Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques

Ignacio E. Grossmann
Department of Chemical Engineering, Carnegie Mellon University
Pittsburgh, PA 15213, USA

June 2001/Rev. April 2002

ABSTRACT

This paper has as a major objective to present a unified overview and derivation of mixed-integer nonlinear programming (MINLP) techniques, Branch and Bound, Outer-Approximation, Generalized Benders and Extended Cutting Plane methods, as applied to nonlinear discrete optimization problems that are expressed in algebraic form. The solution of MINLP problems with convex functions is presented first, followed by a brief discussion on extensions for the nonconvex case. The solution of logic based representations, known as generalized disjunctive programs, is also described. Theoretical properties are presented, and numerical comparisons on a small process network problem.

Keywords: mixed-integer programming, disjunctive programming, nonlinear programming.

INTRODUCTION

Mixed-integer optimization provides a powerful framework for mathematically modeling many optimization problems that involve discrete and continuous variables. Over the last few years there has been a pronounced increase in the development of these models, particularly in process systems engineering (see Grossmann et al, 1996; Grossmann, 1996a; Grossmann and Daichendt, 1996; Pinto and Grossmann, 1998; Shah, 1998; Grossmann et al, 1999; Kallrath, 2000).

Mixed-integer linear programming (MILP) methods and codes have been available and applied to many practical problems for more than twenty years (e.g. see Nemhauser and Wolsey, 1988). The most common method is the LP-based branch and bound method (Dakin, 1965), which has been implemented in powerful codes such as OSL, CPLEX and XPRESS. Recent trends in MILP include the development of branch-and-price (Barnhart et al. 1998) and branch-and-cut methods such as the lift-and-project method by Balas, Ceria and Cornuejols (1993) in which cutting planes are generated as part of the branch and bound enumeration. See also Johnson et al. (2000) for a recent review on MILP.

It is not until recently that several new methods and codes are becoming available for mixed-integer nonlinear problems (MINLP) (Grossmann and Kravanja, 1997). In this paper we provide a review the various methods emphasizing a unified treatment for their derivation. As will be shown, the different methods can be derived from three basic NLP subproblems and from one cutting plane MILP problem, which essentially correspond to the basic subproblems of the Outer-Approximation method. Properties of the algorithms are first considered for the case when the nonlinear functions are convex in the discrete and continuous variables. Extensions are then presented for handling nonlinear equations and nonconvexities. Finally, the paper considers properties and algorithms of the recent logic-based representations for discrete/continuous optimization that are known as generalized disjunctive programs. Numerical results on a small example are presented comparing the various algorithms.

BASIC ELEMENTS OF MINLP METHODS

The most basic form of an MINLP problem when represented in algebraic form is as follows:

$$\begin{array}{ll} & \min Z = f(x,y) \\ s.t. & g_j(x,y) \le 0 \;\; j \in J \\ & x \in X, y \in Y \end{array} \qquad \text{(P1)}$$

where *f(·), g(·)* are *convex, differentiable* functions, J is the index set of inequalities, and x and y are the continuous and discrete variables, respectively. The set X is commonly assumed to be a convex compact set, e.g. $X = \{x \mid x \in \boldsymbol{R}^n, Dx \le d, x^L \le x \le x^U\}$; the discrete set Y corresponds to a polyhedral set of integer points, $Y = \{y \mid y \in \boldsymbol{Z}^m, Ay \le a\}$, which in most applications is restricted to 0-1 values, $y \in \{0,1\}^m$. In most applications of interest the objective and constraint functions *f(·), g(·)* are linear in y (e.g. fixed cost charges and mixed-logic constraints): $f(x,y) = c^T y + r(x),\; g(x,y) = By + h(x).$

Methods that have addressed the solution of problem (P1) include the branch and bound method (BB) (Gupta and Ravindran, 1985; Nabar and Schrage, 1991; Borchers and Mitchell, 1994; Stubbs and Mehrotra, 1999; Leyffer, 2001), Generalized Benders Decomposition (GBD) (Geoffrion, 1972), Outer-Approximation (OA) (Duran and Grossmann, 1986; Yuan et al., 1988; Fletcher and Leyffer, 1994), LP/NLP based branch and bound (Quesada and Grossmann, 1992), and Extended Cutting Plane Method (ECP) (Westerlund and Pettersson, 1995).

NLP Subproblems. There are three basic NLP subproblems that can be considered for problem (P1):

a) NLP relaxation

$$\begin{aligned} & \min Z_{LB}^k = f(x, y) \\ s.t.\ & g_j(x, y) \leq 0 \quad j \in J \\ & x \in X,\ y \in Y_R \qquad\qquad\qquad \text{(NLP1)} \\ & y_i \leq \alpha_i^k \ \ i \in I_{FL}^k \\ & y_i \geq \beta_i^k \ \ i \in I_{FU}^k \end{aligned}$$

where Y_R is the continuous relaxation of the set Y, and I_{FL}^k, I_{FU}^k are index subsets of the integer variables y_i, $i \in I$, which are restricted to lower and upper bounds, α_i^k, β_i^k, at the *k'th* step of a branch and bound enumeration procedure. It should be noted that $\alpha_i^k = \lfloor y_i^l \rfloor, \beta_i^k = \lceil y_i^m \rceil$, $l < k$, $m < k$ where y_i^l, y_i^m, are noninteger values at a previous step, and $\lfloor . \rfloor, \lceil . \rceil$, are the floor and ceiling functions, respectively.

Also note that if $I_{FU}^k = I_{FL}^k = \emptyset$ $(k=0)$, (NLP1) corresponds to the continuous NLP relaxation of (P1). Except for few and special cases, the solution to this problem yields in general a noninteger vector for the discrete variables. Problem (NLP1) also corresponds to the *k'th* step in a branch and bound search. The optimal objective function Z_{LB}^o provides an absolute lower bound to (P1); for $m \geq k$, the bound is only valid for $I_{FL}^k \subset I_{FL}^m$, $I_{FU}^k \subset I_{FL}^m$.

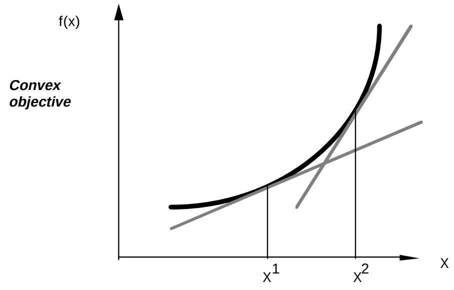
b) NLP subproblem for fixed y^k:

$$\begin{aligned} & \min Z_U^k = f(x, y^k) \\ s.t.\ & g_j(x, y^k) \leq 0 \quad j \in J \qquad\qquad \text{(NLP2)} \\ & x \in X \end{aligned}$$

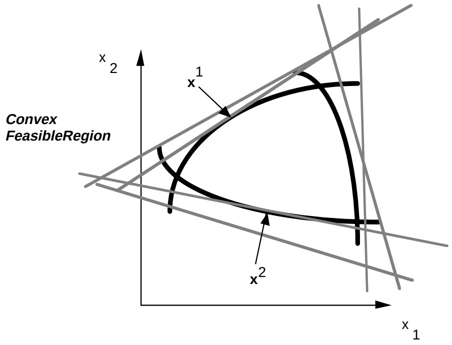
which yields an upper bound Z_U^k to (P1) provided (NLP2) has a feasible solution. When this is not the case, we consider the next subproblem:

c) Feasibility subproblem for fixed y^k.

$$\begin{aligned} & \min u \\ s.t.\ & g_j(x, y^k) \leq u \quad j \in J \qquad\qquad \text{(NLPF)} \\ & x \in X,\ u \in R^1 \end{aligned}$$



Underestimate Objective Function



Overestimate Feasible Region

Fig. 1. Geometrical interpretation of linearizations in master problem (M-MIP)

which can be interpreted as the minimization of the infinity-norm measure of infeasibility of the corresponding NLP subproblem. Note that for an infeasible subproblem the solution of (NLPF) yields a strictly positive value of the scalar variable u.

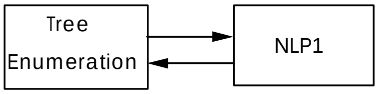
MILP cutting plane.

