Mathematical Programming 36 (1986) 307-339
North-Holland

AN OUTER-APPROXIMATION ALGORITHM FOR A CLASS OF MIXED-INTEGER NONLINEAR PROGRAMS

Marco A. DURAN* and Ignacio E. GROSSMANN

Department of Chemical Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213, USA

Received 28 May 1984

An outer-approximation algorithm is presented for solving mixed-integer nonlinear programming problems of a particular class. Linearity of the integer (or discrete) variables, and convexity of the nonlinear functions involving continuous variables are the main features in the underlying mathematical structure. Based on principles of decomposition, outer-approximation and relaxation, the proposed algorithm effectively exploits the structure of the problems, and consists of solving an alternating finite sequence of nonlinear programming subproblems and relaxed versions of a mixed-integer linear master program. Convergence and optimality properties of the algorithm are presented, as well as a general discussion on its implementation. Numerical results are reported for several example problems to illustrate the potential of the proposed algorithm for programs in the class addressed in this paper. Finally, a theoretical comparison with generalized Benders decomposition is presented on the lower bounds predicted by the relaxed master programs.

Key words: Mixed-integer nonlinear programming, outer-approximation, decomposition, cutting planes, computer-aided design.

1. Introduction

The nonlinear mathematical programming problem that is addressed in this paper involves both continuous (x) and integer (y) (or discrete valued) variables. The main characteristics defining the underlying mathematical structure are linearity of the integer variables and convexity of the nonlinear functions involving continuous variables. This particular class of problems can be represented by the following mixed-integer nonlinear programming (MINLP) program,

$$\begin{aligned} z = \min\ & c^{\mathrm{T}}y + f(x) \\ \text{s.t.}\ & g(x) + By \leqslant 0, \\ & x \in X \subset R^n, \\ & y \in U \subset R^m_+, \end{aligned} \qquad (P)$$

where the nonlinear functions $f: R^n \to R$ and those in the vector function $g: R^n \to R^p$ are assumed to be continuously differentiable and convex on the n-dimensional compact polyhedral convex set $X = \{x: x \in R^n,\ A_1 x \leqslant a_1\}$; $U = \{y: y \in Y,$ integer,

* Current address: Department of Engineering, Universidad Autonoma Metropolitana, Iztapalapa, Mexico City 09340, Mexico

$A_2y \leq a_2\}$ is a finite discrete set, for instance the non-negative integer points of some convex polytope, and for most applications Y corresponds to the unit hypercube $Y = \{0, 1\}^m$. B, A_1, A_2, and c, a_1, a_2 are respectively matrices and vectors of conformable dimensions; the vectors are column vectors unless specified otherwise; finally, some of the rows in B may be the zero row vector, which then defines nonlinear constraints in only the continuous variables.

The mathematical programming structure given by program P, with all of its variants (e.g. only f nonlinear, only g nonlinear), arises in several areas of practical interest. The applications include the synthesis problem in chemical process design [11, 13, 25], the optimal retrofit design of batch processes [46], the scheduling of units in a process industry powerhouse for satisfying utility demands at minimum operational cost [16], the mixed-integer nonlinear programming approach to the marketing problem of planning the optimal positioning of a new product in a multiattribute space [1, 18, 34, 47], the optimal unit commitment problem in an electric power system [6, 32, 38], the planning of facility investments for electric power generation [7, 8], the problem of determining the best treatment plant configuration for a regional water quality control management system [24], and the topological optimization of structures for designing transportation networks [29, 33].

MINLP problems belonging to the class described by program P can be solved using several different well known techniques. Such procedures include the generalized Benders decomposition method of Geoffrion [20, 5], the alternative dual approach of Balas [2], or a branch and bound search with solution of a nonlinear programming (NLP) subproblem at each node of the enumeration tree (see [27, 28, 17] for instance). Some of these methods address MINLP problems of a class more general than the one in this paper (e.g. nonseparability of continuous and discrete variables), and therefore one cannot expect that they will necessarily perform in the most efficient way when solving program P. Since there are no specific algorithms for efficiently handling MINLP problems in the particular class defined by program P, it is the purpose of this paper to propose an outer-approximation algorithm that fully exploits the linearity of the discrete variables and the convexity of the continuous valued functions.

The main ideas in the proposed algorithm are as follows. Because of the linearity of the discrete variables, the continuous and discrete feasible spaces of program P can be independently characterized. Furthermore, the continuous space corresponds to the intersection of a finite number of compact convex regions, where each region is determined by a different discrete parameterization. Hence, linearity in the continuous variables can be introduced into the problem P if a polyhedral representation is provided for each of those compact convex sets. To achieve this goal, outer-approximation [19] of a convex set by intersection of its collection of supporting half-spaces can be used. That outer-approximation will define the master program in the procedure as the equivalent mixed-integer linear programming (MILP) representation of the original MINLP program P. Because of the potentially many continuous points required for outer-approximation, a strategy based on relaxation

[19] will be implemented to build up increasingly tight relaxations of the master program which will select discrete combinations. The continuous points for outer-approximation will be given by the optimal primal solutions of convex nonlinear programs that represent the projection [19] of problem P onto the discrete space. The proposed algorithm consists then of solving an alternating finite sequence of nonlinear programming subproblems and relaxed versions of a mixed-integer linear master program. The algorithm can also be viewed as a cutting plane method for solving the general convex programming problem where some of the variables are discrete and appear as in program P.

From the above discussion it follows that the outer-approximation algorithm and the generalized Benders decomposition method of Geoffrion [20] make use of the same mathematical tools, namely, projection, outer-approximation and relaxation. The main difference between these procedures however, is the type of outer-approximation to define the corresponding master program. While in the outer-approximation algorithm the optimal primal information of the subproblems is used to define a mixed-integer linear master program, in the generalized Benders decomposition method the optimal dual information is used, such that the master program corresponds to an initially poorly constrained pseudo-pure integer linear program (i.e. a program involving only discrete variables and one artificial upper bound continuous variable). A detailed discussion of the relationship between the proposed algorithm and the generalized Benders decomposition method of Geoffrion [20] is given in Duran [11].

This paper presents the ideas underlying the proposed outer-approximation algorithm and its basic properties such as convergence and optimality. Implementation considerations, mainly for efficiently solving the sequence of relaxed versions of the master program, are briefly discussed. Finally, to gain insight into the algorithm properties and to illustrate its computational performance, numerical results are reported for several example problems which were also solved with the generalized Benders decomposition method as presented by Geoffrion [20], and a branch and bound procedure. Based on these results, an interesting trend that was identified is that the lower bounds predicted by the outer-approximation algorithm were tighter than the bounds predicted by generalized Benders decomposition. A theoretical proof on the relation of these bounds is given for the first iteration. The complete analysis is given in [11] and [12].

2. Outer-approximation

From a conceptual point of view, algorithms based on outer-approximation [19] describe the solution region of a given problem as the intersection of an infinite collection of sets. The present algorithm will make use of outer-approximation based on characterization of convex sets (see [37, 40] for instance) through intersection of supporting half-spaces. The objective of the approximation will be to provide a polyhedral representation of the continuous feasible space of program P. Such a representation will render linearity in the continuous variables, and enable to replace the difficult MINLP program P with a mixed-integer linear programming (MILP) problem.

Since f in the objective function is convex, so is $[f-\mu]$, where μ is a scalar variable. Without loss of generality, problem P can then be rewritten as the following program with a linear objective function:

$$\begin{aligned} z = \min\ & c^{\mathrm{T}} y + \mu \\ \text{s.t.}\ & f(x) - \mu \leqslant 0, \\ & g(x) + By \leqslant 0, \\ & \quad x \in X,\ y \in U,\ \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}]. \end{aligned} \tag{P_0}$$

where f_{L} and f_{U} are valid finite bounds given by $f_{\mathrm{L}} = \min\{f(x): x \in X\}$, and $f_{\mathrm{U}} = \max\{f(x): x \in X\}$. It will be assumed throughout this paper that the following suitable form of Slater's constraint qualification holds; namely, there exists a point $x \in X$ such that $g(x) + By < 0$, for each $y \in U \cap V$, where

$$V = \{y: g(x) + By \leqslant 0 \text{ for some } x \in X\}. \tag{1}$$

Let $F(y)$ define for each $y \in U \cap V$ the associated continuous feasible space in program P_0,

$$F(y) = \{x, \mu: x \in X, \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}], f(x) - \mu \leqslant 0, g(x) + By \leqslant 0\}. \tag{2}$$

Due to convexity of the functions in f and g, and to the compactness of the polyhedral convex set X, $F(y)$ for each $y \in U \cap V$ is a closed convex set [37, 40]. Therefore, the natural polyhedral representation [37, 40] of $F(y)$ is given by the intersection of its collection of homogeneous half-spaces. By the assumed linearity of the discrete variables, different y assignments lead only to different locations in the space for the regions $F(y)$. Therefore, intersecting the polyhedral representations of $F(y)$ for all $y \in U \cap V$ and using the differentiability property of the functions, the feasible region of the program P_0 can be defined by the following infinite set of supporting half-spaces,

$$\left.\begin{aligned} 0 \geqslant f(x) - \mu \geqslant f(x^i) + \nabla f(x^i)^{\mathrm{T}}(x - x^i) - \mu \\ 0 \geqslant g(x) + By \geqslant g(x^i) + \nabla g(x^i)^{\mathrm{T}}(x - x^i) + By \end{aligned}\right\}, \quad \text{all } x^i \in X, \\ x \in X,\ \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}],\ y \in U. \tag{3}$$

Here, $\nabla f(x^i)$ is the n-gradient vector and $\nabla g(x^i)$ the $n \times p$ jacobian matrix evaluated at given $x^i \in X$. The half-spaces in (3) correspond to the approximation of the convex functions in f and g by the pointwise maximum of the collection of their linear supports. Examples of outer-approximation at a finite number of points are illustrated in Figs. 1 and 2. In Fig. 1, H_{11}, H_{12}, and H_{21}, H_{22}, correspond to supporting half-spaces of g_1 and g_2, respectively, at points x_1 and x_2.

Outer-approximation of the feasible space of program P_0, as defined by (3), renders linearity in the constraints and objective function of P, and leads to the

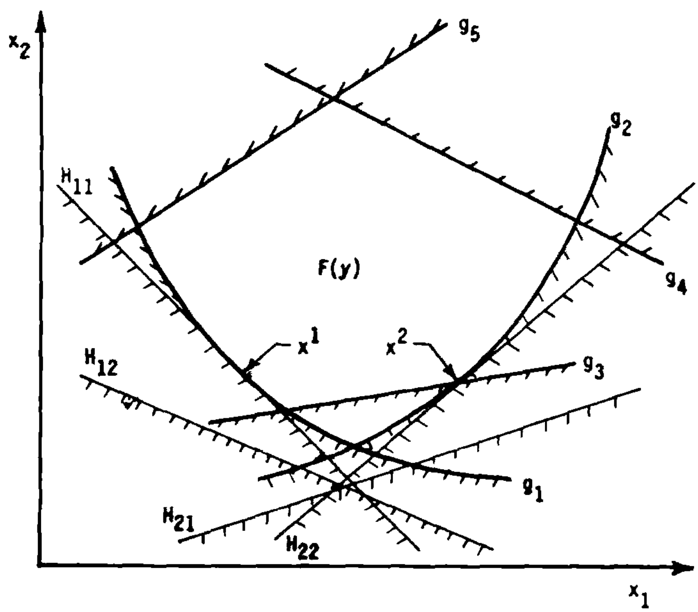

Fig. 1. Outer-approximation (at two points) of a convex set in R^2.

