodes $j \in V$ in the graph can be partitioned into sets of source (gas well) nodes V_W, sink (demand site) nodes V_D, and intermediate nodes V_I. The set of mixing/splitting nodes V_B, which is a subset of V_I, are also identified. Further, two index sets are associated with each node $i \in V$ in the network, namely, the set $V_d(i) = \{j: (i, j) \in E\}$ of immediate downstream nodes (flow from i to j), and the set $V_u(i) = \{j: (j, i) \in E\}$ of immediate upstream nodes (flow from j to i).

The objective function to be minimized is selected as the annual operating and maintenance costs of the compressors plus the sum of the annualized capital costs of the pipe and compressors. The binary variables y_j, which denote existence/nonexistence of compressor stations, are used for a fixed-cost charge formulation of the investment cost for the compressors. This yields the following objective function and related capacity constraints,

$$C_f \sum_{j\in V_W \cup V_I} y_j + C_o \sum_{j\in V_W \cup V_I} w_j + C_p \sum_{(i,j)\in E} \ell_{ij}\, d_{ij} \qquad \text{(B1)}$$

$$w_j - U_c^U y_j \le 0 \quad \text{all } j \in V_W \cup V_I,\ y_j \in \{0, 1\} \qquad \text{(B2)}$$

C_f, C_o, and C_p are, respectively, the fixed-charge and unit costs. U_c^U is a known upper bound on the size of the compressors. The performance relationship for pipeline segments $(i, j) \in E$ is given by the Weymouth (1912) flow equation,

$$d_{ij} = (\ell_{ij})^\alpha\, [p_{d,i}^2 - p_{s,j}^2]^{-\alpha}\, (B_{ij})^\alpha \quad \text{all } (i, j) \in E \qquad \text{(B3)}$$

where

$$B_{ij} = s_g\, T\, [p_o/(0.375\, T_o)]^2\, (q_{ij})^2 \qquad \text{(B4)}$$

s_g is the gas specific gravity, (p_o, T_o) are the standard conditions, T is an average gas temperature, and $\alpha = (3/16)$. For each demand node $j \in V_D$, the discharge and suction pressures are the same and equal to the specified supply pressure $p_{\text{out},j}$. The compressors are assumed to be adiabatic, and hence power requirements w_j are given by the expressions,

$$\left[\frac{p_{d,j}}{p_{s,j}}\right]^b - F_j w_j = 1 \quad \text{all } j \in V_W \cup V_I \qquad \text{(B5)}$$

where

$$F_j = [(\gamma - 1)\,\eta/(4.0426\, T\gamma)] \left[\frac{1}{q_j}\right] \qquad \text{(B6)}$$

$$b = z\,\frac{(\gamma - 1)}{\gamma} \qquad \text{(B7)}$$

$$q_j = \sum_{i\in V_u(j)} q_{ij} \qquad \text{(B8)}$$

z is the compressibility factor, γ the heat capacity ratio, η compressor efficiency, and T the gas temperature at suction conditions. The units assumed in the above equations are MPa for pressure, K for temperature, MMm^3/day for flow rates, kW for power, km for pipe length, and m for pipe diameter. Other constraints in the problem are order relations between discharge and suction pressures for each potential compressor,

$$p_{d,j} - p_{s,j} \geq 0 \quad \text{all } j \in V_I$$
$$p_{d,j} \geq p_{\text{in},j} \quad \text{all } j \in V_W \tag{B9}$$

where $p_{\text{in},j}$ are given pressure conditions at the gas wells; constraints that account for the monotonicity of the pressure profile in each pipe segment are given by

$$p_{d,j} - p_{s,i} \geq 0 \quad \text{all } j \in V_W \cup V_I, i \in \{V_d(j) \backslash V_D\}$$
$$p_{d,j} \geq p_{\text{out},i} \quad \text{all } i \in V_D, j \in V_u(i) \tag{B10}$$

The location of compressors in the network (and consequently the optimal location of splitting and mixing points), and the length of each pipeline segment can be handled by defining the following coordinate difference variables,

$$\left.\begin{array}{l} \delta x_{ij} = x_j - x_i \\ \delta z_{ij} = z_j - z_i \end{array}\right\} \text{all } (i,j) \in E \tag{B11}$$

Hence, the location of compressors can be expressed as relative positioning, and the length of each pipeline segment can be obtained as the Euclidean distance between the corresponding nodes, this is,

$$\ell_{ij} = [(\delta x_{ij})^2 + (\delta z_{ij})^2]^{1/2} \quad \text{all } (i,j) \in E \tag{B12}$$