The convexity of the nonlinear functions is exploited by replacing them with supporting hyperplanes, that are generally, but not necessarily, derived at the solution of the NLP subproblems. In particular, the new values y^K (or (x^K, y^K)) are obtained from a cutting plane MILP problem that is based on the K points, (x^k, y^k), $k=1...K$ generated at the K previous steps:

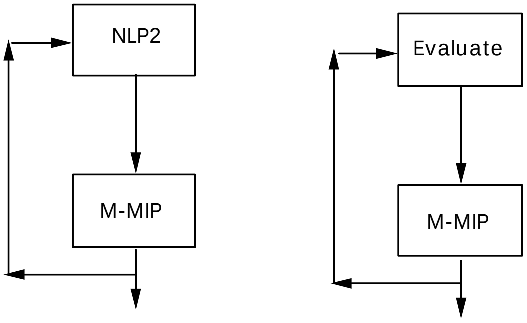
$$
\begin{aligned}
&\min Z_L^K = \alpha \\
st\quad &\left.\begin{aligned}
&\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
&g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \;\; j \in J^k
\end{aligned}\right\} k = 1,\ldots K \qquad \text{(M} - \text{MIP)} \\
&x \in X, y \in Y
\end{aligned}
$$

where $J^k \subseteq J$. When only a subset of linearizations is included, these commonly correspond to violated constraints in problem (P1). Alternatively, it is possible to include all linearizations in (M-MIP). The solution of (M-MIP) yields a valid lower bound Z_L^K to problem (P1). This bound is nondecreasing with the number of linearization points *K*. Note that since the functions *f(x,y)* and *g(x,y)* are convex, the linearizations in (M-MIP) correspond to outer-approximations of the nonlinear feasible region in problem (P1). A geometrical interpretation is shown in Fig.1, where it can be seen that the convex objective function is being underestimated, and the convex feasible region overestimated with these linearizations.

Algorithms. The different methods can be classified according to their use of the subproblems (NLP1), (NLP2) and (NLPF), and the specific specialization of the MILP problem (M-MIP) as seen in Fig. 2. It should be noted that in the GBD and OA methods (case (b)), and in the LP/NLP based branch and bound mehod (case (d)), the problem (NLPF) is solved if infeasible subproblems are found. Each of the methods is explained next in terms of the basic subproblems.

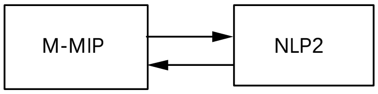


(a) Branch and bound



(b) GBD, OA

(c) ECP



(d) LP/NLP based branch and bound

Fig. 2. Major Steps In the Different Algorithms

I. Branch and Bound. While the earlier work in branch and bound (BB) was aimed at linear problems (Dakin, 1965), this method can also be applied to nonlinear problems (Gupta and Ravindran, 1985; Nabar and Schrage, 1991; Borchers and Mitchell, 1994; Stubbs and Mehrotra, 1999; Leyffer, 2001). The BB method starts by solving first the continuous NLP relaxation. If all discrete variables take integer values the search is stopped. Otherwise, a tree search is performed in the space of the integer variables $y_i, i \in I$. These are successively fixed at the corresponding nodes of the tree, giving rise to relaxed NLP subproblems of the form (NLP1) which yield lower bounds for the subproblems in the descendant nodes. Fathoming of nodes occurs when the lower bound exceeds the current upper bound, when the subproblem is infeasible or when all integer variables y_i take on discrete values. The latter yields an upper bound to the original problem.

The BB method is generally only attractive if the NLP subproblems are relatively inexpensive to solve, or when only few of them need to be solved. This could be either because of the low dimensionality of the discrete variables, or because the integrality gap of the continuous NLP relaxation of (P1) is small.

II. Outer-Approximation (Duran and Grossmann, 1986; Yuan et al., 1988; Fletcher and Leyffer, 1994). The OA method arises when NLP subproblems (NLP2) and MILP master problems (M-MIP) with $J^k = J$ are solved successively in a cycle of iterations to generate the points (x^k, y^k). For its derivation, the OA algorithm is based on the following theorem (Duran and Grossmann, 1986):

Theorem 1. *Problem (P) and the following MILP master problem (M-OA) have the same optimal solution (x*, y*),*

$$\begin{aligned}
&\min Z_L = \alpha \\
&st \quad \left.\begin{aligned} &\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ &g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \quad j \in J \end{aligned}\right\} k \in K^* \qquad \text{(M - OA)} \\
&x \in X, y \in Y
\end{aligned}$$

where K*={$k \mid$ for all feasible $y^k \in Y$, (x^k, y^k) is the optimal solution to the problem (NLP2), and for all infeasible $y^k \in Y$, (x^k, y^k) is the optimal solution to the problem (NLPF)}

Since the master problem (M-OA) requires the solution of all feasible discrete variables y^k, the following MILP relaxation is considered, assuming that the solution of K different NLP subproblems (K =|KFS$\cup$ KIS|, KFS set of solutions from NLP2, KIS set of solutions from NLPF) is available:

$$\begin{aligned}
&\min Z_L^K = \alpha \\
&st \quad \left.\begin{aligned} &\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ &g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \quad j \in J \end{aligned}\right\} k = 1,...K \qquad \text{(RM - OA)} \\
&x \in X, y \in Y
\end{aligned}$$

Given the assumption on convexity of the functions *f(x,y)* and *g(x,y)*, the following property can easily be established,

Property 1. *The solution of problem (RM-OA),* Z_L^K, *corresponds to a lower bound of the solution of problem (P1).*

Note that this property can be verified in Fig. 1. Also, since function linearizations are accumulated as iterations proceed, the master problems (RM-OA) yield a non-decreasing sequence of lower bounds, $Z_L^1 \ldots \leq Z_L^k \leq \ldots \leq Z_L^K$, since linearizations are accumulated as iterations k proceed.

The OA algorithm as proposed by Duran and Grossmann (1986) consists of performing a cycle of major iterations, k=1,..K, in which (NLP1) is solved for the corresponding y^k, and the relaxed MILP master problem (RM-OA) is updated and solved with the corresponding function linearizations at the point (x^k,y^k), for which the corresponding subproblem NLP2 is solved. If feasible, the solution to that problem is used to construct the first MILP master problem; otherwise a feasibility problem (NLPF) is solved to generate the corresponding continuous point (Fletcher and Leyffer, 1994). The initial MILP master problem (RM-OA) then generates a new vector of discrete variables. The (NLP2) subproblems yield an upper bound that is used to define the best current solution, $UB^K = \min_k \{Z_U^k\}$. The cycle of iterations is continued until this upper bound and the lower bound of the relaxed master problem, Z_L^K, are within a specified tolerance. One way to avoid solving the feasibility problem (NLPF) in the OA algorithm when the discrete variables in problem (P1) are 0-1, is to introduce the following integer cut whose objective is to make infeasible the choice of the previous 0-1 values generated at the K previous iterations (Duran and Grossmann, 1986):

$$\sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \leq \left| B^k \right| - 1 \quad k = 1, \ldots K \qquad \text{(ICUT)}$$

where $B^k=\{\ i\ |\ y_i^k\ =\ 1\}$, $N^k=\{\ i\ |\ y_i^k\ =\ 0\}$, k=1,...K. This cut becomes very weak as the dimensionality of the 0-1 variables increases. However, it has the useful feature of ensuring that new 0-1 values are generated at each major iteration. In this way the algorithm will not return to a previous integer point when convergence is achieved. Using the above integer cut the termination takes place as soon as $Z_L^K \geq UB^K$.

The OA method generally requires relatively few cycles or major iterations. One reason for this behavior is given by the following property:

Property 2. *The OA algorithm trivially converges in one iteration if f(x,y) and g(x,y) are linear.*

This property simply follows from the fact that if *f(x,y)* and *g(x,y)* are linear in *x* and *y* the MILP master problem (RM-OA) is identical to the original problem (P1).

It is also important to note that the MILP master problem need not be solved to optimality. In fact given the upper bound UB^K and a tolerance ε, it is sufficient to generate the new (y^K, x^K) by solving,

$$
\begin{aligned}
&\min Z_L^K = 0\alpha \\
&s.t.\ \alpha \geq UB^K - \varepsilon \\
&\left.\begin{aligned}
&\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
&g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \quad j \in J
\end{aligned}\right\} k = 1,..K \qquad \text{(RM - OAF)} \\
&x \in X, y \in Y
\end{aligned}
$$

While in (RM-OA) the interpretation of the new point y^K is that it represents the best integer solution to the approximating master problem, in (RM-OAF) it represents an integer solution whose lower bounding objective does not exceed the current upper bound, UB^K; in other words it is a feasible solution to (RM-OA) with an objective below the current estimate. Note that the OA iterations are terminated when (RM-OAF) has no feasible solution.