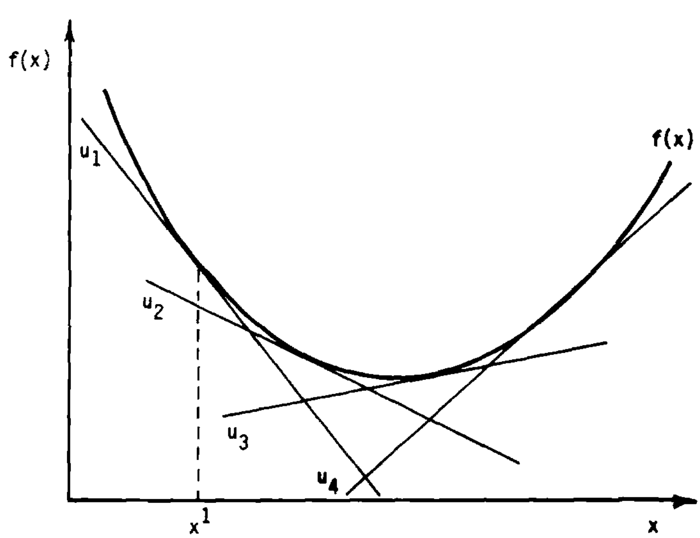

Fig. 2. Outer-approximation (at four points) of a convex function in R^1.

following semi-infinite mixed-integer linear programming formulation,

$$z = \min\ c^{\mathrm{T}}y + \mu$$
$$\text{s.t.}\quad \left.\begin{array}{l} f(x^i) + \nabla f(x^i)^{\mathrm{T}}(x - x^i) - \mu \leqslant 0 \\ g(x^i) + \nabla g(x^i)^{\mathrm{T}}(x - x^i) + By \leqslant 0 \end{array}\right\}, \text{ all } x^i \in X, \qquad (P_1)$$
$$x \in X,\ \ \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}],\ \ y \in U.$$

By the same outer-approximation procedure above, it can easily be shown that V, the set of y variables that yield non-empty continuous feasible regions, has the equivalent representation,

$$V \equiv \{y\colon g(x^i) + \nabla g(x^i)^{\mathrm{T}}(x - x^i) + By \leqslant 0 \text{ all } x^i \in X, \text{ for some } x \in X\} \qquad (4)$$

and that under this definition V is embedded in the constraint set of program P_1.

The following lemma, which can be easily proved [11], establishes the equivalence between problem P_0, and the semi-infinite mixed-integer linear program P_1.

Lemma 1. *Let the assumptions with respect to functions and sets in problem P_0 hold, then problems P_1 and P_0 are equivalent.*

3. Master program

Although problem P_1 is a mixed-integer linear programming program, it involves an infinite number of constraints and is in general difficult to solve. However, advantage can be taken of the fact that the set $U \cap V$ is discrete and finite. That is, the concept of projection [19] of program P onto the discrete space can be used to identify selected continuous points x^i for outer-approximations in problem P_1. The projection of program P onto y is given by,

$$z = \min_{y} \left[\operatorname*{infimum}_{x \in X} \{c^{\mathrm{T}}y + f(x)\colon g(x) + By \leqslant 0\} \right] \qquad (5)$$
$$\text{s.t.}\quad y \in U \cap V.$$

It can easily be shown that this projected problem is equivalent to program P. For given $y^i \in U \cap V$, the infimal value function of the "inner" problem in (5) is precisely the optimal value of program P for fixed y^i. Further, by the assumptions in the problem, for each $y^i \in U \cap V$ the infimum is attained and corresponds to the optimal value $z(y^i)$ of the nonlinear programming subproblem,

$$z(y^i) = c^{\mathrm{T}}y^i + \min f(x)$$
$$\text{s.t.}\quad g(x) + By^i \leqslant 0, \qquad (S(y))$$
$$x \in X.$$

It is clear that program P is not a convex program in x and y jointly, but fixing y renders it so in x for $S(y)$. Thus, the first observation is that for y^i to be a candidate for the optimal solution to problem P, y^i must be such that $S(y^i)$ is feasible (i.e. $y^i \in U \cap V$), and then the best continuous point x^i associated with y^i is the optimal solution of the corresponding subproblem $S(y^i)$. Secondly, according to theorems for characterization of integer polyhedra [22] and linear programming theory, the mixed-integer solution to problem P_1 is such that the integer part y^i is an extreme point of the convex hull of feasible integer solutions (Conv($U \cap V$)), and the continuous part is given by the boundary point $(x = x^i, f(x^i))$ in the linear support to $f(x)$ (3) associated with y^i. Therefore, as given by projection of program P onto y-space, the finite set of continuous points x^i to be considered for outer-approximation in problem P_1 are actually the optimal solutions of the subproblems $S(y^i)$ defined for the finite number of all integer points $y^i \in U \cap V$. The master program in its final form is then given by the following mixed-integer linear programming program,

$$\begin{aligned} z = \min\ & c^{\mathrm{T}} y + \mu \\ \text{s.t.}\ & \left.\begin{aligned} & f(x^i) + \nabla f(x^i)^{\mathrm{T}}(x - x^i) - \mu \leqslant 0 \\ & g(x^i) + \nabla g(x^i)^{\mathrm{T}}(x - x^i) + By \leqslant 0 \end{aligned}\right\}, \text{ for all } i \in T, \\ & x \in X,\ \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}],\ y \in U \end{aligned} \qquad (M)$$

where

$$T = \{i\colon x^i \text{ optimal solution to } S(y^i), \text{ all } y^i \in U \cap V\}. \qquad (6)$$

Since the feasible space of problem P is assumed nonempty and compact, finite optimal solutions exist for both programs P and M. The following theorem establishes the relation between these optimal solutions.

Theorem 1. *(x^*, y^*) is optimal in P iff (x^*, y^*) is optimal in M with $\mu^* = f(x^*)$.*

Proof. Assume that (x^*, y^*) is optimal in P. Then, $y^* \in U \cap V$ and $(x^*, \mu^*) \in F(y^*)$, where $\mu^* = f(x^*)$. Further, for (x^*, y^*) the infimum value is attained in the projected problem (5), i.e. $z = c^{\mathrm{T}} y^* + f(x^*) \leqslant c^{\mathrm{T}} y + f(x)$ for all x, y. Since $(x^*, \mu^*) \in F(y^*)$, $y^* \in U \cap V$, the outer-approximation half-spaces in (3) hold for all x^i, $i \in T$. But $x^* = x^k$, $k \in T$, and hence (x^*, μ^*, y^*) is feasible in program M. Therefore, since $(x^*, \mu^*) \in F(y^*)$, it follows that $0 = f(x^*) - \mu^* \geqslant \sup\{f(x^i) + \nabla f(x^i)^{\mathrm{T}}(x^* - x^i) - \mu^*$: all $i \in T\}$. The pointwise maximum must be achieved at a boundary point $(x^i, f(x^i))$, otherwise the result follows for the unconstrained minimum. Therefore, for $y^* \in U \cap V$, x^* optimal in $S(y^*)$ implies $z = c^{\mathrm{T}} y^* + \mu^*$, $\mu^* = f(x^*)$. The proof in the other direction is similar. □

Although the master program M involves outer-approximation at only a finite number of well defined points x^i, it has the drawback that it requires predetermination of the outer-approximations associated with all possible values $y^i \in U \cap V$. To circumvent that difficulty a strategy based on relaxation [19] can be used. The relaxed version of the master program to be solved at iteration k can be formulated as

$$\begin{aligned} z^k &= \min\ c^{\mathrm{T}} y + \mu, \\ \text{s.t.}\ \ & (x, y) \in \Omega^k, \\ & x \in X,\ y \in U,\ \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}], \end{aligned} \tag{M^k}$$

where

$$\begin{aligned} \Omega^k = \{x, y &: f(x^i) + \nabla f(x^i)^{\mathrm{T}}(x - x^i) - \mu \leq 0, \\ & g(x^i) + \nabla g(x^i)^{\mathrm{T}}(x - x^i) + By \leq 0,\ \text{all}\ i \in T^k \subseteq T\}, \end{aligned} \tag{7}$$

$$T^k = \{i: x^i \text{ optimal solution to } S(y^i),\ i = 1, 2, \ldots, k\}.$$

The use of relaxation as the strategy for solving program M then implies: (i) at iteration k, solve the relaxed master program M^k that ignores all but some of the constraints in M (i.e. ignores $i \in \{T \setminus T^k\}$). (ii) If the solution to M^k, (x, y^{k+1}), does not satisfy certain termination criteria, solve the subproblem $S(y^{k+1})$ to determine the continuous point x^{k+1} for outer-approximation. (iii) Construct the new relaxed master M^{k+1} by intersecting the feasible space at iteration k with the set of closed half-spaces associated with x^{k+1} (i.e. $T^{k+1} = T^k \cup \{k+1\}$ in (7) to define Ω^{k+1}).

4. Bounding properties

Let G and Γ^k denote respectively the feasible spaces of problems P_0 and the relaxed version M^k of the master program M,

$$G = \{x, y: x \in X,\ y \in U,\ f(x) - \mu \leq 0,\ g(x) + By \leq 0\}, \tag{8}$$

$$\Gamma^k = (X \times U) \cap \Omega^k. \tag{9}$$

It then follows from (3) that for any relaxed master program M^k, the corresponding outer-approximation set Γ^k overestimates the feasible set G as stated in the following lemma [11]:

Lemma 2. *$G \subseteq \Gamma^k$ for all $k \geq 1$.*

Therefore, according to Lemma 2, in an outer-approximation/relaxation strategy problem P_0: $\min\{c^{\mathrm{T}} y + \mu: (x, y) \in G,\ \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}]\}$ will be solved via a sequence of approximating problems,

$$M^k:\quad \min\{c^{\mathrm{T}} y + \mu: (x, y) \in \Gamma^k, \mu \in [f_{\mathrm{L}}, f_{\mathrm{U}}]\},\quad k = 1, 2, \ldots,$$

$$\text{where } G \subseteq \Gamma^k \subseteq \Gamma^{k-1} \subseteq \cdots \subseteq \Gamma^1. \tag{10}$$

From the concept of problem relaxation [19] it then follows that

$$\min\{c^{\mathrm{T}}y+\mu\colon (x,y)\in G, \mu\in[f_{\mathrm{L}},f_{\mathrm{U}}]\}\geqslant \min\{c^{\mathrm{T}}y+\mu\colon (x,y)\in \Gamma^k, \mu\in[f_{\mathrm{L}},f_{\mathrm{U}}]\}\geqslant\cdots\geqslant \min\{c^{\mathrm{T}}y+\mu\colon (x,y)\in \Gamma^1, \mu\in[f_{\mathrm{L}},f_{\mathrm{U}}]\}. \tag{11}$$

That is, the sequence of optimal objective function values z^k, found as solutions to successive relaxed master programs M^k, must be a monotone nondecreasing sequence of lower bounds on the optimal value of the original MINLP problem P. Since subproblem $S(y^i)$ is obtained from program P by fixing $y^i\in U\cap V$, $S(y^i)$ is a restriction to P and the following also holds,

$$z=\min\{c^{\mathrm{T}}y+f(x)\colon (x,y)\in G\}\leqslant z(y^i) = \min\{c^{\mathrm{T}}y^i+f(x)\colon x\in X, g(x)+By^i\leqslant 0\}. \tag{12}$$

That is, for any $y^i\in U\cap V$ the optimal objective function value of subproblem $S(y^i)$ provides a valid upper bound on the optimal value of problem P. Obviously, the sequence of values $z(y^i)$ need not be monotone nonincreasing.

