New Algorithms for Nonlinear Generalized Disjunctive Programming

Sangbum Lee and Ignacio E. Grossmann*

Department of Chemical Engineering, Carnegie Mellon University

Pittsburgh, PA 15213

U.S.A.

October 1999 / March 2000

* To whom all correspondence should be addressed: Phone: 412-268-2230, Fax: 412-268-7139, e-mail: grossmann @cmu.edu

ABSTRACT

Generalized Disjunctive Programming (GDP) has been introduced recently as an alternative model to MINLP for representing discrete/continuous optimization problems. The basic idea of GDP consists of representing discrete decisions in the continuous space with disjunctions, and constraints in the discrete space with logic propositions. In this paper, we describe a new convex nonlinear relaxation of the nonlinear GDP problem that relies on the use of the convex hull of each of the disjunctions involving nonlinear inequalities. The proposed nonlinear relaxation is used to reformulate the GDP problem as a tight MINLP problem, and for deriving a branch and bound method. Properties of these methods are given, and the relation of this method with the Logic Based Outer-Approximation method is established. Numerical results are presented for problems in jobshop scheduling, synthesis of process networks, optimal positioning of new products and batch process design.

Keywords: Generalized disjunctive programming, branch and bound, mixed-integer nonlinear programming, nonlinear convex hull.

INTRODUCTION

Mixed Integer Non-Linear Programming (MINLP) models are widely used in discrete/continuous optimization (Grossmann and Kravanja, 1997). MINLP problems arise, for instance, in process synthesis (heat exchanger networks and reactor networks), in process design (optimal positioning of product and feed location in distillation column), in the synthesis of process networks, and in the design and scheduling of batch and continuous multiproduct plants. Algorithms for solving MINLP problems include Branch and Bound (BB) (Gupta and Ravindran, 1985; Borchers and Mitchell, 1994; Stubbs and Mehrotra, 1999; Leyffer, 1999), Outer-Approximation (OA) (Duran and Grossmann, 1986; Yuan et al., 1988; Fletcher and Leyffer, 1994), Generalized Benders Decomposition (GBD) (Geoffrion, 1972), Extended Cutting Plane (ECP) (Westerlund and Pettersson, 1995), LP/NLP based branch and bound (Quesada and Grossmann, 1992), and branch-and-cut (Stubbs and Mehrotra, 1999). For a detailed review, see Grossmann and Kravanja (1997).

Generalized Disjunctive Programming (GDP), which can be regarded as a generalization of disjunctive programming (Balas, 1985), has been introduced as an alternative model to the MINLP problem that uses disjunctions and logic propositions (Raman and Grossmann, 1994).

While the MINLP model is based entirely on algebraic equations and inequalities for discrete/continuous optimization problem, the GDP model allows a combination of algebraic and logical equations, which facilitates the representation of discrete decisions. Türkay and Grossmann (1996) have proposed a logic-based Outer-Approximation algorithm for solving nonlinear GDP problems for process networks involving two terms in each disjunction. This algorithm is based on the idea of extending the Outer-Approximation algorithm by solving NLP subproblems in reduced space, and MILP master problems corresponding to the convex hull of the linearization of the nonlinear inequalities. In addition, several NLP subproblems must be solved to initialize the master problem in order to cover all the terms in the disjunctions. This algorithm has been implemented in LOGMIP, a computer code developed by Vecchietti and Grossmann (1999).

In this paper, we address the solution of GDP problems that involve disjunctions with multiple terms. We first describe the convex hull of a disjunction involving convex nonlinear inequalities, which provide the tightest relaxation of the disjunction. The equations describing the convex hull are used as a basis to develop a convex nonlinear relaxation of the GDP problem. This NLP relaxation can be used for reformulating it as an MINLP problem, or for developing a special purpose branch and bound method which will be described in detail. We examine the relation of the proposed method with the one Türkay and Grossmann (1996) which can handle only disjunctions with two terms and is restricted to process networks. We describe in this paper the basic ideas of the proposed method, and emphasize its geometrical interpretation. Detailed proofs can be found in Lee and Grossmann (1999). The proposed methods are applied to small analytical examples, and to problems dealing with jobshop scheduling, process networks, optimal positioning of new products, and design of a batch process.