III. Generalized Benders Decomposition (Geoffrion, 1972). The GBD method (see Flippo and Kan 1993) is similar to the Outer-Approximation method. The difference arises in the definition of the MILP master problem (M-MIP). In the GBD method only active inequalities are considered $J^k = \{j \mid g_j(x^k, y^k) = 0\}$ and the set $x \in X$ is disregarded. In particular, consider an outer-approximation given at a given point (x^k, y^k),

$$
\begin{aligned}
&\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
&g(x^k, y^k) + \nabla g\ (x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0
\end{aligned}
\qquad (\mathrm{OA}^k)
$$

where for a fixed y^k the point x^k corresponds to the optimal solution to problem (NLP2). Making use of the Karush-Kuhn-Tucker conditions and eliminating the continuous variables x, the inequalities in (OA^k) can be reduced as follows (Quesada and Grossmann (1992):

$$
\alpha \geq f(x^k, y^k) + \nabla_y f(x^k, y^k)^T \left(y - y^k\right) + \left(\mu^k\right)^T \left[g(x^k, y^k) + \nabla_y g\ (x^k, y^k)^T \left(y - y^k\right)\right] \qquad (\mathrm{LC}^k)
$$

which is the Lagrangian cut projected in the y-space. This can be interpreted as a surrogate constraint of the equations in (OA^k), because it is obtained as a linear combination of these.

For the case when there is no feasible solution to problem (NLP2), then if the point x^k is obtained from the feasibility subproblem (NLPF), the following feasibility cut projected in y can be obtained using a similar procedure,

$$\left(\lambda^k\right)^T \left[g(x^k, y^k) + \nabla_y g\ (x^k, y^k)^T \left(y - y^k\right)\right] \le 0 \qquad \text{(FC}^k\text{)}$$

In this way, the problem (M-MIP) reduces to a problem projected in the y-space:

$$\begin{aligned} &\min Z_L^K = \alpha \\ &st\ \alpha \ge\ f(x^k, y^k) + \nabla_y f(x^k, y^k)^T \left(y - y^k\right) \qquad \text{(RM}-\text{GBD)} \\ &\quad + \left(\mu^k\right)^T \left[g(x^k, y^k) + \nabla_y g\ (x^k, y^k)^T \left(y - y^k\right)\right]\ \ k \in KFS \\ \\ &\left(\lambda^k\right)^T \left[g(x^k, y^k) + \nabla_y g\ (x^k, y^k)^T \left(y - y^k\right)\right] \le 0\ \ \ k \in KIS \\ &\quad x \in X, \alpha \in R^1 \end{aligned}$$

where *KFS* is the set of feasible subproblems (NLP2) and *KIS* the set of infeasible subproblems whose solution is given by (NLPF). Also $|KFS \cup\ KIS\ | = K$. Since the master problem (RM-GBD) can be derived from the master problem (RM-OA), in the context of problem (P1), Generalized Benders decomposition can be regarded as a particular case of the Outer-Approximation algorithm. In fact the following property, holds between the two methods (Duran and Grossmann, 1986):

Property 3 *Given the same set of K subproblems, the lower bound predicted by the relaxed master problem (RM-OA) is greater or equal to the one predicted by the relaxed master problem (RM-GBD).*

The above proof follows from the fact that the Lagrangian and feasibility cuts, (LC^k) and (FC^k), are surrogates of the outer-approximations (OA^k). Given the fact that the lower bounds of GBD are generally weaker, this method commonly requires a larger number of cycles or major iterations. As the number of 0-1 variables increases this difference becomes more pronounced. This is to be expected since only one new cut is generated per iteration. Therefore, user-supplied constraints must often be added to the master problem to strengthen the bounds. Also, it is sometimes possible to generate multiple cuts from the solution of an NLP subproblem in order to strengthen the lower bound (Magnanti and Wong, 1981). As for the OA algorithm, the trade-off is that while it generally predicts stronger lower bounds than GBD, the computational cost for solving the master problem (M-OA) is greater since the number of constraints added per iteration is equal to the number of nonlinear constraints plus the nonlinear objective.

The following convergence property applies to the GBD method (Sahinidis and Grossmann, 1991):

Property 4. *If problem (P1) has zero integrality gap, the GBD algorithm converges in one iteration once the optimal* (x^*, y^*) *is found.*

The above property implies that the only case one can expect the GBD method to terminate in one iteration, is when the initial discrete vector is the optimum, and when the objective value of the NLP relaxation of problem (P1) is the same as the objective of the optimal mixed-integer solution. Given the relationship of GBD with the OA algorithm, Property 4 is also inherited by the OA method.

One further property that relates the OA and GBD algorithms is the following (Türkay and Grossmann, 1996):

Property 5. *The cut obtained from performing one Benders iteration on the MILP master (RM-OA) is equivalent to the cut obtained from the GBD algorithm.*

By making use of this property, instead of solving the MILP (RM-OA) to optimality, for instance by LP-based branch and bound, one can generate a GBD cut by simply performing one Benders (1962) iteration on the MILP. This property will prove to be useful when deriving a logic-based version of the GBD algorithm as will be discussed later in the paper.

IV. Extended Cutting Plane (Westerlund and Pettersson, 1995). The ECP method, which is an extension of Kelly's cutting plane algorithm for convex NLP (Kelley, 1960), does not rely on the use of NLP subproblems and algorithms. It relies only on the iterative solution of the problem (M-MIP) by successively adding a linearization of the most violated constraint at the predicted point (x^k,y^k) : $J^k = \{ \hat{j} \in \arg\{ \max_{j \in J} g_j(x^k, y^k) \}$ Convergence is achieved when the maximum constraint violation lies within the specified tolerance. The optimal objective value of (M-MIP) yields a non-decreasing sequence of lower bounds. It is of course also possible to either add to (M-MIP) linearizatons of all the violated constraints in the set J^k , or linearizations of all the nonlinear constraints $j \in J$. In the ECP method the objective must be defined as a linear function, which can easily be accomplished by introducing a new variable to transfer nonlinearities in the objective as an inequality.

Note that since the discrete and continuous variables are converged simultaneously, the ECP method may require a large number of iterations. However, this method shares with the OA method Property 2 for the limiting case when all the functions are linear.

V. LP/NLP based Branch and Bound (Quesada and Grossmann, 1992). This method is similar in spirit to a branch and cut method, and avoids the complete solution of the MILP master problem (M-OA) at each major iteration. The method starts by solving an initial NLP subproblem, which is linearized as in (M-OA). The basic idea consists then of performing an LP-based branch and bound method for (M-OA) in which NLP subproblems (NLP2) are solved at those nodes in which feasible integer solutions are found. By updating the representation of the master problem in the current open nodes of the tree with the addition of the corresponding linearizations, the need of restarting the tree search is avoided.

This method can also be applied to the GBD and ECP methods. The LP/NLP method commonly reduces quite significantly the number of nodes to be enumerated. The trade-off, however, is that the number of NLP subproblems may increase. Computational experience has indicated that often the number of NLP subproblem remains unchanged. Therefore, this method is better suited for problems in which the bottleneck corresponds to the solution of the MILP master problem. Leyffer (1993) has reported substantial savings with this method.

EXTENSIONS OF MINLP METHODS

In this section we present an overview of some of the major extensions of the methods presented in the previous section.

Quadratic Master Problems. For most problems of interest, problem (P1) is linear in y: $f(x,y) = \phi(x) + c^T y$, $g(x,y) = h(x) + By$. When this is not the case Fletcher and Leyffer (1994) suggested to include a quadratic approximation to (RM-OAF) of the form:

$$\begin{aligned} &\min\ Z^K = \alpha + \frac{1}{2}\begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}^T \nabla^2 L(x^k, y^k) \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \\ &s.t. \quad \alpha \leq UB^K - \varepsilon \end{aligned} \qquad \text{(M-MIQP)}$$

$$\left.\begin{array}{l} \alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ g(x^k, y^k) + \nabla g(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \end{array} \right\} k = 1, \ldots K$$

$$x \in X, \; y \in Y, \; \alpha \in \boldsymbol{R}^1$$

where $\nabla^2 L(x^k, y^k)$ is the Hessian of the Lagrangian of the last NLP subproblem. Note that Z^K does not predict valid lower bounds in this case. As noted by Ding-Mei and Sargent (1992), who developed a master problem similar to M-MIQP, the quadratic approximations can help to reduce the number of major iterations since an improved representation of the continuous space is obtained. Note also that for convex *f(x, y)* and *g(x,y)* using (M-MIQP) leads to rigorous solutions since the outer-approximations remain valid. Also, if the function *f(x,y)* is nonlinear in y, and y is a general integer variable, Fletcher and Leyffer (1994) have shown that the original OA algorithm may require a much larger number of iterations to converge than when the master problem (M-MIQP) is used. This, however, comes at the price of having to solve an MIQP instead of an MILP. Of course, the ideal situation is the case when the original problem (P1) is quadratic in the objective function and linear in the constraints, as then (M-MIQP) is an exact representation of such a mixed-integer quadratic program.