The bounding properties discussed above can be used both to enhance the algorithm efficiency and to provide termination conditions. In particular, without loss of generality the constraint $\mu\in[f_{\mathrm{L}},f_{\mathrm{U}}]$ in the master program can be replaced by the stronger valid bounds,

$$z^{k-1}\leqslant c^{\mathrm{T}}y+\mu<z^* \tag{13}$$

where $z^{k-1}=\min\{c^{\mathrm{T}}y+\mu\colon (x,y)\in\Gamma^{k-1}\}$, and z^* is the current best upper bound. Although the constraint $z^{k-1}\leqslant c^{\mathrm{T}}y+\mu$ is redundant, it will be considered here because it may act as a weak cut to avoid needless enumeration in successive relaxed master programs.

5. Infeasible subproblems

For $y\in U$ to be a candidate for the optimal solution to program P, y must be such that the y-parameterized subproblem $S(y)$ is feasible. In other words, the condition $y\in U\cap V$ must hold, where by (4) V has the equivalent representation,

$$V\equiv\{y\colon g(x^i)+\nabla g(x^i)^{\mathrm{T}}(x-x^i)+By\leqslant 0 \text{ all } x^i\in X, \text{ for some } x\in X\}. \tag{14}$$

It should be pointed out that because of the relaxation strategy employed in the algorithm, as iterations proceed subsets of the constraints in V, as given in (14), are being automatically generated from the outer-approximation half-spaces (3) associated with the $y\in U\cap V$ that have already been tested. That is, at iteration k in the algorithm a relaxation V^k of the set V will be present in the relaxed master program M^k, and will be given by

$$V^k=\{y\colon g(x^i)+\nabla g(x^i)^{\mathrm{T}}(x-x^i)+By\leqslant 0, \text{ all } i\in T^k, \text{ for some } x\in X\}. \tag{15}$$

However, for each iteration k the set V^k is an overestimation of V, and therefore the selection in program M^k of an integer combination y leading to a feasible subproblem $S(y)$ (i.e. $y \in V$), cannot always be guaranteed. Hence, if a $y \in V^k$ such that $y \notin V$ is obtained as solution to a relaxed master program version M^k, the selection of that y has to be prevented in subsequent iterations. The easiest way of performing this task, without altering the bounding properties in the algorithm, is to eliminate $y \notin V$ by adding appropriate half-spaces to V^k so as to construct the more constrained set V^{k+1} to be considered in the subsequent program M^{k+1}. The natural tool for testing whether or not $y^i \in V^k$ is also in the set V is the y^i-parametrized subproblem $S(y^i)$ itself. When optimizing an infeasible subproblem $S(y^i)$, any NLP algorithm will yield an associated result $x^i \in X$ (e.g. minimization of constraint violations). Therefore, the following set of supporting half-spaces can be added to V^k such as to define V^{k+1},

$$f(x^i)+\nabla f(x^i)^{\mathrm{T}}(x-x^i)-\mu \leq 0, \qquad g(x^i)+\nabla g(x^i)^{\mathrm{T}}(x-x^i)+By \leq 0. \tag{16}$$

It is clear that the constraints in (16) eliminate from consideration not only the point (x^i, y^i) but other infeasible regions. However, the constraints in (16) do not guarantee that the $y^i \notin V$ will not be selected again (i.e. $y^i \in V^{k+1}$), since they could be satisfied for some $x \neq x^i$, $x \in X$, $y^i \in V^{k+1}$ according to

$$0 < g(x^i)+By^i \geq g(x^i)+\nabla g(x^i)^{\mathrm{T}}(x-x^i)+By^i \leq 0.$$

Hence, in order to totally eliminate $y^i \notin V$ such that $y^i \in V^k$ from further consideration, an integer cut for deleting y^i must also be introduced.

6. Algorithm

The outer-approximation algorithm can now be formally stated. All the hypotheses implied in the development, in particular the ones which ensure that P has a finite optimal solution, are assumed to hold. The algorithm is then as follows:

Define for given $x^i \in R^n$

$$C(x^i)=\{x, y: f(x^i)+\nabla f(x^i)^{\mathrm{T}}(x-x^i)-\mu \leq 0,$$
$$g(x^i)+\nabla g(x^i)^{\mathrm{T}}(x-x^i)+By \leq 0,\ \mu \in R^1\}.$$

Step 1. Set $\Omega^0 = R^n \times R^m$, lower bound $z^0 = -\infty$, upper bound $z^* = +\infty$, $i=1$. Select an integer combination $y^1 \in U$, or $y^1 \in U \cap V$ if available.

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

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

One of the following cases must occur:

(a) Problem $S(y^i)$ has a finite optimal solution $(x^i, z(y^i))$, where $z(y^i)$ is a valid upper bound on the optimal value of the MINLP program P.

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

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

Set $\Omega^i = \Omega^{i-1} \cap C(x^i)$, and go to Step 3.

(b) Problem $S(y^i)$ is infeasible (i.e. $y^i \notin V$) with associated result x^i.
Derive and add to M^i an integer cut to eliminate y^i from further consideration.
Set $\Omega^i = \Omega^{i-1} \cap C(x^i)$.

Step 3. Solve the current relaxed MILP master program M^i.

$$
\begin{aligned}
z^i = \min\ & c^{\mathrm{T}}y + \mu \\
\text{s.t.}\ & (x, y) \in \Omega^i, \\
& z^{i-1} \leq c^{\mathrm{T}}y + \mu < z^* \\
& x \in X,\ y \in U,\ \mu \in R^1, \\
& y \in (\text{set of integer cuts}).
\end{aligned}
\qquad (M^i)
$$

One of the following cases must occur:

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

The optimal solution to the original MINLP program P 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 2a.

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

Set $y^{i+1} = y$, and $i = i + 1$ to indicate a new iteration.

Return to Step 2.

The above iterative procedure indicates that the algorithm consists of solving an alternating sequence of nonlinear programming subproblems $S(y)$ and relaxed mixed-integer linear master programs M^i. It should be noted that if all the functions in problem P were linear, the relaxed master program at the first iteration would be identical to the original problem, and hence the above algorithm would terminate in at most two iterations. For the nonlinear case, the algorithm converges in a finite number of steps to the optimal solution of problem P as shown in the next section.

7. Convergence and optimality

The convergence of the proposed algorithm can be proved based on at least two different criteria. The first criterion rests on the bounding properties derived for the procedure. The second one is standard and relies on the finiteness of the set U of integer constrained variables.

In the i-th relaxed master program M^i (iteration i), the constraint $z^{i-1} \leq c^T y + \mu < z^*$ explicitly considers the bounding properties in the procedure (see Section 4), and the lack of a feasible mixed-integer solution at step 3a of the algorithm implies the condition $c^T y + \mu \geq z^*$ for all of the remaining solutions because of the monotonicity of the sequence of lower bounds. That condition indicates crossing of lower (z^i) and upper (z^*) bounds, and thus convergence of the algorithm. The optimal solution (x^*, y^*) of program P is then given by the current best upper bound z^* corresponding to the optimal solution x^* of the y^*-parameterized subproblem $S(y^*)$. Optimality of this solution can be proved if in the constraint $c^T y + \mu < z^*$ the equality is also considered. The termination criterion at step 3a of the algorithm would then be to Stop when a repetition of discrete variable y^* is obtained. That is because in the next iteration the equality $z^i = c^T y^* + \mu^* = z^*$ would hold with $\mu^* = f(x^*)$. Under the above termination criterion, and considering the constraint $z^{i-1} = c^T y + \mu = z^*$, the relaxed program M^i can be expressed in terms of only the last outer-approximation, namely the one at x^*, such that when solving the program

$$\begin{aligned} & z^i = \min c^T y + \mu, \\ & \text{s.t.} \quad f(x^*) + \nabla f(x^*)^T (x - x^*) - \mu \leq 0, \\ & \qquad g(x^*) + \nabla g(x^*)^T (x - x^*) + By \leq 0, \\ & \qquad\quad z^i = z^*, \; x \in X, \; y \in U \cap (\text{integer cuts}), \; \mu \in R^1, \end{aligned} \qquad [(M^i)^*]$$

the algorithm will terminate at this stage ($y^{i+1} = y^*$, $y^t = y^*$, $t < i+1$).

Theorem 2. *If (x^*, y^*) is optimal in the relaxed version $(M^i)^*$ for some i, i.e. (x^{i+1}, y^{i+1}), then (x^*, y^*) is also optimal in problem P.*

Proof. Note that our assumptions on the problem ensure the existence of an optimal solution (x^*, y^*). Also, since x^* is optimal solution to subproblem $S(y^*)$, by convexity the inequalities in $(M^i)^*$ hold for any $x \in X = \{x: x \in R^n, A_1 x \leq a_1\}$, and so (x^*, y^*) is feasible to $(M^i)^*$. Both problems $S(y^*)$ and $(M^i)^*$ satisfy the given form of the Slater's condition. Then, the necessary and sufficient conditions for (x, y^*) to be optimal in $(M^i)^*$ is that $\exists \lambda \in R^1$, $u \in R^p$, $v \in R^q$, such that x, μ minimize L, where

$$\begin{aligned} L = {} & c^T y^* + \mu + \lambda [f(x^*) + \nabla f(x^*)^T (x - x^*) - \mu] \\ & + u^T [g(x^*) + \nabla g(x^*)^T (x - x^*) + By^*] + v^T [A_1 x - a_1] \end{aligned}$$

and satisfy

$$\begin{aligned} & f(x^*) + \nabla f(x^*)^T (x - x^*) - \mu \leq 0, \qquad g(x^*) + \nabla g(x^*)^T (x - x^*) + By^* \leq 0, \\ & A_1 x^* - a_1 \leq 0, \qquad \lambda [f(x^*) + \nabla f(x^*)^T (x - x^*) - \mu] = 0, \\ & u^T [g(x^*) + \nabla g(x^*)^T (x - x^*) + By^*] = 0, \qquad v^T [A_1 x^* - a_1] = 0, \\ & \lambda, \; u, \; v \geq 0. \end{aligned}$$

From the lagrangian function minimization it then follows that

$$\frac{\partial L}{\partial \mu}=0 \Rightarrow \lambda=1, \qquad \frac{\partial L}{\partial x}=0 \Rightarrow \lambda \nabla f(x^*)+\nabla g(x^*)u+A_1^{\mathrm{T}}v=0.$$

Thus if x^* is optimal to $(M^i)^*$, one can set $x=x^*$ in the above optimality conditions and determine

$$\nabla f(x^*)+\nabla g(x^*)u+A_1^{\mathrm{T}}v=0,$$
$$f(x^*)=\mu, \qquad g(x^*)+By^*\leqslant 0, \qquad A_1x^*-a_1\leqslant 0,$$
$$u^{\mathrm{T}}[g(x^*)+By^*]=0, \qquad v^{\mathrm{T}}[A_1x^*-a_1]=0,$$
$$u\geqslant 0, \quad v\geqslant 0,$$

that in turn imply

$$L^*=c^{\mathrm{T}}y^*+f(x^*)+u^{\mathrm{T}}[g(x^*)+By^*]+v^{\mathrm{T}}[A_1x^*-a_1]=c^{\mathrm{T}}y^*+f(x^*)$$

which are the optimality conditions of subproblem $S(y^*)$ with optimal solution x^*, $z^*=c^{\mathrm{T}}y^*+f(x^*)$. Hence, according to projection, (x^*, y^*) is the optimal solution of the program P, since if $y\neq y^*$ were optimal solution to P with $z<z^*$ that would imply a contradiction to the assumption that $z^*=z^i\leqslant z=\min\{c^{\mathrm{T}}y+f(x)\colon (x,y)\in G\}$. □

It is clear that the number of iterations required for convergence to the optimal solution is dependent on the particular nature of the problem. For the class of programs addressed in this paper, the finiteness of the outer-approximation algorithm rests on the fact that the set U is a finite discrete set as given by the assumptions in the problem.