Since the coordinates for gas wellheads $m \in V_W$, and demand points $\ell \in V_D$, are part of the specifications for the problem, constraints representing transportation paths k between gas wells and supply sites can be given by

$$\sum_{(i,j) \in S_k} \delta x_{ij} = L_{k,x} \quad \text{all } k$$
$$\sum_{(i,j) \in S_k} \delta z_{ij} = L_{k,z} \quad \text{all } k \tag{B13}$$

where

$$\left.\begin{array}{l} L_{k,x} = x_\ell - x_m \\ L_{k,z} = z_\ell - z_m \end{array}\right\} \ell \in V_D, m \in V_W$$

and where S_k is the set of pipeline segments in the transportation path k. Additional constraints are related to logical conditions that state that whenever a given compressor exists, the associated downstream pipeline segment must also exist; that is,

$$\left.\begin{array}{r} \delta x_{ij} - U_x^U y_j \leq 0 \\ -\delta x_{ij} + U_x^L y_j \leq 0 \\ \delta z_{ij} - U_z^U y_j \leq 0 \\ -\delta z_{ij} + U_z^L y_j \leq 0 \end{array}\right\} \text{all } j \in \{V_I \backslash V_B\}, i \in V_d(j) \tag{B14}$$

where U_x^U, U_z^U and U_x^L, U_z^L are, respectively, upper and lower bounds on the coordinate difference variables δx_{ij} and δz_{ij}. Notice that it has been taken into account that these variables are not restricted in sign. Also, to ensure continuity of pressure profile, no constraints have been derived for either source nodes $j \in V_W$ or mixing/splitting points $j \in V_B$. Finally, there are either bounds or nonnegativity constraints on continuous variables, and integrality constraints on binary variables.

Through elimination of pipeline segment diameter (d_{ij}) and length (ℓ_{ij}) variables, using the explicit system of Eqs. B3 and B12, the problem can be formulated in terms of only pressures, coordinate differences, and compressor powers. Since power requirements w_j for compressors appear with positive coefficients in the objective function Eq. B1, and this is a minimization problem, the compressor problem expressions in Eq. B5 can be written as less than or equal inequality constraints. It can readily be shown that the variable transformations,

$$p_{d,j} = \exp(u_j)$$
$$p_{s,j} = \exp(v_j) \tag{B15}$$

lead to convex nonlinear functions that appear in the objective function and in the constraints of Eq. B5. Performing all of the manipulations above, the following MINLP program can be obtained as the underlying general model for the gas pipeline problem as addressed in this paper:

$$\begin{aligned} \text{Minimize} \quad & C_f \sum_{j \in V_W U V_I} y_j + C_o \sum_{j \in V_W U V_I} w_j \\ & + C_p \Bigg\{ \sum_{i \in V_W \mathrm{U} V_I, j \in \{V_d(i) \backslash V_D\}} B_{ij}^a [\delta x_{ij}^2 + \delta z_{ij}^2]^{1/2(\alpha+1)} \\ & \cdot [\exp(2u_i) - \exp(2v_j)]^{-\alpha} \\ & + \sum_{j \in V_D, i \in V_u(j)} B_{ij}^a [\delta x_{ij}^2 + \delta z_{ij}^2]^{1/2(\alpha+1)} \\ & \cdot [\exp(2u_i) - (p_{\text{out},j})^2]^{-\alpha} \Bigg\} \end{aligned}$$

subject to:

$$u_j - v_j \geq 0 \qquad j \in V_I$$

$$u_j - v_i \geq \epsilon \qquad j \in V_W \cup V_I, i \in \{V_d(j) \backslash V_D\}$$

$$u_j \geq \ln(p_{\text{in},j}) \qquad j \in V_W$$

$$u_j \geq \ln(p_{\text{out},i}) + \epsilon \qquad i \in V_D, j \in V_u(i)$$

$$\exp[b(u_j - v_j)] - F_j w_j \leq 1 \qquad j \in V_I$$

$$\exp\{b[u_j - \ln(p_{\text{in},j})]\} - F_j w_j \leq 1 \qquad j \in V_W$$

$$\left.\begin{array}{l} \sum_{(i,j) \in S_k} \delta x_{ij} = L_{k,x} \\ \sum_{(i,j) \in S_k} \delta z_{ij} = L_{k,z} \end{array}\right\} \text{all } k$$

$$\left.\begin{array}{r} \delta x_{ij} - U_x^U y_j \leq 0 \\ -\delta x_{ij} - U_x^L y_j \leq 0 \\ \delta z_{ij} - U_z^U y_j \leq 0 \\ -\delta z_{ij} + U_z^L y_j \leq 0 \\ w_j - U_c^U y_j \leq 0 \end{array}\right\} \begin{array}{r} \text{all } j \in \{V_I \backslash V_B\}, i \in V_d(j) \\ j \in V_W \cup V_I \end{array}$$

$$\delta x_{km}, \delta z_{km} \in R^1, u_j, v_i, w_j \geq 0 \quad \text{all } j \in V_W \cup V_I$$

$$y_j \in \{0, 1\} \qquad i \in V_I, (k, m) \in E$$

where $\epsilon = 1 \times 10^{-8}$ was introduced to avoid singularities in the objective function. This MINLP problem has exactly the same mathematical structure as program P2, and it can be shown that all of the nonlinear functions are convex.