Reducing dimensionality of the master problem in OA. The master problem (RM-OA) can involve a rather large number of constraints, due to the accumulation of linearizations. One option is to keep only the last linearization point, but this can lead to nonconvergence even in convex problems, since then the monotonic increase of the lower bound is not guaranteed. A rigorous way of reducing the number of constraints without greatly sacrificing the strength of the lower bound can be achieved in the case of the "largely" linear MINLP problem:

$$\begin{array}{rl} \min & Z = a^T w + r(v) + c^T y \qquad \text{(PL)} \\ s.t. & Dw + t(v) + Cy \leq 0 \\ & Fw + Gv + Ey \leq b \\ & w \in W, \; v \in V, \; y \in Y \end{array}$$

where *(w, v)* are continuous variables and *r(v)* and *t(v)* are nonlinear convex functions. As shown by Quesada and Grossmann (1992), linear approximations to the nonlinear objective and constraints can be aggregated with the following MILP master problem:

$$\begin{array}{rl} \min & Z_L^K = a^T w + \beta + c^T y \qquad \text{(M-MIPL)} \\ s.t. & \beta \geq r(v^k) + (\lambda^k)^T [Dw\, t(v^k) + Cy] - (\mu^k)^T (G(v - v^k) \quad k = 1, \ldots.K \end{array}$$

$$Fw + Gv + Ey \le b$$

$$w \in W,\ v \in V,\ y \in Y,\ \beta \in \boldsymbol{R}^1$$

Numerical results have shown that the quality of the bounds is not greatly degraded with the above MILP as might happen if GBD is applied to (PL).

Handling of equalities. For the case when linear equalities of the form $h(x, y) = 0$ are added to (P1) there is no major difficulty since these are invariant to the linearization points. If the equations are nonlinear, however, there are two difficulties. First, it is not possible to enforce the linearized equalities at K points. Second, the nonlinear equations may generally introduce nonconvexities, unless they relax as convex inequalities (see Bazaara et al, 1994). Kocis and Grossmann (1987) proposed an equality relaxation strategy in which the nonlinear equalities are replaced by the inequalities,

$$T^k\ \nabla h(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \qquad (1)$$

where $T^k = \{t_{ii}^k\}$, and $t_{ii}^k = sign\ (\lambda_i^k)$ in which λ_i^k is the multiplier associated to the equation $h_i(x, y) = 0$. Note that if these equations relax as the inequalities $h(x, y) \le 0$ for all y, and $h(x, y)$ is convex, this is a rigorous procedure. Otherwise, nonvalid supports may be generated. Also, note that in the master problem of GBD, (RM-GBD), no special provision is required to handle equations since these are simply included in the Lagrangian cuts. However, similar difficulties as in OA arise if the equations do not relax as convex inequalities.

Handling of nonconvexities. When $f(x,y)$ and $g(x,y)$ are nonconvex in (P1), or when nonlinear equalities, $h(x, y) = 0$, are present, two difficulties arise. First, the NLP subproblems (NLP1), (NLP2), (NLPF) may not have a unique local optimum solution. Second, the master problem (M-MIP) and its variants (e.g. M-MIPF, M-GBD, M-MIQP), do not guarantee a valid lower bound Z_L^K or a valid bounding representation with which the global optimum may be cut off. One possible approach to circumvent this problem is reformulation. This, however, is restricted to special cases, most notably in geometric programming constraints (posynomials) in which exponential transformations, $u=exp(x)$, can be applied for convexification.

Rigorous global optimization approaches for addressing nonconvexities in MINLP problems can be developed when special structures are assumed in the continuous terms (e.g. bilinear, linear fractional, concave separable). Specifically, the idea is to use convex envelopes or underestimators to formulate lower-bounding convex MINLP problems. These are then combined with global optimization techniques for continuous variables (Falk and Soland, 1969;

Horst and Tuy, 1996; Ryoo and Sahinidis, 1995; Quesada and Grossmann, 1995; Grossmann, 1996; Zamora and Grossmann, 1999; Floudas, 2000), which usually take the form of spatial branch and bound methods. The lower bounding MINLP problem has the general form,

$$\begin{aligned} &\min Z = \bar{f}(x, y) \\ s.t.\ \ &\bar{g}_j(x, y) \le 0 \ \ j \in J \\ &x \in X, y \in Y \end{aligned} \qquad \text{(LB-P1)}$$

where $\bar{f}, \bar{g},$ are valid convex underestimators such that $\bar{f}(x,y) \le f(x,y),$ and the inequalities $\bar{g}(x,y) \le 0$ are satisfied if $g(x,y) \le 0.$ A typical example of convex underestimators are for instance the convex envelopes by McCormick (1976) for bilinear terms.

Examples of global optimization methods for MINLP problems include the branch and reduce method by Ryoo and Sahinidis (1995) and Tawarmalani and Sahinidis (2000), the α-BB method by Adjiman et al (2000), the reformulation/spatial branch and bound search method by Smith and Pantelides (1999), the branch and cut method by Kesavan and Barton (2000), and the disjunctive branch and bound method by Lee and Grossmann (2001). All these methods rely on a branch and bound procedure. The difference lies on how to perform the branching on the discrete and continuous variables. Some methods perform the spatial tree enumeration on both the discrete and continuous variables on problem (LB-P1). Other methods perform a spatial branch and bound on the continuous variables and solve the corresponding MINLP problem (LB-P1) at each node using any of the methods reviewed earlier in the paper. Finally, other methods, branch on the discrete variables of problem (LB-P1), and switch to a spatial branch and bound on nodes where a feasible value for the discrete variables is found. The methods also rely in on procedures for tightening the lower and upper bounds of the variables, since these have a great effect on the quality of the underestimators. Since the tree searches are not finite (except for ε-convergence), these methods can be computationally expensive. However, their major advantage is that they can rigorously find the global optimum. It should also be noted that specific cases of nonconvex MINLP problems have been handled. An example is the work of Pörn and Westerlund (2000), who have addressed the solution of MINLP problems with pseudo-convex objective function and convex inequalities through an extension of the ECP method.

The other option for handling nonconvexities is to apply a heuristic strategy to try to reduce as much as possible the effect of nonconvexities. While not being rigorous, this requires much less computational effort. We will describe here an approach for reducing the effect of nonconvexities at the level of the MILP master problem.

Viswanathan and Grossmann (1990) proposed to introduce slacks in the MILP master problem to reduce the likelihood of cutting-off feasible solutions. This master problem (Augmented Penalty/Equality Relaxation) (APER) has the form:

$$\min \quad Z^K = \alpha + \sum_{k=1}^{K} \left[w_p^k p^k + w_q^k q^k \right] \qquad \text{(M-APER)}$$

$$\begin{array}{rl} s.t. & \left.\begin{array}{r} \alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ T^k \, \nabla h(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq p^k \\ g(x^k, y^k) + \nabla g(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq q^k \end{array}\right\} \; k = 1, \ldots K \\ & \sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \leq \left| B^k \right| - 1 \quad k = 1, \ldots K \\ & x \in X, \;\; y \in Y, \;\; \alpha \in \boldsymbol{R}^1, \;\; p^k, q^k \geq 0 \end{array}$$

where w_p^k, w_q^k are weights that are chosen sufficiently large (e.g. 1000 times magnitude of Lagrange multiplier). Note that if the functions are convex then the MILP master problem (M-APER) predicts rigorous lower bounds to (P1) since all the slacks are set to zero.

It should also be noted that another modification to reduce the undesirable effects of nonconvexities in the master problem is to apply global convexity tests followed by a suitable validation of linearizations. One possibility is to apply the tests to all linearizations with respect to the current solution vector (y^K, x^K) (Kravanja and Grossmann, 1994). The convexity conditions that have to be verified for the linearizations are as follows:

$$\left.\begin{array}{r} f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} - \alpha \leq \varepsilon \\ T^k \, \nabla h(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq \varepsilon \\ g(x^k, y^k) + \nabla g(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq \varepsilon \end{array}\right\} \; k = 1, \ldots K - 1 \qquad \text{(GCT)}$$

where ε is a vector of small tolerances (e.g. 10^{-10}). Note that the test is omitted for the current linearizations K since these are always valid for the solution point (y^K, x^K) . Based on this test, a validation of the linearizations is performed so that the linearizations for which the above verification is not satisfied are simply dropped from the master problem. This test relies on the assumption that the solutions of the NLP subproblems are approaching the global optimum, and that the successive validations are progressively defining valid feasibility constraints around the global optimum. Also note that if the right hand side coefficients of linearizations are modified to validate the linearization, the test corresponds to the one in the two-phase strategy by Kocis and Grossmann (1988).