Theorem 3. *Assume that all of the assumptions and properties in the procedure hold, in particular Theorem* 2 *and the bounding properties implied by Lemma* 2. *Given that U is a finite discrete set, the outer-approximation algorithm terminates in a finite number of iterations.*

Proof. The proof is standard and the main tools are the facts that U is finite, and that no y can ever be selected twice as a solution in the sequence of relaxed master programs at step 3 of the algorithm. If repetition of integer assignments were allowed, bounding properties and Theorem 2 would imply convergence to the optimal solution. Note that the worst performance of the algorithm would be the total enumeration of the integer elements in U, which is finite. □

8. Implementation considerations and refinements

The definition of the master program is the main feature that differentiates the proposed outer-approximation algorithm from similar algorithms such as the generalized Benders decomposition method. The relaxed versions of the MILP master program can be expected to provide good global approximations to the original nonlinear program, and hence good predictions for the lower bounds. This could have the advantage of reducing the number of iterations to find the optimal solution. However, since the relaxed master programs will grow in size as iterations proceed, the solution of the sequence of MILP problems can become the major bottleneck in large-scale applications. Although efficient codes are available [41] for solving MILP programs, the efficiency of the algorithm can be enhanced if the following considerations are taken into account. Firstly, integer cuts can be derived and added at each iteration such as to reduce the enumeration effort when solving subsequent relaxed master programs M^i. Secondly, a very desirable refinement would be to keep the size of the programs M^i as small as possible by using a constraint dropping scheme. Thirdly, the solution of the relaxed master problem could be prematurely terminated as soon as an integer solution is found that lies below the current upper bound. The latter option is clearly trivial to implement, but the first two require some explanation.

Integer cuts

It has been shown [3] that cutting planes derived from disjunctions associated with binary assignments in a partial search tree, often provide better bounds than the *LP* relaxation at each node. In any enumerative procedure it is possible to identify partial assignments $(y)_p$ of discrete variables associated with fathomed nodes whose corresponding subproblems are infeasible. Therefore, if for instance a branch and bound method is used for the solution of the increasingly constrained MILP problems M^i, the information on the infeasible fathomed nodes can be used to generate integer constraints that will eliminate those partial assignments from consideration when solving master versions of subsequent iterations. Other types of weak integer cuts that could be derived are the ones that will ensure that those previously considered integer combinations (for both infeasible and feasible NLP subproblems) cannot be encountered again. The two types of cuts described above may help expediting the enumeration procedure in subsequent relaxed master problems. For the case when the integer combination is an element of some unit hypercube (i.e. binary variables), the following well-known integer cut [4] will perform the above tasks.

Lemma 3. *Given any integer combination* $y^i = \{y^i_j: j = 1, \ldots, m\} \in \{0, 1\}^m$ *with index sets* $B^i = \{j: y^i_j = 1\}$, $NB^i = \{j: y^i_j = 0\}$, s.t. $|B^i| + |NB^i| = m$, *the integer constraint*

$$\sum_{j \in B^i} y_j - \sum_{j \in NB^i} y_j \leq |B^i| - 1$$

will be violated only by y^i *and no other* $y^k \neq y^i$.

Remark 1. For deriving integer cuts to eliminate partial assignments, the integer combination to be considered is a partial solution $(y)_p$ in a reduced integer space (i.e. $(y)_p \in \{0, 1\}^r$, $r \leq m$).

The above integer cuts are easy to derive and implement. An example of these cuts is given in Fig. 3, where cuts are derived from the infeasible nodes in an enumeration tree. Note that these cuts are valid because the MILP problems M^i will remain infeasible at these nodes in subsequent iterations. The generation of integer cuts should be introduced at Steps 2a and 3b of the algorithm for eliminating $y^i \in U \cap V$ and partial assignments, respectively.

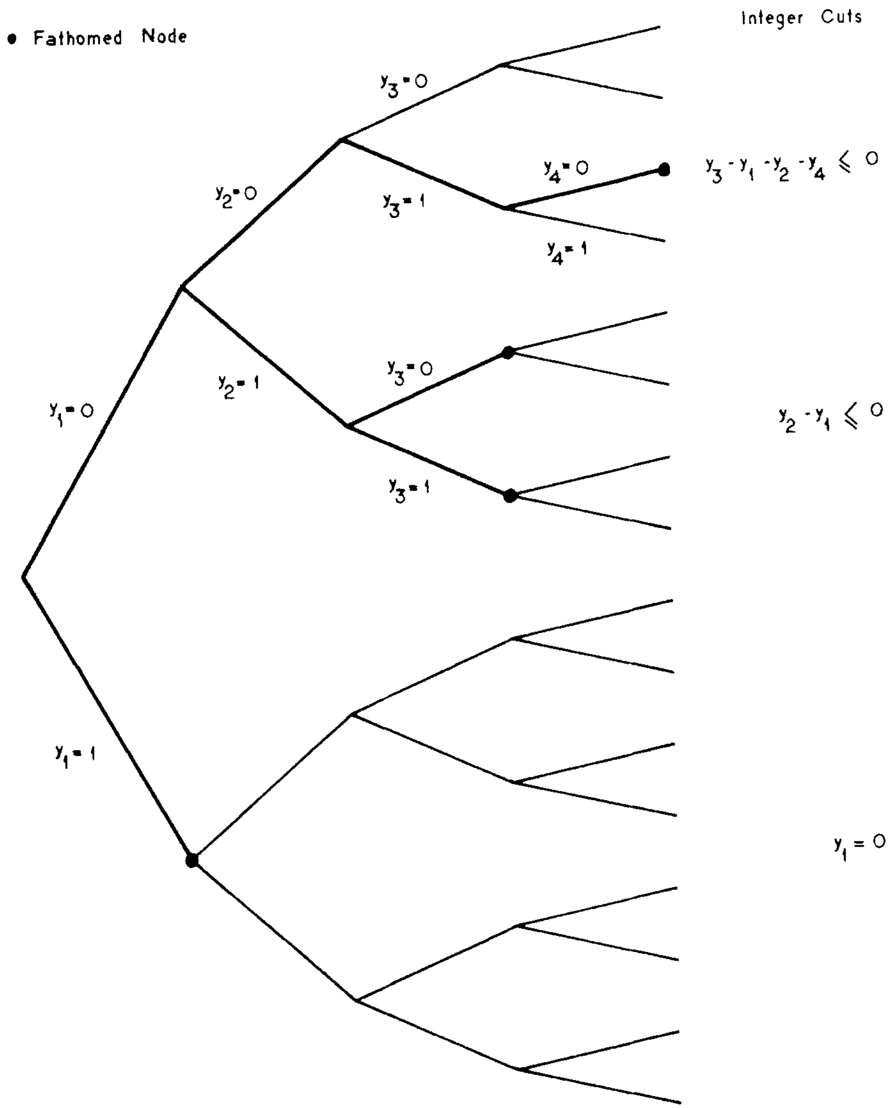

Fig. 3. Integer cuts from infeasible nodes in a search tree.

Additional integer cuts that could also be considered to expedite the solution of the relaxed master problem are cuts similar to the ones that have been proposed by Crowder et al. [10] and Van Roy and Wolsey [45] to tighten the constraint set in integer linear programming problems.

MILP programs: Dropping constraints

Provisions for keeping the size of the relaxed master programs as small as possible are clearly desirable to reduce the computational effort in Step 3 of the algorithm. The nature of those provisions must be such that they will not upset the inherent convergence properties of the proposed method. One alternative is to apply a mechanical analysis procedure to reduce or presolve each MILP problem M^i prior to its actual solution (e.g. see [9, 26, 41]). Another possibility is to develop conditions for a constraint dropping variant of the algorithm.

Since the proposed outer-approximation method can be regarded as a cutting plane procedure [31], a constraint dropping scheme could be considered based on the extensive and excellent work that has been done in this area for cutting plane algorithms that address nonlinear programs without integrality restrictions (see for instance [14, 42, 43, 44]). Further, since Slater's constraint qualification is assumed to hold for problem P_0, the strategy for dropping constraints given in the central cutting plane algorithm of Elzinga and Moore [15] is also interesting. Precise conditions to drop constraints from time to time without adversely affecting convergence properties in outer-approximation methods have been reported for instance in Mayne et al. [35], Hogan [30], and Gonzaga and Polak [23].

None of the above possibilities for trying to keep the size of the relaxed master programs as small as possible have been implemented yet in our procedure, but they would certainly be worth exploring.

9. Test problems

Four example problems were solved to illustrate the computational performance of the proposed outer-approximation algorithm. The actual formulations and data, and a brief discussion of the underlying physical meaning for the test problems are given in Appendix II. The first three problems in the set are simplified versions of process synthesis problems [11, 13], where the goal is to simultaneously determine the optimal structure and operating conditions of a process that will meet certain given design specifications. The fourth example corresponds to a modest extension of the known [1, 18, 34, 47] MINLP problem formulation for determining the optimal positioning of a new product in a multiattribute space. Note that problem No. 4 is actually a maximization problem, and to have a common framework in the next discussion results will be reported for the equivalent minimization formulation (i.e. minimizing the negative of the objective function). Table 1 summarizes the

Table 1
Parameters for test problems

Problem No.	Total number of variables $(n+m)$	Number of binary variables (m)	Total number of constraints[a]	Number of nonlinear constraints (p)
1	6	3	6	2
2	11	5	14	3
3	17	8	23	4
4	30	25	30	25

[a] Without including upper and lower bound constraints for the continuous variables.

parameters in the example problems characterizing size (i.e. number of variables and constraints), and degree of complexity (i.e. number of integer variables and nonlinear constraints).

The example problems were also solved using the generalized Benders decomposition method of Geoffrion [20] and a depth-first branch and bound procedure. It should be pointed out that versions without built-in refinements were used for all of the above methods. The objective in solving the test problems with the other two methods was not to perform an extensive numerical comparison with the proposed outer-approximation algorithm, but rather to identify some general computational trends. Although the problems vary in size and complexity, the sample is small and does not include many of the possible variants on the mathematical structure of the class. Therefore, based on the results of the present numerical experiments, no definite claims can be made about the superiority in time-efficiency of the proposed algorithm for the whole class of programs P. However, as will be mentioned later in the paper, the present limited numerical results have motivated an interesting generalization about the quality of the lower bounds predicted in the outer-approximation algorithm.