Literature Cited

Beale, E. M. L., "Integer Programming," *The State of the Art in Numerical Analysis,* D. Jacobs, ed., Academic Press, London, 409 (1977).

Beale, E. M. L., and J. A. Tomlin, "Special Facilities in a General Mathematical Programming System for Nonconvex Problems Using Ordered Sets of Variables," *Proc. 5th Int. Conf. Operational Res.,* J. Lawrence, ed., Tavistock, London, 447 (1970).

Bickel, T. C., D. M. Himmelblau, and T. F. Edgar, "Optimal Design of Gas Transmission Networks," *Soc. Pet. Eng. J.,* (Apr. 1978).

Dunn, G. H., "Optimal Design of Gas Transmission Networks," M.S. Thesis, Dept. Chem. Eng., Univ. Texas, Austin (1980).

Duran, M. A. "A Mixed-Integer Nonlinear Programming Approach for the Systematic Synthesis of Engineering Systems," Ph.D. Thesis, Dept. Chem. Eng., Carnegie-Mellon Univ., Pittsburgh (1984).

Duran, M. A., and I. E. Grossmann, "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. The Relation with Generalized Benders Decomposition," Report DRC-06-68-84, Design Res. Ctr, Carnegie-Mellon Univ., Pittsburgh (1984).

———, "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs," submitted, *Math. Prog.* (1985a).

———, "Simultaneous Optimization and Heat Integration of Chemical Processes," *AIChE J.* **32,** 123 (1986).

Geoffrion, A. M., "Generalized Benders Decomposition," *J. Optimization Theory and Appl.,* **10,** 237 (1972).

Glover, F., "Improved Linear Integer Programming Formulations of Nonlinear Integer Problems," *Management Sci.,* **22,** 455 (1975).

Grossmann, I. E., "Mixed-Integer Programming Approach for the Synthesis of Integrated Process Flowsheets," *Comp. Chem. Eng.* **9,** 463 (1985).

Gupta, O. K., "Branch-and-Bound Experiments in Nonlinear Integer Programming," Ph.D. Thesis, Sch. Ind. Eng., Purdue Univ. West Lafayette, IN (1980).

Murtagh, B. A., and M. A. Saunders, "A Projected Lagrangian Algorithm and Its Implementation for Sparse Nonlinear Constraints, and MINOS/AUGMENTED User's Manual," Tech. Reports SOL 80-IR and SOL 80-14, Systems Optimization Lab., Dept. Ons. Res., Stanford Univ., CA (1980).

Nishida, N., G. Stephanopoulos, and A. W. Westerberg, "A Review of Process Synthesis," *AIChE J.,* **27,** 321 (1981).

Papoulias, S. A., and I. E. Grossmann, "A Structural Optimization Approach in Process Synthesis. I: Utility Systems," *Comp. Chem. Eng.,* **7,** 695 (1983a).

———, "A Structural Optimization Approach in Process Synthesis. II: Heat Recovery Networks," *Comp. Chem. Eng.,* **7,** 707 (1983b).

———, "A Structural Optimization Approach in Process Synthesis. III: Total Processing Systems," *Comp. Chem. Eng.,* **7,** 723 (1983c).

Schrage, L., *LP Models with LINDO (Linear Interactive Discrete Optimizer),* Scientific Press, Palo Alto (1981).

Soliman, F. I., "Optimization of Piping Networks," Ph.D. Thesis, Dept. Ons. Res., Sch. Mech. Ind. Eng., Univ. New South Wales, Australia (1982).

Stephanopoulos, G., "Synthesis of Process Flowsheets: An Adventure in Heuristic Design or a Utopia of Mathematical Programming?" *Foundations of Computer-Aided Chemical Process Design,* R. S. H. Mah and W. D. Seider, eds., Engineering Foundation, New York, **2,** 439 (1981).

Vaselenak, J. A., "Studies in the Design and Scheduling of Batch Processes," Ph.D. Thesis, Dept. Chem. Eng., Carnegie-Mellon Univ., Pittsburgh (1985).

Weymouth, T. R., "Problems in Natural Gas Engineering," *Trans. ASME,* **34,** 185 (1912).

Manuscript received Feb. 5, 1985, and revision received July 22, 1985.