COMPUTER CODES FOR MINLP

The number of computer codes for solving MINLP problems is still rather small. The program DICOPT (Viswanathan and Grossmann, 1990) is an MINLP solver that is available in the modeling system GAMS (Brooke et al., 1998). The code is based on the master problem (M-APER) and the NLP subproblems (NLP2). This code also uses the relaxed (NLP1) to generate the first linearization for the above master problem, with which the user need not specify an initial integer value. Also, since bounding properties of (M-APER) cannot be guaranteed, the search for nonconvex problems is terminated when there is no further improvement in the feasible NLP subproblems. This is a heuristic that works reasonably well in many problems. Codes that implement the branch-and-bound method using subproblems (NLP1) include the code MINLP_BB that is based on an SQP algorithm (Leyffer, 2001) and is available in AMPL, the code BARON (Sahinidis, 1996) that also implements global optimization capabilities, and the code SBB which is available in GAMS (Brooke et al, 1998). The code α–ECP implements the extended cutting plane method by Westerlund and Pettersson (1995), including the extension by Pörn and Westerlund (2000). Finally, the code MINOPT (Schweiger and Floudas, 1998) also implements the OA and GBD methods, and applies them to mixed-integer dynamic optimization problems. It is difficult to make general remarks on the efficiency and reliability of all these codes and their corresponding methods since no systematic comparison has been made. However, one might anticipate that branch and bound codes are likely to perform better if the relaxation of the MINLP is tight. Decomposition methods based on OA are likely to perform better if the NLP subproblems are relatively expensive to solve, while GBD can perform with some efficiency if the MINLP is tight, and there are many discrete variables. ECP methods tend to perform well on mostly linear problems.

LOGIC BASED METHODS

Recently there has been a new trend of representing discrete/continuos optimization problems by models consisting of algebraic constraints, logic disjunctions and logic relations (Beaumont, 1991; Raman and Grossmann, 1993, 1994; Türkay and Grossmann, 1996; Hooker and Osorio, 1999; Hooker, 2000; Lee and Grossmann, 2000). In particular, the mixed-integer program (P1) can also be formulated as a generalized disjunctive program (Raman and Grossmann,1994), which can be regarded as a generalization of disjunctive programming (Balas, 1985):

$$
\begin{aligned}
&Min \quad Z = \sum_k c_k + f(x) \\
&st \quad g(x) \leq 0 \qquad\qquad\qquad\qquad (GDP) \\
&\bigvee_{i \in D_k} \begin{bmatrix} Y_{ik} \\ h_{ik}(x) \leq 0 \\ c_k = \gamma_{ik} \end{bmatrix} \quad k \in SD \\
&\Omega(Y) = True \\
&x \in R^n, c \in R^m, Y \in \{true, false\}^m
\end{aligned}
$$

In problem (GDP) Y_{ik} are the Boolean variables that establish whether a given term in a disjunction is true [$h_{ik}(x) \leq 0$], while $\Omega(Y)$ are logical relations assumed to be in the form of propositional logic involving only the Boolean variables. Y_{ik} are auxiliary variables that control the part of the feasible space in which the continuous variables, x, lie, and the variables c_k represent fixed charges which are activated to a value γ_{ik} if the corresponding term of the disjunction is true. Finally, the logical conditions, $\Omega(Y)$, express relationships between the disjunctive sets. In the context of synthesis problems the disjunctions in (GDP) typically arise for each unit i in the following form:

$$
\begin{bmatrix} Y_i \\ h_i(x) \leq 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in I \qquad (2)
$$

in which the inequalities h_i apply and a fixed cost γ_i is incurred if the unit is selected (Y_i); otherwise ($\neg Y_i$) there is no fixed cost and a subset of the x variables is set to zero with the matrix B^i, which has all zero elements, except for b_{jj}=1 if variable x_j must be set to zero.

It is important to note that any problem posed as (GDP) can always be reformulated as an MINLP of the form of problem (P1), and any problem in the form of (P1) can be posed in the form of (GDP). For modeling purposes, however, it is advantageous to start with model (GDP) as it captures more directly both the qualitative (logical) and quantitative (equations) part of a problem (Vecchietti and Grossmann, 1999, 2000). As for the transformation from (GDP) to

(P1), the most straightforward way is to replace the Boolean variables Y_{ik} by binary variables y_{ik}, and the disjunctions by "big-M" constraints of the form,

$$\begin{aligned} & h_{ik}(x) \le M(1-y_{ik}),\ i \in D_k,\ k \in SD \\ & \sum_{i \in D_k} y_{ik} = 1,\ \ k \in SD \end{aligned} \tag{3}$$

where M is a large valid upper bound. Finally, the logic propositions $\Omega(y)=True$, are converted into linear inequalities as described in Williams (1985) (see also Raman and Grossmann, 1991). The drawback with the "big-M" constraints in (3) is that their relaxation is often weak.

For the solution of problem (GDP), Grossmann and Lee (2001) have shown, based on the work by Stubbs and Mehrotra (1999), that the convex hull of the disjunction in the Generalized Disjunctive Program (GDP), is given by the following theorem:

Theorem 2. *The convex hull of each disjunction $k \in SD$ in problem (GDP),*

$$\bigvee_{i \in D_k} \begin{bmatrix} Y_{ik} \\ h_{ik}(x) \le 0 \\ c_k = \gamma_{ik} \end{bmatrix} \tag{4}$$

where $h_{ik}(x) \le 0$ are convex inequalities, is a convex set and is given by,

$$\begin{aligned} x = \sum_{i \in D_k} \nu_{ik},\ \sum_{i \in D_k} \lambda_{ik} = 1 \\ c_k = \sum_{i \in D_k} \lambda_{ik} \\ \lambda_{ik} h_{ik}(\nu_{ik}/\lambda_{ik}) \le 0 \quad i \in D_k \end{aligned} \tag{CH$_k$}$$

This proof follows from performing an exact linearization of (4) with the non-negative variables λ_{ik}, and by relying on the proof by Stubbs and Mehrotra(1999) that $\lambda h(\nu/\lambda)$ is a convex function if $h(x)$ is a convex function. In (CH$_k$) the variables ν_{ik} can be interpreted as disaggregated variables that are assigned to each disjunctive term, while λ_{ik}, can be interpreted as weight factors that determine the validity of the inequalities in the corresponding disjunctive term. Note also that (CH$_k$) reduces to the result by Balas (1985) for the case of linear constraints. The following corollary follows (Grossmann and Lee, 2001) from Theorem 2:

Corollary. *The nonlinear programming relaxation of (GDP) is given by,*

$$
\begin{aligned}
&\min Z^L = \sum\sum \gamma_{ik}\lambda_{ik} + f(x) \\
&st \quad g(x) \le 0 \\
&\quad x = \sum_{i\in D_k} \nu_{ik}\,,\ \sum_{i\in D_k} \lambda_{ik} = 1 \qquad k \in SD \\
&\quad \lambda_{ik} h_{ik}(\nu_{ik} / \lambda_{ik}) \le 0 \qquad i \in D_k\,, k \in SD \\
&\qquad A\lambda \le a
\end{aligned}
\tag{RDP}
$$

$$x \in R^n,\ \nu_{ik} \ge 0,\quad 0 < \lambda_{ik} \le 1,\ \ i\in D_k,\ k\in SD$$

and yields a valid lower bound to the solution of problem (GDP).

The relaxation problem (RDP), which is related to the work by Ceria and Soares (1999), can be used as a basis to construct a special purpose branch and bound method as has been proposed by Lee and Grossmann (2000). The basic idea in this method is to directly branch on the constraints corresponding to particular terms of each disjunction, while considering the convex hull of the remaining disjunctions or disjunctive terms. Compared to the conventional branch and bound method applied to the equivalent MINLP problem, the disjunctive branch and bound often yields tighter lower bounds. Alternatively, problem (RDP) can also be used to reformulate problem (GDP) as a tight MINLP problem of the form,

$$
\begin{aligned}
&\min Z^L = \sum\sum \gamma_{ik}\lambda_{ik} + f(x) \\
&st \quad g(x) \le 0 \\
&\quad x = \sum_{i\in D_k} \nu_{ik}\,,\ \sum_{i\in D_k} \lambda_{ik} = 1 \qquad k \in SD \\
&\quad (\lambda_{ik} + \varepsilon) h_{ik}(\nu_{ik} / (\lambda_{ik} + \varepsilon)) \le 0 \qquad i \in D_k\,,\ k \in SD \\
&\quad -U_{ik}\lambda_{ik} \le \nu_{ik} \le U_{ik}\lambda_{ik} \\
&\qquad\qquad A\lambda \le a \\
&x \in R^n,\ \nu_{ik} \ge 0,\quad \lambda_{ik} = \{0,1\}, \qquad i\in D_k,\ k\in SD
\end{aligned}
\tag{MIP-DP}
$$

in which ε is a small tolerance to avoid numerical difficulties, and λ_{ik} are binary variables that represent the Boolean variables Y_{ik}. All the algorithms that were discussed in the section on MINLP methods can be applied to solve this problem.