In relation to the implementation of the procedure, the example problems were solved using a preliminary version of the outer-approximation algorithm that does not include the refinements for integer cuts to eliminate partial assignments, nor for conditions to drop constraints in the MILP programs. The solution of the nonlinear programming (NLP) subproblems was obtained using the reduced gradient method (with a projected langrangian approach for nonlinear constraints) as implemented in the computer code MINOS/AUGMENTED [36]. The mixed-integer linear problems, associated with relaxed master programs, were solved with the computer code LINDO [39], which uses a depth-first branch and bound procedure. With respect to the generalized Benders decomposition (GBD) method, it was implemented as originally presented by Geoffrion [20] (i.e. adding to the master program only one new integer cut per iteration, such cut corresponding to the most violated (or nearly so) constraint). The NLP subproblems and relaxed versions of the pseudo-pure integer master program in GBD were solved using the codes MINOS and LINDO respectively. The branch and bound (BB) procedure was implemented in a standard fashion and a depth-first strategy was used. For solving the MINLP example problems with BB, at each node of the search tree the corresponding integrality constraints were relaxed and the resulting NLP's solved using the code MINOS.

In all of the above methods, the initial guess for an integer assignment was obtained rounding the solution of the MINLP problems with integrality relaxation. The rounding was performed such as to obtain a feasible associated NLP subproblem. Relevant numerical results describing the computational performance of the three methods are presented in the next section. The reported CPU-times correspond to a DEC-20 computer system.

10. Numerical results—Discussion

The main parameters characterizing the numerical performance of the proposed algorithm and the other two methods are reported in Tables 2 to 6. As seen from the results in these tables, for the test problems that were solved, the outer-approximation algorithm proved to be more efficient than the versions of both GBD and BB that were used. An interesting trend in the numerical behavior of the algorithm can readily be identified; namely, the fact that the number of iterations required by the proposed algorithm was always substantially smaller than by GBD (roughly 60% smaller on the average). Although the solution of each relaxed master program in the outer-approximation algorithm required a larger computational effort than the solution of the corresponding master program in GBD, the total computer times for the respective sequence of master problems were comparable in both methods. However, the smaller number of iterations required by the proposed method implied a smaller number of nonlinear programming (NLP) subproblems that had to be solved. This led to savings in total computational time of roughly

Table 2
Results example problem No. 1

	Branch and bound (depth first)	Generalized Benders decomposition	Proposed algorithm
iterations	8*	4	2
CPU-time(sec): #			
NLP subproblems	21.2	7.7	3.9
Master problems		4.9	4.3
Total	21.2	12.6	8.2
optimal solution found at iteration	7*	3	1

* Enumerated nodes, # DEC-20.

Table 3

Results example problem No. 2

	Branch and bound (depth first)	Generalized Benders decomposition	Proposed algorithm
iterations	25*	8	3
CPU-time(sec): #			
NLP subproblems	60.2	17.3	6.2
Master problems		11.0	9.5
Total	60.2	28.2	15.7
optimal solution found at iteration	17*	6	2

* Enumerated nodes, # DEC-20.

Table 4

Results example problem No. 3; $y^1 = \{y_j : j = 1, \ldots, 8\} = [1, 0, 0, 1, 0, 1, 0, 0]$

	Branch and bound (depth first)	Generalized Benders decomposition	Proposed algorithm
iterations	43*	10	4
CPU-time(sec): #			
NLP subproblems	128.1	28.2	7.4
Master problems		19.7	21.4
Total	128.1	47.9	28.8
optimal solution found at iteration	36*	9	2

* Enumerated nodes, # DEC-20.

Table 5

Results example problem No. 3; starting guess: optimal binary combination $y^1 = y^* = [0, 1, 0, 1, 0, 1, 0, 1]$

	Branch and bound	Generalized Benders	Proposed algorithm
iterations	39*	8	3
CPU-time(sec):			
NLP subproblems	117.1	21.6	5.6
Master problems		13.5	15.0
Total	117.1	35.1	20.6

40% on average with respect to GBD. Relative to branch and bound, the computer time savings with the present algorithm were 74% on average.

With the outer-approximation algorithm the optimal solution in the four examples was found roughly within the first half of the total number of iterations. In contrast,

Table 6

Results example problem No. 4

	Branch and bound (depth first)	Generalized Benders decomposition	Proposed algorithm
iterations	—	35‡	6
CPU-time(sec): #			
NLP subproblems	—	239.4	32.7
Master problems		835.4	971.0
Total	—	1074.8	1003.7
optimal solution found at iteration	—	‡	3

DEC-20, ‡ up to iteration 35 no optimal solution was found.

GBD had a tendency to find the optimal solution in the last few iterations. Finally, as problem size and degree of complexity increased, the differences in the computational performance of the three methods increased in favor of the proposed algorithm. For the fourth example problem, which involves 25 binary variables and 25 nonlinear constraints, the outer-approximation algorithm required only 6 iterations (optimal solution found at the third iteration). In almost the same total computational time, GBD performed 35 iterations in which no optimal solution was found.

The main difference between the outer-approximation algorithm and GBD is the definition of the master program. While in the former method the master is a MILP program, in GBD it corresponds to a pseudo-pure integer linear program. Therefore, the above numerical behavior was a clear indication that the proposed sequence of relaxed master programs provided a better global approximation than GBD to the original MINLP program. Evidence to support that was given by the "good" quality of the sequences of lower bounds predicted by the outer-approximation algorithm in all of the example problems. In Fig. 4, the progress of the solution procedure is shown for test problem No. 3. For that problem, the first predicted lower bound was -105.6 with the present procedure, and -886.8 with GBD. Further, while with the outer-approximation algorithm the gap between lower bounds and optimal objective function value (68.0) was closed in four iterations, for GBD the progress was slower and required 10 iterations. For the fourth test problem (minimization case), the first predicted lower bound was -14.1521 with the proposed algorithm and -55.1904 with GBD. After 35 iterations, the lower bound in GBD was -52.3904 which represents an improvement of only 6% toward the optimal objective function value (-7.7891).

11. Theoretical comparison with generalized Benders decomposition

In the four examples of the previous section the lower bound predicted at each iteration by the outer-approximation algorithm was always substantially higher than

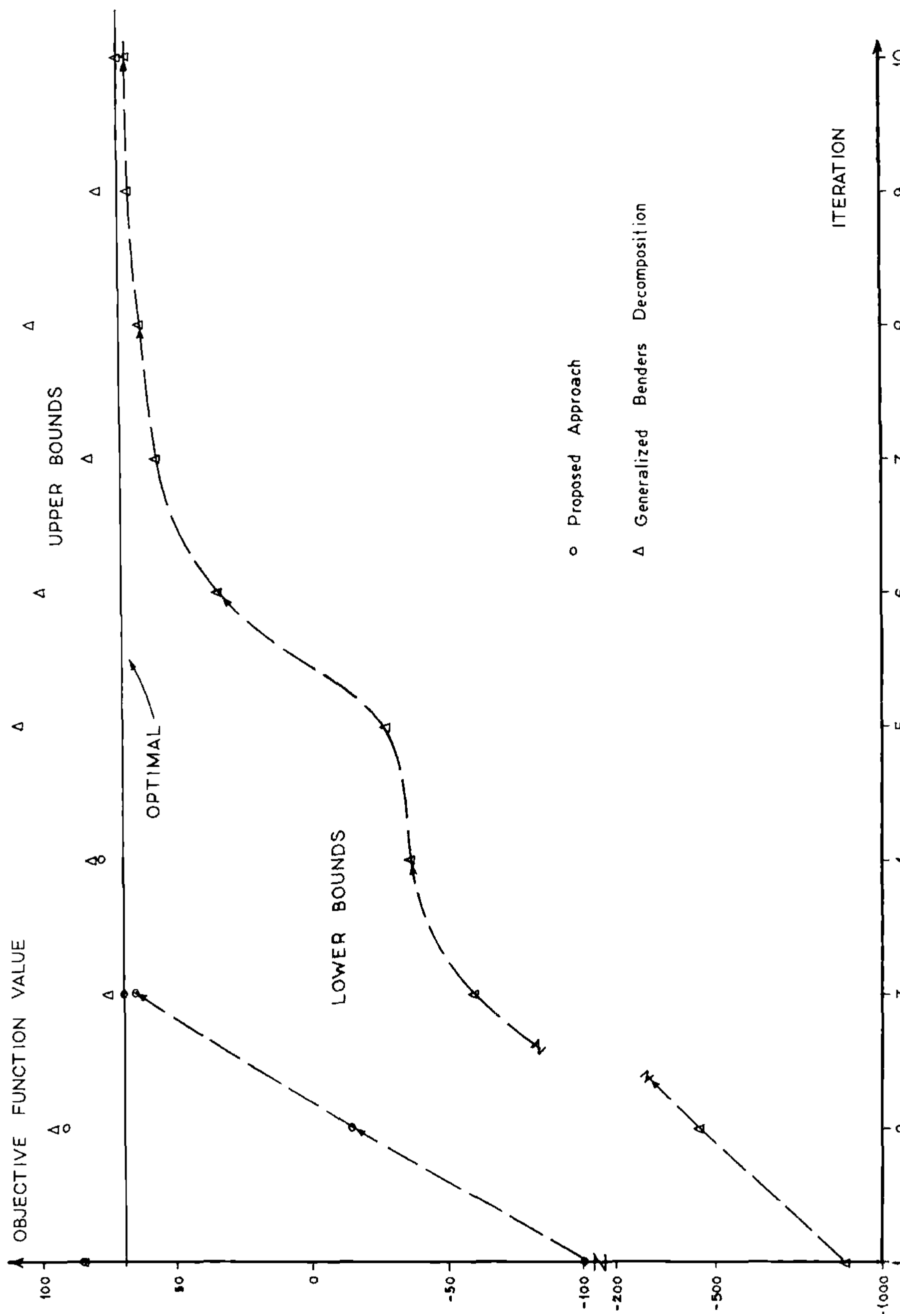

Fig. 4. Test problem No. 3: Sequences of Upper and Lower bounds.

the one predicted by the GBD method. A theoretical explanation that generalizes the above numerical behavior for the class of MINLP problems addressed in this paper is given in Duran's Ph.D. thesis [11] and is also available in the technical report [12]. In these references convex duality theory as applied to the NLP subproblems was used to prove that at each iteration the lower bound predicted by the relaxed master program in the outer-approximation algorithm will always be greater than or equal to the lower bound obtained in the GBD method. This in turn implies that the outer-approximation algorithm always requires fewer or the same number of iterations than the GBD method when applied to problem P.

The proof presented by Duran [11] determines first the relationship between the feasible spaces of the relaxed master programs in both methods. Then the relation among objective cuts (supports) is established in order to show that the lower bound predicted by the outer-approximation method is greater or equal than the one predicted by GBD for a given integer value. Finally, an inductive proof is given to show that the relation between the two bounds is maintained at every iteration. All the different cases arising from the selection of integer values that lead to feasible or infeasible NLP subproblems are considered. The proof given in Duran [11] is rather lengthy, and therefore out of the scope of this paper. However, Appendix I presents a condensed proof for the particular case when the lower bounds of the two methods are predicted at the first iteration, and based on any initial feasible integer value for the NLP subproblem.

It should be noted that the work per iteration in the outer-approximation algorithm is greater than the one in GBD due to the larger size of the relaxed master program in the former method. Therefore, despite the theoretical result on the relation of the lower bounds, no conclusions can be drawn on the computing times that are required by the two methods. However, the outer-approximation algorithm would seem to be promising in applications where the NLP subproblems are expensive to solve since it will often require relatively few iterations.