GENERALIZED DISJUNCTIVE PROGRAMMING

Consider the Generalized Disjunctive Programming problem (Raman and Grossmann, 1994), which is an extension of the work of Balas (1985). In general, the GDP model includes Boolean variables, disjunctions and logic propositions as shown in problem (P),

$$
\begin{aligned}
&\min Z = \sum_{k\in K} c_k + f(x) \\
&s.t. \qquad r(x) \le 0 \\
&\bigvee_{j\in J_k} \begin{bmatrix} Y_{jk} \\ g_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{bmatrix}, \ k \in K \\
&\Omega(Y) = True \\
&x \ge 0, c_k \ge 0, Y_{jk} \in \{true, false\}
\end{aligned} \tag{P}
$$

Here $x \in R^n$ is the vector of continuous variables and Y_{jk} are Boolean variables. $c_k \in R^1$ are continuous variables and γ_{jk} are fixed charges; f: $R^n \to R^1$ is the term for continuous variables x in the objective function and r: $R^n \to R^q$ are common constraint sets that hold regardless of the discrete decisions. $f(x)$ and $r(x)$ are convex functions. A disjunction is composed of an OR operator ($\vee$) and a number of terms. In each term, there are the Boolean variables Y_{jk}, a set of convex nonlinear inequalities $g_{jk}(x)$, g^i_{jk}: $R^n \to R^1$, $i\in I_{jk}$, $k \in K$, where I_{jk} is an index set of inequalities, and a cost variable c_k. If Y_{jk} is true, then $g_{jk}(x) \le 0$ and $c_k = \gamma_{jk}$ are enforced. Otherwise, the corresponding constraints are ignored. We assume here that each term in the disjunctions gives rise to a non-empty feasible region. In process synthesis problems, $g_{jk}(x)$ are heat or mass balance equations, or specifications of the process, and γ_{jk} are fixed charges for each process. J_k is an index set of the terms for each disjunction k, $J_k = \{ j \mid j = 1,2,...,m_k \}$, $k \in K$. Finally, $\Omega(Y) = True$ correspond to logic propositions in terms of the Boolean variables. The logic propositions $\Omega(Y)$ are expressed in Conjunctive Normal Form (CNF):

$$
\Omega(Y) = \bigwedge_{s=1,2,...,S} \left[\bigvee_{(j,k)\in P_s} (Y_{jk}) \bigvee_{(j,k)\in Q_s} (\neg Y_{jk}) \right] \tag{1}
$$

where P_s is the set of Boolean variables Y_{jk} which are true, and Q_s is the set of Boolean variables Y_{jk} which are false in clause s, $s = 1,2,\ldots,S$. In CNF, every clause that is expressed in terms of the 'OR' operator must be true.

In problem (P), the functions $f(x)$, $r(x)$, and $g_{jk}(x)$ are assumed to be convex and bounded. Also, it is assumed that problem (P) has a non-empty compact feasible region. The GDP problem (P) can be reformulated as the following MINLP problem (BM) by replacing the Boolean variables Y_{jk} by binary variables y_{jk}, and using big-M constraints. The logic constraints $\Omega(Y)$ are converted into linear inequalities (Williams, 1985) which leads to the following big-M MINLP;

$$\min Z = \sum_{k\in K}\sum_{j\in J_k}\gamma_{jk}y_{jk} + f(x)$$
$$s.t. \quad r(x) \le 0$$
$$g_{jk}(x) \le M_{jk}(1-y_{jk}),\ j\in J_k, k\in K \tag{BM}$$
$$\sum_{j\in J_k} y_{jk} = 1,\ k\in K$$
$$Ay \le a$$
$$x \ge 0,\ y_{jk}\in\{0,1\},\ j\in J_k, k\in K$$

In this model, M_{jk} are the "big-M" parameters that render the inequalities $g_{jk}(x)$ redundant when $y_{jk} = 0$. The inequalities $Ay \le a$ can be systematically derived from the CNF form of $\Omega(Y)$ as discussed in Raman and Grossmann (1991). Note also that the relaxation of (BM) is obtained by treating the binary variables as continuous in the range $0 \le Y_{jk} \le 1$.