We consider next OA and GBD algorithms for solving problem (GDP). As described in Türkay and Grossmann (1996), for fixed values of the Boolean variables, $Y_{\hat{i}k}$ = *true* and Y_{ik} = false for $\hat{i} \neq i$, the corresponding NLP subproblem is as follows:

$$\begin{aligned}
&\min Z = \sum_{k \in SD} c_k + f(x) \\
&s.t. \quad g(x) \le 0 \\
&\left.\begin{aligned} h_{ik}(x) &\le 0 \\ c_k &= \gamma_{ik} \end{aligned}\right\} \; for\, Y_{\hat{i}k} = true \;\; \hat{i} \in D_k, k \in SD \\
&\left.\begin{aligned} B^i x &= 0 \\ c_k &= 0 \end{aligned}\right\} \; for\, Y_{ik} = false \;\; i \in D_k, i \ne \hat{i}, k \in SD \\
&x \in R^{n,} \;\; c_i \in R^{m,}
\end{aligned} \qquad \text{(NLPD)}$$

Note that for every disjunction $k \in SD$ only constraints corresponding to the Boolean variable $Y_{\hat{i}k}$ that is true are imposed, thus leading to a reduction in the size of the problem. Also, fixed charges γ_{ik} are only applied to these terms. Assuming that K subproblems (NLPD) are solved in which sets of linearizations $l = 1,...K$ are generated for subsets of disjunction terms $L_{ik} = \{ l \mid Y^l_{ik} = true \}$, one can define the following disjunctive OA master problem:

$$\begin{aligned}
&Min \quad Z = \sum_k c_k + \alpha \\
&s.t. \quad \left.\begin{aligned} \alpha &\ge f(x^l) + \nabla f(x^l)^T (x - x^l) \\ g(x^l) &+ \nabla g(x^l)^T (x - x^l) \le 0 \end{aligned}\right\} \; l = 1,..., L \qquad \text{(MGDP)} \\
&\bigvee_{i \in D_k} \begin{bmatrix} Y_{ik} \\ h_{ik}(x^l) + \nabla h_{ik}(x^l)^T (x - x^l) \le 0 \\ l \in L_{ik} \\ c_k = \gamma_{ik} \end{bmatrix} \quad k \in SD \\
&\Omega(Y) = True \\
&\alpha \in R, \;\; x \in R^n, c \in R^m, Y \in \{true, false\}^m
\end{aligned}$$

It should be noted that before applying the above master problem it is necessary to solve various subproblems (NLPD) so as to produce at least one linear approximation of each of the terms in the disjunctions. As shown by Türkay and Grossmann (1996) selecting the smallest number of subproblems amounts to solving a set covering problem, which is of small size and easy to solve. In the context of a process flowsheet synthesis problems, another way of generating the linearizations in (MGDP) is by starting with an initial flowsheet and suboptimizing the remaining subsystems as in the modelling/decomposition strategy (Kocis and Grossmann, 1989; Kravanja and Grossmann, 1990).

The above problem (MGDP) can be solved by the methods described by Beaumont (1991), Raman and Grossmann (1994), and Hooker and Osorio (1996). It is also interesting to note that for the case of process networks, Türkay and Grossmann (1996) have shown that if the convex hull representation of the disjunctions in (2) is used in (MGDP), then converting the logic relations *Ω(Y)* into the inequalities $Ay \le a$, leads to the following MILP problem,

$$
\begin{array}{ll}
 & Min \quad Z = \sum_k c_k + \alpha \\
s.t. & \left.\begin{array}{l} \alpha \ge f(x^l) + \nabla f(x^l)^T (x - x^l) \\ g(x^l) + \nabla g(x^l)^T (x - x^l) \le 0 \end{array}\right\} \; l = 1,\dots,L \qquad \text{(MIPDF)} \\
\multicolumn{2}{l}{\nabla_{x_{Z_i}} h_i(x^l)^T x_{Z_i} + \nabla_{x_{N_i}} h_i(x^l)^T x_{N_i}^1 \le \left[-h_i(x^l) + \nabla_x h_i(x^l)^T x^l\right] y_i \quad l \in K_L^i, i \in I} \\
 & x_{N_i} = x_{N_i}^1 + x_{N_i}^2 \\
 & 0 \le x_{N_i}^1 \le x_{N_i}^U y_i \\
 & 0 \le x_{N_i}^2 \le x_{N_i}^U (1 - y_i) \\
 & \qquad Ay \le a \\
\multicolumn{2}{l}{x \in R^n, \; x_{N_i}^1 \ge 0, x_{N_i}^2 \ge 0, \; y \in \{0,1\}^m}
\end{array}
$$

where the vector *x* is partitioned into the variables (x_{Z_i}, x_{N_i}) for each disjunction i according to the definition of the matrix B^i (i.e. x_z referes to non-zero rows of this matrix). The linearization set is given by $K_L^i = \{l \mid Y_i^l = \text{True}, \; l = 1,...,L\}$ that denotes the fact that only a subset of inequalities were enforced for a given subproblem l. It is interesting to note that the logic-based Outer-Approximation algorithm represents a generalization of the modeling/decomposition strategy Kocis and Grossmann (1989) for the synthesis of process flowsheets.

Türkay and Grossmann (1996) have also shown that while a logic-based Generalized Benders method (Geoffrion, 1972) cannot be derived as in the case of the OA algorithm, one can exploit the property for MINLP problems that performing one Benders iteration (Türkay and Grossmann, 1996) on the MILP master problem of the OA algorithm, is equivalent to generating a Generalized Benders cut. Therefore, a logic-based version of the Generalized Benders method consists of performing one Benders iteration on the MILP master problem (MIPDF) (see property 5). It should also be noted that slacks can be introduced to (MGDP) and to (MIPDF) to reduce the effect of nonconvexities as in the augmented-penalty MILP master problem (Viswanathan and Grossmann, 1990). Finally, Lee and Grossmann (2000) noted that for the case when the disjunctions have the form of (2), there is the following realtionship of problem (MIP-DP) with the logic-based outer-approximation by Türkay and Grossmann (1996). If one considers fixed values of λ_{ik} this leads to an NLP subproblem of the form (NLPD). If one then performs a linearization on problem (MIP-DP), this leads to the MILP problem (MIPDF).

Example

We present here numerical results on an example problem dealing with the synthesis of a process network that was originally formulated by Duran and Grossmann (1986) as an MINLP problem, and later by Türkay and Grossmann (1986) as a GDP problem. Fig. 3 shows the superstructure which involves the possible selection of 8 processes. The Boolean variables Y_j denote the existence or non-existence of processes 1-8. The global optimal solution is $Z^* = 68.01$, consists of the selection of processes 2,4,6, and 8.

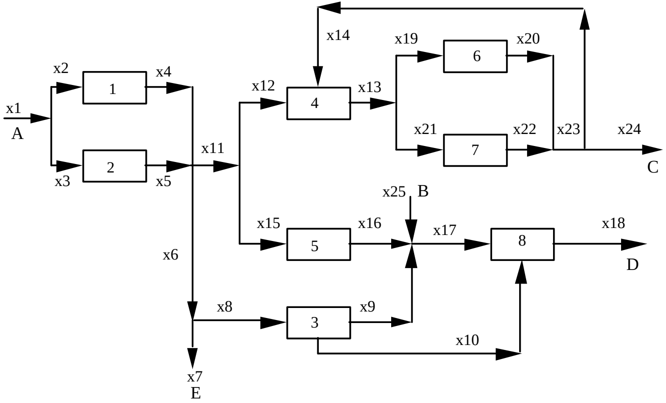


Fig. 3. Superstructure for process network example

The model in the form of the GDP problem involves disjunctions for the selection of units, and propositional logic for the relationship of these units. Each disjunction contains the equations for each unit (these relax as convex inequalities). The model is as follows:

a) Objective function:

$$\min \quad Z=c_1+c_2+c_3+c_4+c_5+c_6+c_7+c_8+x_2-10x_3+x_4-15x_5-40x_9+15x_{10}$$
$$+15x_{14}+80x_{17}-65x_{18}+25x_{19}-60x_{20}+35x_{21}-80x_{22}-35x_{25}+122$$

b) Material balances at mixing/splitting points:

$$x_3+x_5-x_6-x_{11} = 0$$
$$x_{13}-x_{19}-x_{21} = 0$$
$$x_{17}-x_9-x_{16}-x_{25} = 0$$
$$x_{11}-x_{12}-x_{15} = 0$$

$$x_6 - x_7 - x_8 = 0$$

$$x_{23} - x_{20} - x_{22} = 0$$

$$x_{23} - x_{14} - x_{24} = 0$$

c) Specifications on the flows:

$$x_{10} - 0.8x_{17} \le 0$$

$$x_{10} - 0.4x_{17} \ge 0$$

$$x_{12} - 5x_{14} \le 0$$

$$x_{12} - 2x_{14} \ge 0$$

d) Disjunctions:

$$Unit\,1: \begin{bmatrix} Y_1 \\ \exp(x_3) - 1 - x_2 \le 0 \\ c_1 = 5 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ x_2 = x_3 = 0 \\ c_1 = 0 \end{bmatrix}$$