12. Conclusions

A primal decomposition algorithm based on outer-approximation has been proposed for efficiently solving a particular class of mixed-integer nonlinear programming problems. Outer-approximations of the continuous feasible region in the program have to be performed at only a finite set of well defined points. The proposed procedure alternates between nonlinear programming subproblems and relaxed versions of a mixed-integer linear master program. A study of the bounding and convergence properties of the algorithm has been presented. Conditions for an efficient implementation have also been discussed. The computational performance of the algorithm was illustrated on a set of four example problems, which were also solved with the generalized Benders decomposition method and a branch and bound procedure. In the numerical behavior of the proposed procedure, a general trend was observed consisting mainly of a relatively small number of iterations required for solving each test problem. An explanation to that fact was found in the good quality of the sequence of lower bounds predicted by the relaxations of the MILP master program. A theoretical explanation that generalizes the above observation is given in [11] and [12]. In this paper it has been given only when the lower bounds of the outer-approximation and GBD methods are computed from a feasible NLP subproblem at the first iteration. The preliminary results that have been obtained suggest that the proposed algorithm is promising as an efficient method for solving MINLP problems belonging to the class considered in this paper. Because of the small number of iterations that is required, the proposed method should be particularly suited to MINLP problems where the NLP subproblems are expensive to solve.

Acknowledgments

The authors wish to thank the anonymous referees for their valuable suggestions and comments.

Support from CONACYT and Universidad Autonoma Metropolitana-I, Mexico, is gratefully acknowledged. This work was also supported in part by the National Science Foundation under grants CPE-8121665 and CPE-8351237.

References

[1] S. Albers and K. Brockhoff, "A procedure for new product positioning in an attribute space," *European Journal of Operational Research* 1 (1977) 230-238.

[2] E. Balas, "A duality theorem and an algorithm for (mixed-) integer nonlinear programming," *Linear Algebra and its Applications* 4 (1971) 341-352.

[3] E. Balas, "Disjunctive programming," in: P.L. Hammer, E.L. Johnson and B.H. Korte, eds., *Annals of Discrete Mathematics 5: Discrete Optimization II* (North-Holland, Amsterdam, 1979) pp. 3-51.

[4] E. Balas and R. Jeroslow, "Canonical Cuts on the Unit Hypercube," *SIAM Journal of Applied Mathematics* 23 (1972) 61-79.

[5] J.F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik* 4 (1962) 238-252.

[6] D.P. Bertsekas, G.S. Lauer, N.R. Sandell, Jr., and T.A. Posbergh, "Optimal short-term scheduling of large-scale power systems," *IEEE Transactions on Automatic Control* AC-28 (1983) 1-11.

[7] J.A. Bloom, "Mathematical generation planning models using decomposition and probabilistic simulation," Special Report No. EA-2566-SR, Electric Power Research Institute, Palo Alto, California (August 1982).

[8] J.A. Bloom, "Solving an electricity generating capacity expansion planning problem by generalized Benders' decomposition," *Operations Research* 31 (1983) 84-100.

[9] A.L. Brearley, G. Mitra and H.P. Williams, "Analysis of mathematical programming problems prior to applying the simplex algorithm," *Mathematical Programming* 8 (1975) 54-83.

[10] H. Crowder, E.L. Johnson and M.W. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research* 31 (1983) 803-834.

[11] M.A. Duran, "A mixed-integer nonlinear programming approach for the systematic synthesis of engineering systems," Ph.D. Thesis, Department of Chemical Engineering, Carnegie-Mellon University (Pittsburgh, PA, December 1984).

[12] M.A. Duran and I.E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs. The relation with generalized Benders decomposition," Technical Report DRC-06-68-84, Design Research Center, Carnegie-Mellon University (Pittsburgh, PA, December 1984).

[13] M. A. Duran and I. E. Grossmann, "A mixed-integer nonlinear programming algorithm for process systems synthesis," *American Institute of Chemical Engineers Journal*, 32 (1986) 592-606.

[14] B.C. Eaves and W.I. Zangwill, "Generalized cutting plane algorithms," *SIAM Journal on Control* 9 (1971) 529-542.

[15] J. Elzinga and T.G. Moore, "A central cutting plane algorithm for the convex programming problem," *Mathematical Programming* 8 (1975) 134-145.

[16] J.F. Faccenda and S. Debashish, "Computational experiments with a nonlinear 0-1 powerhouse optimization model," paper presented at the ORSA/TIMS Joint National Meeting, Orlando, FL (November 1983).

[17] R.S. Garfinkel and G.L. Nemhauser, *Integer Programming* (John Wiley & Sons, New York, 1972).

[18] B. Gavish, D. Horsky and K. Srikanth, "An approach to the optimal positioning of a new product," *Management Science* 29 (1983) 1277-1297.

[19] A.M. Geoffrion, "Elements of large-scale mathematical programming, Parts I and II," *Management Science* 16 (1970) 652-691.

[20] A.M. Geoffrion, "Generalized Benders decomposition," *Journal of Optimization Theory and Applications* 10 (1972) 237-260.

[21] F. Glover, "Surrogate constraints," *Operations Research* 16 (1968) 741-749.

[22] R.E. Gomory, "Some polyhedra related to combinatorial problems," *Linear Algebra and its Applications* 2 (1969) 451-558.

[23] C. Gonzaga and E. Polak, "On constraint dropping schemes and optimality functions for a class of outer approximations algorithms," *SIAM Journal on Control and Optimization* 17 (1979) 477-493.

[24] G.W. Graves, "Water pollution control," in: A.V. Balakrishnan, ed., *Techniques of Optimization* (Academic Press, New York, 1972) pp. 499-509.

[25] I.E. Grossmann "Mixed-integer programming approach for the synthesis of integrated process flow-sheets," *Computers and Chemical Engineering* 9 (1985) 463-482.

[26] G. Gunawardane, S. Hoff and L. Schrage, "Identification of special structure constraints in linear programs," *Mathematical Programming* 21 (1981) 90-97.

[27] O.K. Gupta, "Branch and bound experiments in nonlinear integer programming," Ph.D. Thesis, School of Industrial Engineering, Purdue University (West Lafayette, IN, December 1980).

[28] O.K. Gupta and A. Ravindran, "Nonlinear mixed integer programming and discrete optimization," in: R.W. Wayne and K.M. Ragsdell, eds., *Progress in Engineering Optimization* (ASME, New York, 1981) pp. 27-32.

[29] H.H. Hoang, "Topological optimization of networks: A nonlinear mixed integer model employing generalized Benders decomposition," *IEEE Transactions on Automatic Control* AC-27 (1982) 164-169.

[30] W.W. Hogan, "Optimization and convergence for extremal value functions arising from structured nonlinear programs," Working Paper No. 180, Western Management Science Institute, University of California, Los Angeles (September 1971).

[31] J.E. Kelley, Jr., "The cutting-plane method for solving convex programs," *Journal of the Society for Industrial and Applied Mathematics* 8 (1960) 703-712.

[32] G.S. Lauer, D.P. Bertsekas, N.R. Sandell, Jr., and T.A. Posbergh, "Solution of large-scale optimal unit commitment problems," *IEEE Transactions on Power Apparatus and Systems* PAS-101 (1982) 79-86.

[33] L.J. Leblanc, "An algorithm for the discrete network design problem," *Transportation Science* 9 (1975) 183-199.

[34] J.H. May, A.D. Shocker and D. Sudharshan, "A simulation comparison of methods for new product location," Working paper No. 82-126, Graduate School of Business, University of Pittsburgh, Pittsburgh (October 1983).

[35] D.Q. Mayne, E. Polak and R. Trahan, "An outer approximations algorithm for computer-aided design problems," *Journal of Optimization Theory and Applications* 28 (1979) 331-352.

[36] B.A. Murtagh and M.A. Saunders, "A projected lagrangian algorithm and its implementation for sparse nonlinear constraints, and MINOS/AUGMENTED User's Manual," Technical Reports SOL 80-1R and SOL 80-14, Systems Optimization Laboratory, Department of Operations Research, Stanford University, California (February 1981, June 1980).

[37] R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, NJ, 1970).

[38] N.R. Sandell, Jr., D.P. Bertsekas, J.J. Shaw, S.W. Gully and R. Gendron, "Optimal scheduling of large scale hydrothermal power systems," in: *Proceedings of the 1982 IEEE International Large Scale Systems Symposium* (Virginia Beach, Virginia, October 1982) pp. 141-147.

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

[40] J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimensions I* (Springer-Verlag, New York, 1970).

[41] J.A. Tomlin, "Large scale mathematical programming systems," *Computers and Chemical Engineering* 7 (1983) 575-582.

[42] D.M. Topkis, "Cutting-plane methods without nested constraint sets," *Operations Research* 18 (1970) 404-413.

[43] D.M. Topkis, "A note on cutting-plane methods without nested constraint sets," *Operations Research* 18 (1970) 1216-1220.

[44] D.M. Topkis, "A cutting-plane algorithm with linear and geometric rates of convergence," *Journal of Optimization Theory and Applications* 36 (1982) 1-22.

[45] T.J. Van Roy and L.A. Wolsey, "Valid inequalities for mixed 0-1 programs," CORE Discussion Paper 8316 (March 1983).

[46] J.A. Vaselenak, I.E. Grossmann and A.W. Westerberg, "Optimal retrofit design of multi-product batch plants," paper 50d presented at the American Institute of Chemical Engineers 1986 National Meeting, Houston, TX (April 1986).

[47] F.S. Zufryden, "ZIPMAP — A zero-one integer programming model for market segmentation and product positioning," *Journal of the Operational Research Society* 30 (1979) 63-70.

Appendix I: On the lower bounds predicted by GBD and outer-approximation at the first iteration

Theorem A. *Given is an initial integer value $y^1 \in U \cap V$ in problem P. If at y^1 the lower bounds z_1^{OA}, z_1^{GB}, are computed from the relaxed master programs of the outer-approximation and GBD methods respectively, then $z_1^{\mathrm{OA}} \geq z_1^{\mathrm{GB}}$.*

Proof. Since $y^1 \in U \cap V$ and Slater's constraint qualification is assumed to hold, the stationary conditions of the lagrangian of $S(y^1)$ yield (see proof Theorem 2),

$$\nabla f(x^1) + \nabla g(x^1)u^1 + A_1^{\mathrm{T}} v^1 = 0 \tag{A1}$$

where x^1 and $u^1 \geq 0$, $v^1 \geq 0$, are the optimal solution and multipliers of $S(y^1)$.