ILLUSTRATIVE EXAMPLE 1

Consider the following GDP problem with one disjunction,

$$\min Z = (x_1-3)^2 + (x_2-2)^2 + c$$
$$s.t.$$
$$\begin{bmatrix} Y_1 \\ (x_1)^2+(x_2)^2-1\le 0 \\ c=2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1-4)^2+(x_2-1)^2-1\le 0 \\ c=1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ (x_1-2)^2+(x_2-4)^2-1\le 0 \\ c=3 \end{bmatrix} \tag{2}$$
$$0\le x_1, x_2 \le 8, c\ge 0,\ Y_j\in\{true, false\},\ j=1,2,3.$$

There are three terms in the disjunction, and exactly one of them must be true. The feasible region of (2) is given by three disconnected circles as seen in Figure 1. The global optimal solution of (2) is $Z^* = 1.172$, $Y^* = (false, true, false)$ and $x^* = (3.293, 1.707)$.

By using 0-1 variables y_j, (2) can be reformulated as an MINLP problem (BM) with big-M constraints:

$$\min Z = (x_1-3)^2 + (x_2-2)^2 + 2y_1 + y_2 + 3y_3$$
$$s.t. \quad (x_1)^2+(x_2)^2-1 \le M(1-y_1)$$
$$(x_1-4)^2+(x_2-1)^2-1 \le M(1-y_2) \tag{3}$$
$$(x_1-2)^2+(x_2-4)^2-1 \le M(1-y_3)$$
$$y_1+y_2+y_3 = 1$$
$$0\le x_1, x_2\le 8,\ y_1, y_2, y_3\in\{0,1\},\ M=30$$

If y_1 = 1, then the first inequality constraint is enforced and if y_1 = 0, it becomes redundant assuming that M is a sufficiently large number. If the binary variables y_j are treated as continuous variables in the MINLP problem (3), then for M = 30 the relaxed MINLP problem of (3) has the optimal solution Z^* = 1.031 and y^* = (0.029,0.971,0).

GDP PROBLEM WITH ONE DISJUNCTION

For simplicity, we will first assume that in problem (P) we have only one disjunction, i.e., $|K| = 1$. Hence, each term j in the disjunction has only one Boolean variable Y_j, and the index k can be removed from (P) leading to (P1),

$$\begin{aligned} &\min Z = c + f(x) \\ &s.t. \quad r(x) \le 0 \\ &\bigvee_{j \in J} \begin{bmatrix} Y_j \\ g_j(x) \le 0 \\ c = \gamma_j \end{bmatrix} \\ &\Omega(Y) = True \\ &x, c \ge 0, Y_j \in \{true, false\}, j \in J \end{aligned} \tag{P1}$$

Each term in the disjunction defines a feasible region S_j, $j \in J$, where $S_j = \{(x,c) \mid c = \gamma_j, r(x) \le 0, g_j(x) \le 0\}$. Note that problem (2) is a particular instance of problem (P1).

In the following sections, we derive a nonlinear relaxation of problem (P1) which is tighter than the relaxation of the big-M MINLP problem (BM). We use the proposed NLP relaxation as a basis for deriving an MINLP reformulation and propose a special purpose branch and bound method. We then generalize this method to problem (P) which involves multiple disjunctions ($|K| > 1$).

CONVEX HULL OF NONLINEAR DISJUNCTION

Consider the following disjunction that arises in problem (P1):

$$\begin{aligned} &\bigvee_{j \in J} \begin{bmatrix} Y_j \\ g_j(x) \le 0 \\ c = \gamma_j \end{bmatrix} \\ &x, c \ge 0 \end{aligned} \tag{4}$$

where the functions $g_j(x)$ are assumed to be bounded convex functions over x. In addition, x is assumed to be bounded, i.e., $0 \le x \le U$. The disjunction means that exactly one of the Boolean variables Y_j must be true, which in turn means that $g_j(x) \le 0$ and $c = \gamma_j$. These constraints are redundant when Y_j is false.

The convex hull of the disjunction in (4) is given by all points that can be generated from taking the linear combination of points in the feasible regions S_j, $j \in J$. Figure 2 illustrates geometrically the convex hull of the disjunction $(x \in S_1) \vee (x \in S_2) \vee (x \in S_3)$.