$$Unit\,2: \begin{bmatrix} Y_2 \\ \exp(x_5/1.2) - 1 - x_4 \le 0 \\ c_2 = 8 \end{bmatrix} \vee \begin{bmatrix} \neg Y_2 \\ x_4 = x_5 = 0 \\ c_2 = 0 \end{bmatrix}$$

$$Unit\,3: \begin{bmatrix} Y_3 \\ 1.5x_9 - x_8 + x_{10} = 0 \\ c_3 = 6 \end{bmatrix} \vee \begin{bmatrix} \neg Y_3 \\ x_8 = x_9 = x_{10} = 0 \\ c_3 = 0 \end{bmatrix}$$

$$Unit\,4: \begin{bmatrix} Y_4 \\ 1.5(x_{12} + x_{14}) - x_{13} = 0 \\ c_4 = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_4 \\ x_{12} = x_{13} = x_{14} = 0 \\ c_4 = 0 \end{bmatrix}$$

$$Unit\,5: \begin{bmatrix} Y_5 \\ x_{15} - 2x_{16} = 0 \\ c_5 = 6 \end{bmatrix} \vee \begin{bmatrix} \neg Y_5 \\ x_{15} = x_{16} = 0 \\ c_5 = 0 \end{bmatrix}$$

$$Unit\,6: \begin{bmatrix} Y_6 \\ \exp(x_{20}/1.5) - 1 - x_{19} = 0 \\ c_6 = 7 \end{bmatrix} \vee \begin{bmatrix} \neg Y_6 \\ x_{19} = x_{20} = 0 \\ c_6 = 0 \end{bmatrix}$$

$$Unit\,7: \begin{bmatrix} Y_7 \\ \exp(x_{22}) - 1 - x_{21} = 0 \\ c_7 = 4 \end{bmatrix} \vee \begin{bmatrix} \neg Y_7 \\ x_{21} = x_{22} = 0 \\ c_7 = 0 \end{bmatrix}$$

$$Unit\,8: \begin{bmatrix} Y_8 \\ \exp(x_{18}) - 1 - x_{10} - x_{17} = 0 \\ c_8 = 5 \end{bmatrix} \vee \begin{bmatrix} \neg Y_8 \\ x_{10} = x_{17} = x_{18} = 0 \\ c_8 = 0 \end{bmatrix}$$

e) Propositional Logic [Ω=(Y_i)]:

$Y_1 \Rightarrow Y_3 \vee Y_4 \vee Y_5$
$Y_2 \Rightarrow Y_3 \vee Y_4 \vee Y_5$
$Y_3 \Rightarrow Y_1 \vee Y_2$
$Y_3 \Rightarrow Y_8$
$Y_4 \Rightarrow Y_1 \vee Y_2$
$Y_4 \Rightarrow Y_6 \vee Y_7$
$Y_5 \Rightarrow Y_1 \vee Y_2$
$Y_5 \Rightarrow Y_8$
$Y_6 \Rightarrow Y_4$
$Y_7 \Rightarrow Y_4$
$Y_8 \Rightarrow Y_3 \vee Y_5 \vee (\neg Y_3 \wedge \neg Y_5)$

f) Specifications:

$Y_1 \veebar Y_2$
$Y_4 \veebar Y_5$
$Y_6 \veebar Y_7$

g) Variables:

$x_j,\ c_i \geq 0,\ Y_i = \{True, False\} \quad i=1,2,...,8,\ \ j=1,2,...,25$

As seen in Table 1, the branch and bound (BB) algorithm by Lee and Grossmann (2000) finds the optimal solution in only 5 nodes compared with 17 nodes of standard branch and bound method when applied to the MINLP formulation with big-M constraints. A major difference in these two methods is the lower bound predicted by the relaxed NLP. Clearly the bound at the root node in the proposed BB method, which is given by problem (RDP), is much stronger (62.48 vs. 15.08). Table 2 shows the comparison with other algorithms when the problem is reformulated as the tight MINLP problem (MIP-DP). Note that the proposed BB algorithm and the standard BB yield the same lower bound (62.48) since they start by solving the same relaxation problem. The difference in the number of nodes, 5 vs. 11, lies in the branching rules, which are better exploited in the special branch and bound method by Lee and Grossmann (2000). The OA, GBD and ECP methods start with initial guess Y^0 = [1,0,1,1,0,0,1,1]. Note that in GBD and OA methods, one major iteration consists of one NLP subproblem and one MILP master problem. As predicted by the theory, the logic-based OA method yields the lower bound 8.541, which is stronger than the one of the GBD method. Therefore, OA requires 3 major iterations versus 8 from GBD. The ECP method requires 7 iterations, each involving the solution of an MILP. Thus, these results show the improvements that can be obtained through the logic based formulation, such as with the generalized disjunctive program (GDP). It also shows that the OA algorithm requires fewer major iterations than the GBD and ECP methods.

Table 1. Comparison of Branch and Bound methods

Model	**Big-M (BM)**	**MIP-DP**
Method	**Standard BB**	**Proposed BB Algorithm**
No. of nodes	17	5
Relaxed NLP	15.08	62.48

Table 2. Comparison of several algorithms on reformulation MIP-DP.

Method*	**Standard BB**	**Proposed BB**	**OA**	**GBD**	**ECP**
No. of nodes or Iteration	11 (Nodes)	5 (Nodes)	3 (Iter.)	8 (Iter.)	7 (Iter.)
Lower Bound	62.48	62.48	8.541	-551.4	-5.077

**All methods solve the reformulated MINLP problem (MIP-DP).*

CONCLUDING REMARKS

This paper has presented a unified treatment and derivation of the different MINLP algorithms that have been reported in the literature. As has been shown for the case where the problem is expressed in algebraic form, Branch and Bound, Generalized Benders, Outer-Approximation, Extended Cutting Plane and LP/NLP based Branch and Bound can easily be derived from three basic NLP subproblems and one master MILP problem. Similar derivations can be obtained for the case when the problem is expressed as a generalized disjunctive optimization problem. Major theoretical properties of these methods have been presented, as well as extensions for nonconvex problems. The numerical results of the small example have confirmed the theoretical properties that were discussed in the paper.

ACKNOWLEDGMENT

The author would like to gratefully acknowledge financial support from the National Science Foundation under grant ACI-0121497. He also would like to acknowledge the work of many of his previous students who have made this paper possible.

References

Adjiman C.S., I.P. Androulakis and C.A. Floudas, “Global Optimization of Mixed-Integer Nonlinear Problems. *AIChE Journal*, **46**(9), 1769-1797, 2000.

Balas, E. ,"Disjunctive Programming and a hierarchy of relaxations for discrete optimization problems", *SIAM J. Alg. Disc. Meth*., **6**, 466-486 (1985).

Balas, E., Ceria, S. and Cornuejols, G. A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs, *Mathematical Programming*, **58**, 295-324 (1993).

Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh and P.H. Vance, "Branch-and-price: Column generation for solving huge integer programs." *Operations Research*, **46**, 316-329 (1998).

Bazaraa, M.S., H.D. Sherali and C.M. Shetty, "Nonlinear Programming," John Wiley (1994).

Beaumont, N. "An Algorithm for Disjunctive Programs," *European Journal of Operations Research*, **48**, 362-371 (1991).

Benders J.F., Partitioning procedures for solving mixed-variables programming problems. *Numeri. Math.*, **4**, 238-252 (1962).

Biegler, L.T., I.E. Grossmann and A.W. Westerberg, "Systematic Methods for Chemical Process Design", Prentice-Hall (1997).

Borchers, B. and J.E. Mitchell, "An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programming", *Computers and Operations Research*, **21**, 359-367 (1994).

Brooke, A., Kendrick, D, Meeraus, A., Raman, R. "GAMS - A User's Guide", www.gams.com (1998).

Ceria S. and J. Soares, Convex Programming for Disjunctive Optimization. *Mathematical Programming*, **86**(3), 595-614, 1999.

Dakin, R.J., "A Tree Search Algorithm for Mixed-Integer Programming Problems", *Computer Journal*, **8**, 250-255 (1965).

Ding-Mei and R.W.H. Sargent, "A Combined SQP and Branch and Bound Algorithm for MINLP Optimization", Internal Report, Centre for Process Systems Engineering, London (1992).

Duran, M.A. and I.E. Grossmann, "An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs," *Math Programming* **36**, 307 (1986).

Falk J.E. and R.M. Soland, An Algorithm for separable nonconvex programming problems. Management Science, **15**, 550-569, 1969.

Fletcher, R. and S. Leyffer, "Solving Mixed Integer Nonlinear Programs by Outer Approximation", *Math Programming* **66**, 327 (1994).

Flippo, O.E. and A.H.G. Rinnoy Kan, "Decomposition in General Mathematical Programming", *Mathematical Programming*, **60**, 361-382 (1993).