The feasible integer space in the relaxed master program of the outer-approximation method is given by

$$V_{\mathrm{OA}}^1 = \{y \colon y \in U,\ F_{\mathrm{OA}}^1(y) \text{ is non-empty}\}$$

where the feasible continuous space $F_{\mathrm{OA}}^1(y)$, $y \in V_{\mathrm{OA}}^1$, is given by

$$F_{\mathrm{OA}}^1(y) = \{x \colon x \in X,\ g(x^1) + \nabla g(x^1)(x - x^1) + By \leq 0\}.$$

It then follows that for every $y \in V^1_{OA}$, and every $x \in F^1_{OA}(y)$, the following inequalities hold,

$$g(x^1)+\nabla g(x^1)^T(x-x^1)+By \leq 0, \qquad A_1 x - a_1 \leq 0.$$

Since u^1, v^1, are non-negative,

$$(u^1)^T \nabla g(x^1)^T(x-x^1) \leq -(u^1)^T[g(x^1)+By],$$

$$(v^1)^T A_1(x-x^1) \leq -(v^1)^T[A_1 x^1 - a_1].$$

Multiplying (A1) by $(x-x^1)$ and substituting the second and third term of this equation by the two above inequalities yields

$$(u^1)^T[g(x^1)+By]+(v^1)^T[A_1 x^1 - a_1] \leq \nabla f(x^1)^T(x-x^1).$$

Since $(v^1)^T[A_1 x^1 - a_1]=0$, the above inequality also implies

$$z_1^{GB}(y)=c^T y+f(x^1)+(u^1)^T[g(x^1)+By] \leq z_1^{OA}(y, x)$$
$$= c^T y + f(x^1)+\nabla f(x^1)^T(x-x^1)$$

where $z_1^{GB}(y)$ and $z_1^{OA}(y, x)$ are the support functions in the relaxed master programs of the GBD and outer-approximation methods at the first iteration [11, 12]. Since the above inequality holds for every $x \in F^1_{OA}(y)$ at $y \in V^1_{OA}$, it follows that

$$z_1^{GB}(y) \leq \min z_1^{OA}(y, x), \quad x \in F^1_{OA}(y)$$

Therefore, since $V^1_{OA} \subseteq U$,

$$z_1^{GB} = \min_{y \in U} z_1^{GB}(y) \leq \min_{y \in V^1_{OA}} z_1^{GB}(y) \leq z_1^{OA} = \min_{y \in V^1_{OA} x \in F^1_{OA}(y)} z_1^{OA}(y, x).$$

Hence, given an initial feasible integer value at the first iteration, the lower bound predicted by the outer-approximation method is greater or equal than the lower bound predicted by GBD. □

Appendix II: Formulations of the test problems

Problems 1, 2 and 3

The first three example problems in the set are related to the problem of synthesizing a processing system [11, 13, 25]. This problem is the one of simultaneously determining the optimal structural and operating parameters for a process so as to satisfy given design specifications. The first step in the MINLP approach to the synthesis problem is to propose a superstructure that has embedded the competitive alternative process configurations to be considered (see Fig. 1A for the third example problem). For the formulation, a 0-1 variable (y) is associated with each process unit (piece of equipment) to denote its potential existence in the final optimal

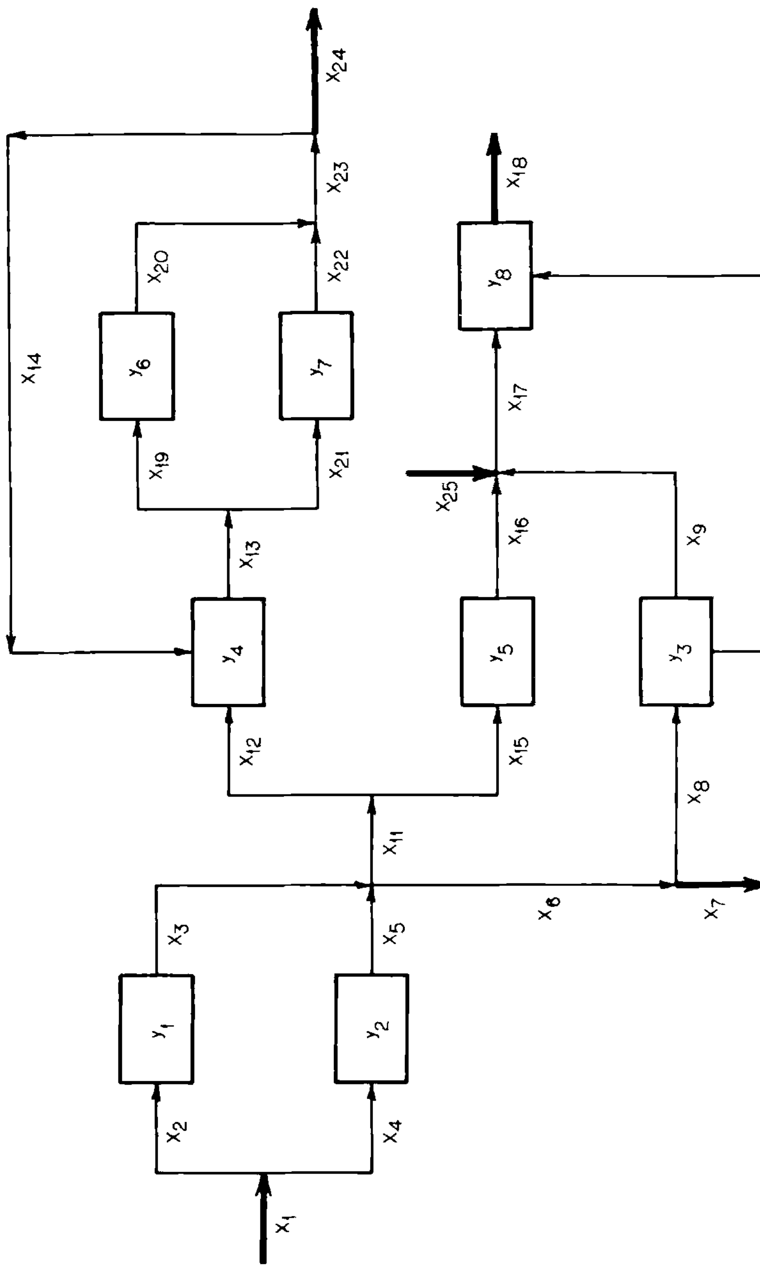

Fig. 1A. Test problem No. 3, superstructure.

configuration. The continuous variables (x) represent process parameters such as flowrates of material. The nonlinearities in the model are mainly due to intrinsic nonlinear input–output performance equations for some of the process units. The constraints in the underlying MINLP program are therefore design specifications, topological considerations, and conservation equations around nodes and units in the superstructure. The objective to be minimized is usually chosen as the annual cost, including both investment and operating costs. Fixed-cost charge approximations are considered for the investment cost part. See references [11, 13, 25] for a detailed discussion of the formulation of the problem.

The final form of the MINLP formulation for the third example problem is given in Table 3A. To illustrate the type of results that are expected for a process synthesis problem, the optimal process configuration for the third example problem is shown in Fig. 2A. This structure corresponds to the solution:

$$y^* = \{y_j:\ j = 1, \ldots, 8\} = (0, 1, 0, 1, 0, 1, 0, 1),$$

$$x^* = \{x_i:\ i = 3, 5, 10, 17, 19, 21, 9, 14, 25\}$$

$$= (0, 2, 0.46784, 0.58480, 2, 0, 0, 0.26667, 0.58480),$$

and objective function $z^* = 68.0097$. The starting guess was $y^1 = \{y_j^1:\ j = 1, \ldots, 8\} = (1, 0, 0, 1, 0, 1, 0, 0)$.

Problem No. 4

The fourth example problem was intended as a modest extension of the MINLP approach to the problem of determining the optimal positioning of a new product in a multiattribute space. An overview of the problem is presented next. See references

Table 1A
Test problem No. 1

minimize	$z = 5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_6 - 18\ln(x_2 + 1) - 19.2\ln(x_1 - x_2 + 1) + 10$
subject to	$0.8\ln(x_2 + 1) + 0.96\ln(x_1 - x_2 + 1) - 0.8x_6 \geq 0$
	$x_2 - x_1 \leq 0$
	$x_2 - Uy_1 \leq 0$
	$x_1 - x_2 - Uy_2 \leq 0$
	$\ln(x_2 + 1) + 1.2\ln(x_1 - x_2 + 1) - x_6 - Uy_3 \geq -2$
	$y_1 + y_2 \leq 1$
	$y \in \{0, 1\}^3,\ a \leq x \leq b,\ x = (x_j: j = 1, 2, 6) \in R^3$
	$a^T = (0, 0, 0),\ b^T = (2, 2, 1),\ U = 2$
initial guess	$y^1 = (1, 0, 1)$
solution:	$y^* = (0, 1, 0),\ x^* = (x_1, x_2, x_6) = (1.30097, 0, 1),\ z^* = 6.00972$

Table 2A

Test problem No. 2

minimize $z = 5y_1 + 8y_2 + 6y_3 + 10y_4 + 6y_5 - 10x_3 - 15x_5 - 15x_9 + 15x_{11} + 5x_{13} - 20x_{16}$

$+ \exp(x_3) + \exp(x_5/1.2) - 60\ln(x_{11} + x_{13} + 1) + 140$

subject to

$$-\ln(x_{11} + x_{13} + 1) \leq 0$$
$$-x_3 - x_5 - 2x_9 + x_{11} + 2x_{16} \leq 0$$
$$-x_3 - x_5 - 0.75x_9 + x_{11} + 2x_{16} \leq 0$$
$$x_9 - x_{16} \leq 0$$
$$2x_9 - x_{11} - 2x_{16} \leq 0$$
$$-0.5x_{11} + x_{13} \leq 0$$
$$0.2x_{11} - x_{13} \leq 0$$
$$\exp(x_3) - Uy_1 \leq 1$$
$$\exp(x_5/1.2) - Uy_2 \leq 1$$
$$1.25x_9 - Uy_3 \leq 0$$
$$x_{11} + x_{13} - Uy_4 \leq 0$$
$$-2x_9 + 2x_{16} - Uy_5 \leq 0$$
$$y_1 + y_2 = 1$$
$$y_4 + y_5 \leq 1$$
$$y \in \{0, 1\}^5, \quad a \leq x \leq b, \quad x = (x_j : j = 3, 5, 9, 11, 13, 16) \in R^6$$
$$a^{\mathrm{T}} = (0, 0, 0, 0, 0, 0), \quad b^{\mathrm{T}} = (2, 2, 2, -, -, 3), \quad U = 10$$

initial guess $y^1 = (1, 0, 0, 0, 0)$

solution: $y^* = (0, 1, 1, 1; 0)$

$x^* = (x_3, x_5, x_{11}, x_{13}, x_9, x_{16}) = (0, 2, 0.65201, 0.32601, 1.07839, 1.07839), \quad z^* = 73.0353$

[1, 18, 34, 47] for a detailed description. Consider a market with a set of already existing products (M) (e.g. different brands of personal computers), and a set of consumers (N). Assume that existing products can be located in a multiattribute space (dimension K) according to coordinates δ_{jk}, $j = 1, \ldots, M$, $k = 1, \ldots, K$. Assume also that each consumer can be characterized in terms of an ideal point z_{ik}, and a set of attribute weights w_{ik}, $i = 1, \ldots, N$, $k = 1, \ldots, K$, both representing his/her concept of an ideal product. Further, a region (hyperellipsoid) defining closeness to the ideal point for each consumer can readily be determined in terms of the existing products. Based on the above preference definition, a consumer will obviously select a product which is closest to his/her ideal point. The objective in the problem as presented above is then to optimally design a new product (x_k, $k = 1, \ldots, K$) so as to attract the largest number of consumers. The scope of the optimal positioning problem can be extended if data are given for the revenue of the firm