Floudas C.A., *Deterministic Global Optimization: Theory, Methods and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

Geoffrion, A. M., "Generalized Benders Decomposition," *Journal of Optimization Theory and Applications*, **10**(4), 237-260 (1972).

Grossmann, I.E., "Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis", *Advances in Chemical Engineering*, Vol. 23, Process Synthesis, pp.171-246 (1996a).

Grossmann, I.E. (ed.), "Global Optimization in Engineering Design", Kluwer, Dordrecht (1996b).

Grossmann, I.E., J.A. Caballero and H. Yeomans, "Advances in Mathematical Programming for Automated Design, Integration and Operation of Chemical Processes," *Korean J. Chem. Eng.*, 16, 407-426 (1999).

Grossmann, I.E. and M.M. Daichendt, "New Trends in Optimization-based Approaches for Process Synthesis", *Computers and Chemical Engineering,* **20**, 665-683 (1996).

Grossmann, I.E. and S. Lee, "Generalized Disjunctive Programming: Nonlinear Convex Hull Relaxation," submitted for publication (2001).

Grossmann, I.E. and Z. Kravanja, "Mixed-integer Nonlinear Programming: A Survey of Algorithms and Applications", The IMA Volumes in Mathematics and its Applications, Vol.93, Large-Scale Optimization with Applications. Part II: Optimal Design and Control (eds, Biegler, Coleman, Conn, Santosa) pp.73-100, Springer Verlag (1997).

Grossmann, I.E., J. Quesada, R Raman and V. Voudouris, "Mixed Integer Optimization Techniques for the Design and Scheduling of Batch Processes," Batch Processing Systems Engineering (Eds. G.V. Reklaitis, A.K. Sunol, D.W.T. Rippin, O. Hortacsu), 451-494, Springer-Verlag, Berlin (1996).

Gupta, O. K. and Ravindran, V., "Branch and Bound Experiments in Convex Nonlinear Integer Programming", *Management Science*, **31**(12), 1533-1546 (1985).

Hooker, J.N. and M.A. Osorio, "Mixed logical.linear programming", *Discrete Applied Mathematics*, **96-97**, pp.395-442 (1999).

Hooker, J.N., “Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction,” Wiley (2000).

Horst, R. and P.M. Tuy, “Global Optimization: Deterministic Approaches,” 3rd Ed., Springer-Verlag, Berlin (1996).

Johnson, E.L., G.L. Nemhauser and M.W.P. Savelsbergh, “Progress in Linear Programming Based Branch-and-Bound Algorithms: Exposition,” *INFORMS Journal of Computing,* **12** (2000).

Kallrath, J., “Mixed Integer Optimization in the Chemical Process Industry: Experience, Potential and Future,” *Trans. I .Chem E.*, **78**, Part A, pp.809-822 (2000)

Kelley Jr., J.E., "The Cutting-Plane Method for Solving Convex Programs", *Journal of SIAM* **8**, 703-712 (1960).

Kesavan P. and P.I. Barton, Decomposition algorithms for nonconvex mixed-integer nonlinear programs. *American Institute of Chemical Engineering Symposium Series*, **96**(323), pp.458-461 (1999).

Kesavan P. and P.I. Barton, Generalized branch-and-cut framework for mixed-integer nonlinear optimization problems. *Computers and Chem. Engng.*, **24**, 1361-1366 (2000).

Kocis, G.R. and I.E. Grossmann, "Relaxation Strategy for the Structural Optimization of Process Flowsheets," *Ind. Eng. Chem. Res.* **26**, 1869 (1987)

Lee, S. and I.E. Grossmann, "New Algorithms for Nonlinear Generalized Disjunctive Programming" *Computers and Chemical Engineering* **24**, 2125-2141 (2000).

Leyffer, S., "Deterministic Methods for Mixed-Integer Nonlinear Programming," Ph.D. thesis, Department of Mathematics and Computer Science, University of Dundee, Dundee (1993).

Leyffer, S., "Integrating SQP and Branch and Bound for Mixed Integer Noninear Programming," *Computational Optimization and Applications*, **18**, pp.295-309 (2001).

Magnanti, T. L. and Wong, R. T. (1981). Acclerated Benders Decomposition: Algorithm Enhancement and Model Selection Criteria, *Operations Research*, **29**, 464-484.

McCormick, G.P., "Computability of Global Solutions to Factorable Nonconvex Programs: Part I – ConvexUnderestimating Problems," *Mathematical Programming*, 10, pp.147-175 (1976).

Nabar, S. and Schrage, L., "Modeling and Solving Nonlinear Integer Programming Problems", Presented at Annual AIChE Meeting, Chicago (1991).

Nemhauser, G. L. and Wolsey, L. A.,"Integer and Combinatorial Optimization," Wiley-Interscience, New York (1988).

Pinto, J. and I.E. Grossmann, "Assignment and Sequencing Models for the Scheduling of Chemical Processes", *Annals of Operations Research* **81** 433-466.(1998).

Pörn, R. and T. Westerlund, "A Cutting Plane Method for Minimizing Pseudo-convex Functions in the Mixed-integer Case," *Computers and Chemical Engineering*, **24**, pp.2655-2665 (2000).

Quesada, I. and I.E. Grossmann, "An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems," *Computers and Chemical Engineering* **16**, 937-947 (1992).

Quesada, I.E. and I.E. Grossmann, "A Global Optimization Algorithm for Linear Fractional and Bilinear Programs," *Journal of Global Optimization*, **6**, 39-76 (1995).

Raman, R. and I.E. Grossmann, "Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis," *Computers and Chemical Engineering* **15**, 73 (1991).

Raman, R. and I.E. Grossmann, "Symbolic Integration of Logic in Mixed Integer Linear Programming Techniques for Process Synthesis," *Computers and Chemical Engineering*, **17**, 909 (1993).

Raman, R. and I.E. Grossmann, "Modelling and Computational Techniques for Logic Based Integer Programming," *Computers and Chemical Engineering*, **18**, 563 (1994).

Ryoo H.S. and N.V. Sahinidis, Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design. *Computers and Chem. Engng*., **19**(5), 551-566 (1995).

Sahinidis, N. V., BARON: A General Purpose Global Optimization Software Package, *Journal of Global Optimization*, 8(2), 201-205, 1996.

Sahinidis, N.V. and I.E. Grossmann, "Convergence Properties of Generalized Benders Decomposition," *Computers and Chemical Engineering*, **15**, 481 (1991).

Schweiger C.A. and C.A. Floudas. "Process Synthesis, Design and Control: A Mixed Integer Optimal Control Framework", Proceedings of DYCOPS-5 on Dynamics and Control of Process Systems, pp. 189-194 (1998).

Shah, N., "Single and Multisite Planning and Scheduling: Current Status and Future Challenges," *AIChE Symp. Ser*. **94** (320), 75 (1998).

Smith, E.M.B.and C.C.Pantelides, "A Symbolic Reformulation/Spatial Branch and Bound Algorithm for the Global Optimization of Nonconvex MINLPs," *Computers and Chemical Engineering*, **23**, pp.457-478 (1999).

Stubbs R. and S. Mehrotra, A Branch-and-Cut Method for 0-1 Mixed Convex Programming. *Mathematical Programming*, **86**(3), 515-532 (1999).

Tawarmalani M. and N.V. Sahinidis, Global Optimization of Mixed Integer Nonlinear Programs: A Theoretical and Computational Study. Submitted to *Mathematical Programming*, (2000).

Türkay, M. and I.E. Grossmann, "A Logic Based Outer-Approximation Algorithm for MINLP Optimization of Process Flowsheets", *Computers and Chemical Enginering*, **20**, 959-978 (1996).

Westerlund, T. and F. Pettersson, "A Cutting Plane Method for Solving Convex MINLP Problems", *Computers and Chemical Engineering*, **19**, S131-S136 (1995).

Williams, H.P., "Mathematical Building in Mathematical Programming", John Wiley, Chichester (1985).

Vecchietti, A. and I.E. Grossmann, "LOGMIP: A Discrete Continuous Nonlinear Optimizer", *Computers and Chemical Engineering* **23**, 555-565 (1999).

Vecchietti, A. and I.E. Grossmann, "Modeling Issues and Implementation of Language for Disjunctive Programming", *Computers and Chemical Engineering* **24**, 2143-2155 (2000).

Viswanathan, J. and I.E. Grossmann, "A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization," *Comput. chem. Engng.* **14**, 769 (1990).

Yuan, X., S. Zhang, L. Piboleau and S. Domenech, "Une Methode d'optimisation Nonlineare en Variables Mixtes pour la Conception de Procedes", *RAIRO*, **22**, 331 (1988).

Zamora, J.M. and I.E. Grossmann, "A Branch and Contract Algorithm for Problems with Concave Univariate, Bilinear and Linear Fractional Terms,"*Journal of Gobal Optimization* **14**, 217-249 (1999).