Table 3A
Test problem No. 3

minimize $z = 5y_1 + 8y_2 + 6y_3 + 10y_4 + 6y_5 + 7y_6 + 4y_7 + 5y_8 - 10x_3 - 15x_5 + 15x_{10} + 80x_{17} + 25x_{19} + 35x_{21}$
$- 40x_9 + 15x_{14} - 35x_{25} + \exp(x_3) + \exp(x_5/1.2) - 65\ln(x_{10} + x_{17} + 1)$
$- 90\ln(x_{19} + 1) - 80\ln(x_{21} + 1) + 120$

subject to

$$-1.5\ln(x_{19} + 1) - \ln(x_{21} + 1) - x_{14} \le 0$$
$$-\ln(x_{10} + x_{17} + 1) \le 0$$
$$-x_3 - x_5 + x_{10} + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 0.5x_9 - x_{14} - 2x_{25} \le 0$$
$$-x_3 - x_5 + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 2x_9 - x_{14} - 2x_{25} \le 0$$
$$-2x_{17} - 0.8x_{19} - 0.8x_{21} + 2x_9 + x_{14} + 2x_{25} \le 0$$
$$-0.8x_{19} - 0.8x_{21} + x_{14} \le 0$$
$$-x_{17} + x_9 + x_{25} \le 0$$
$$-0.4x_{19} - 0.4x_{21} + 1.5x_{14} \le 0$$
$$0.16x_{19} + 0.16x_{21} - 1.2x_{14} \le 0$$
$$x_{10} - 0.8x_{17} \le 0$$
$$-x_{10} + 0.4x_{17} \le 0$$
$$\exp(x_3) - Uy_1 \le 1$$
$$\exp(x_5/1.2) - Uy_2 \le 1$$
$$x_9 - Uy_3 \le 0$$
$$0.8x_{19} + 0.8x_{21} - Uy_4 \le 0$$
$$2x_{17} - 2x_9 - 2x_{25} - Uy_5 \le 0$$
$$x_{19} - Uy_6 \le 0$$
$$x_{21} - Uy_7 \le 0$$
$$x_{10} + x_{17} - Uy_8 \le 0$$
$$y_1 + y_2 = 1, \quad y_4 + y_5 \le 1$$
$$-y_4 + y_6 + y_7 = 0$$
$$y_3 - y_8 \le 0$$
$$y \in \{0, 1\}^8, \quad a \le x \le b, \quad x = (x_j\colon j = 3, 5, 10, 17, 19, 21, 9, 14, 25) \in R^9$$
$$a^T = (0, 0, 0, 0, 0, 0, 0, 0, 0), \quad b^T = (2, 2, 1, 2, 2, 2, 2, 1, 3), \quad U = 10$$

from the new product sales to consumer i (c_i), as well as a function f for representing the cost of reaching locations of the new product within an attribute space defined by a set of constraints $X = \{x\colon Ax \le b,\ lb \le x \le ub\}$. Under this new definition of the problem, the objective for the firm could then be the maximization of profits. Thus, if a 0–1 variable (y_i) is associated with each consumer to denote whether or

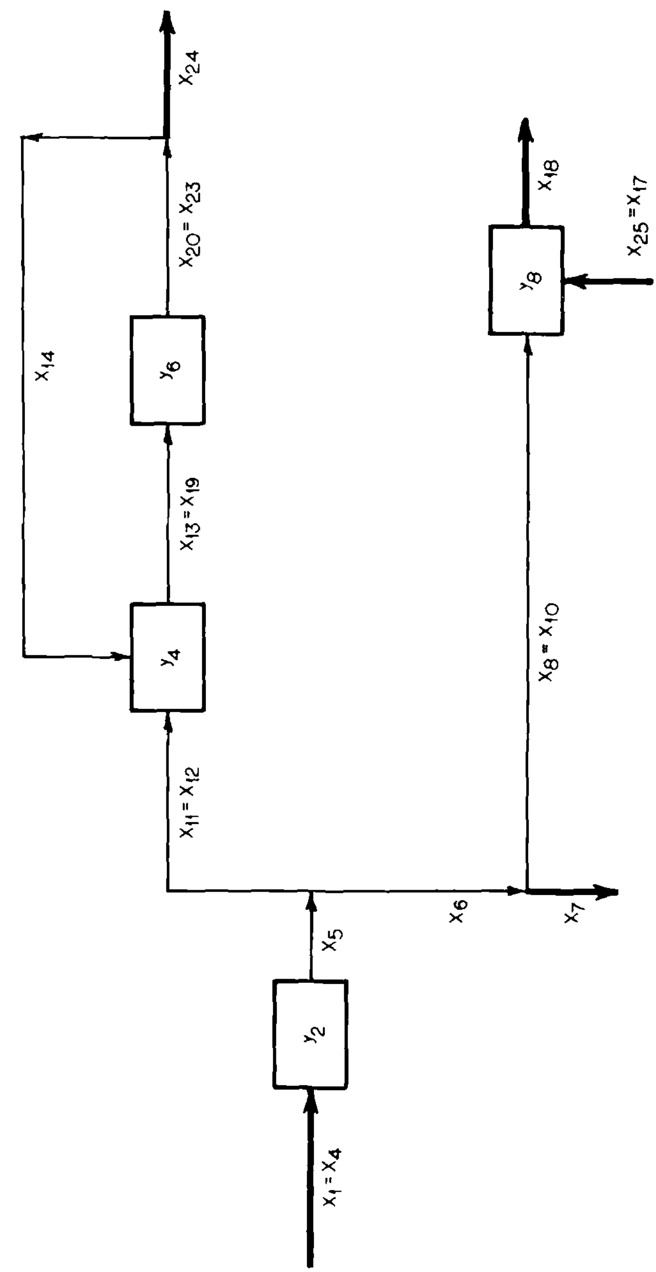

Fig. 2A. Test problem No. 3, optimal configuration.

not he/she is attracted by the new product, the formulation of the problem leads to a MINLP program of the class addressed in this paper. The formulation of the fourth example problem is given in Table 4A. The data for the coordinates of existing products (δ_{jk}) and ideal points (z_{ik}), and attribute weights (w_{ik}) are from the 10 existing products, 25 consumers, 5 attributes problem in Gavish et al., [18].

Table 4A

Test problem No. 4

maximize $z = \sum_{i=1}^{25} c_i y_i + f(x)$

subject to $\sum_{k=1}^{K=5} w_{ik}(x_k - z_{ik})^2 - (1 - y_i)H \le R_i^2, \quad i = 1, \ldots, N = 25$

$$x_1 - x_2 + x_3 + x_4 + x_5 \le 10$$

$$0.6x_1 - 0.9x_2 - 0.5x_3 + 0.1x_4 + x_5 \le -0.64$$

$$x_1 - x_2 + x_3 - x_4 + x_5 \ge 0.69$$

$$0.157x_1 + 0.05x_2 \le 1.5$$

$$0.25x_2 + 1.05x_4 - 0.3x_5 \ge 4.5$$

$$2.0 \le x_1 \le 4.5$$

$$0.0 \le x_2 \le 8.0$$

$$3.0 \le x_3 \le 9.0$$

$$0.0 \le x_4 \le 5.0$$

$$4.0 \le x_5 \le 10$$

$$y_i \in \{0, 1\}, \ i = 1, \ldots, N = 25, \ x \in R^5$$

where

$$f(x) = -0.6x_1^2 + 0.9x_2 + 0.5x_3 - 0.1x_4^2 - x_5$$

$$H = 1000, \qquad R_i^2 = \min_{j=1,\ldots,M=10} \left\{ \sum_{k=1}^{K=5} w_{ik}(\delta_{jk} - z_{ik})^2 \right\}, \quad i = 1, \ldots, N = 25$$

$c^T = [1, 0.2, 1, 0.2, 0.9, 0.9, 0.1, 0.8, 1, 0.4, 1, 0.3, 0.1, 0.3, 0.5, 0.9, 0.8, 0.1, 0.9, 1, 1, 1, 0.2, 0.7, 0.7]$

Solution: $y^* = [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1]$

$x^* = [2.0, 7.58132, 7.95783, 3.62350, 4.0], \quad z^* = 7.78913$

Table 5A

Data for test problem No. 4

i	Ideal points (z_{ik}) $k=1$	2	3	4	5
1	2.26	5.15	4.03	1.74	4.74
2	5.51	9.01	3.84	1.47	9.92
3	4.06	1.80	0.71	9.09	8.13
4	6.30	0.11	4.08	7.29	4.24
5	2.81	1.65	8.08	3.99	3.51
6	4.29	9.49	2.24	9.78	1.52
7	9.76	3.64	6.62	3.66	9.08
8	1.37	6.99	7.19	3.03	3.39
9	8.89	8.29	6.05	7.48	4.09
10	7.42	4.60	0.30	0.97	8.77
11	1.54	7.06	0.01	1.23	3.11
12	7.74	4.40	7.93	5.95	4.88
13	9.94	5.21	8.58	0.13	4.57
14	9.54	1.57	9.66	5.24	7.90
15	7.46	8.81	1.67	6.47	1.81
16	0.56	8.10	0.19	6.11	6.40
17	3.86	6.68	6.42	7.29	4.66
18	2.98	2.98	3.03	0.02	0.67
19	3.61	7.62	1.79	7.80	9.81
20	5.68	4.24	4.17	6.75	1.08
21	5.48	3.74	3.34	6.22	7.94
22	8.13	8.72	3.93	8.80	8.56
23	1.37	0.54	1.55	5.56	5.85
24	8.79	5.04	4.83	6.94	0.38
25	2.66	4.19	6.49	8.04	1.66

i	Attribute weights (w_{ik}) $k=1$	2	3	4	5
1	9.57	2.74	9.75	3.96	8.67
2	8.38	3.93	5.18	5.20	7.82
3	9.81	0.04	4.21	7.38	4.11
4	7.41	6.08	5.46	4.86	1.48
5	9.96	9.13	2.95	8.25	3.58
6	9.39	4.27	5.09	1.81	7.58
7	1.88	7.20	6.65	1.74	2.86
8	4.01	2.67	4.86	2.55	6.91
9	4.18	1.92	2.60	7.15	2.86
10	7.81	2.14	9.63	7.61	9.17
11	8.96	3.47	5.49	4.73	9.43
12	9.94	1.63	1.23	4.33	7.08
13	0.31	5.00	0.16	2.52	3.08
14	6.02	0.92	7.47	9.74	1.76
15	5.06	4.52	1.89	1.22	9.05
16	5.92	2.56	7.74	6.96	5.18
17	6.45	1.52	0.06	5.34	8.47
18	1.04	1.36	5.99	8.10	5.22
19	1.40	1.35	0.59	8.58	1.21
20	6.68	9.48	1.60	6.74	8.92
21	1.95	0.46	2.90	1.79	0.99
22	5.18	5.10	8.81	3.27	9.63
23	1.47	5.71	6.95	1.42	3.49
24	5.40	3.12	5.37	6.10	3.71
25	6.32	0.81	6.12	6.73	7.93

j	Existing products (δ_{jk}) $k=1$	2	3	4	5
1	0.62	5.06	7.82	0.22	4.42
2	5.21	2.66	9.54	5.03	8.01
3	5.27	7.72	7.97	3.31	6.56
4	1.02	8.89	8.77	3.10	6.66
5	1.26	6.80	2.30	1.75	6.65
6	3.74	9.06	9.80	3.01	9.52
7	4.64	7.99	6.69	5.88	8.23
8	8.35	3.79	1.19	1.96	5.88
9	6.44	0.17	9.93	6.80	9.75
10	6.49	1.92	0.05	4.89	6.43