As is shown in Appendix A, the convex hull of (4) is given by the following set of equations:

$$
\begin{gathered}
x = \sum_{j \in J} \nu^j, \qquad c = \sum_{j \in J} \lambda_j \gamma_j \\
0 \le \nu^j \le \lambda_j U_j, \; j \in J \\
\sum_{j \in J} \lambda_j = 1, \; 0 \le \lambda_j \le 1, \; j \in J \\
\lambda_j g_j(\nu^j / \lambda_j) \le 0, \; j \in J \\
x, c, \nu^j \ge 0, \; j \in J
\end{gathered}
\tag{5}
$$

The equations in (5) define a convex set in the space (x, c, ν, λ). This property follows from the fact that all equations in (5) are linear, and the last inequality is convex. As proven by Hiriart-Urruty and Lemaréchal (1993), if $g(x)$ is convex and bounded over the feasible region and $\lambda \ge 0$, then the function $h(\nu,\lambda) = \lambda g(\nu/\lambda)$ is a bounded convex function when $h(0,0)$ is defined as its limiting value, 0. Hence the inequalities $\lambda g(\nu/\lambda) \le 0$ are convex (see also Stubbs and Mehrotra, 1999).

The equations in (5) describe the convex relaxation of the disjunction in (4). Note in (5) that x is expressed as the sum of disaggregated variables ν^j, and c is expressed as a convex combination of γ_j with weight factors λ_j. The relaxation in (5) provides the tightest relaxation of the disjunctive feasible region of (4) as it corresponds to its convex hull. Also, if $\lambda_j \rightarrow 1$, then $x \rightarrow \nu^j$ and $c \rightarrow \gamma_j$. $\lambda_j = 1$ implies that Y_j is true and the j-th constraint $\lambda_j g_j(\nu^j/\lambda_j) \le 0$ in (5) is the same as the constraint of the j-th term in (4). Hence, the j-th term in the disjunction of (4) is satisfied when λ_j equals one in (5) (Y_j is *true*). Finally, notice that if $g_j(x)$ is a linear function, (5) reduces to the equations proposed by Balas (1985).

NONLINEAR CONVEX RELAXATION PROBLEM

We define a continuous relaxation of (P1) using as a basis the equations of the convex hull (5). Since this relaxation problem has no Boolean variables, the continuous variables λ_j are used instead, and the logic propositions are represented with the inequalities $A\lambda \leq a$. The Convex Relaxation Programming (CRP) problem for one disjunction (|K| = 1) is then given as follows:

$$\begin{aligned}
\min Z^L &= \sum_{j \in J} \gamma_j \lambda_j + f(x) \\
s.t. \quad & r(x) \leq 0 \\
& x = \sum_{j \in J} \nu^j \\
& 0 \leq \nu^j \leq \lambda_j U_j, \ j \in J \\
& \sum_{j \in J} \lambda_j = 1 \\
& \lambda_j g_j(\nu^j / \lambda_j) \leq 0, \ j \in J \\
& A\lambda \leq a \\
& x, \nu^j \geq 0, 0 \leq \lambda_j \leq 1, \ j \in J
\end{aligned} \qquad \text{(CRP)}$$

where for implementation the inequality $\lambda_j g_j(\nu^j / \lambda_j) \leq 0$ must be reformulated as $(\lambda_j + \varepsilon) g_j(\nu^j / (\lambda_j + \varepsilon)) \leq 0$ where ε is a small tolerance (typical value 0.0001). Note that in (CRP), the number of constraints increases by (n+n×m+1), where n is the dimension of vector x. This is due to the constraints $x = \Sigma\nu^j$, $0 \leq \nu^j \leq \lambda_j U_j$ and $\Sigma\lambda_j = 1$. The number of variables increases by m×(n+1), where m is the number of terms in the disjunction (m = |J|). Problem (CRP), which can be regarded as an extension from the work of Ceria and Soares (1999) for disjunctive programming, corresponds to a convex nonlinear programming problem. This follows from the fact that the logic inequalities are linear and the feasible region of problem (CRP) is convex. Since the objective function contains the linear summation term and $f(x)$ is convex, the objective function is convex. Therefore, problem (CRP) is a convex NLP problem.

It also follows that if the problem (GDP) has a bounded optimal solution, then the optimum of (CRP) is unique and corresponds to its global minimum. Furthermore, the feasible region of (CRP), F_C, is a relaxation of the feasible region of problem (P1), F_P. Therefore, since $F_P \subseteq F_C$, and the objective function of (CRP) is also a relaxation of the objective function of (P1), the solution of (CRP), $(Z^L)^*$, yields a lower bound of the optimal solution of problem (P1), Z^*, i.e.,

$(Z^L)^* \leq Z^*$.

The above properties of (CRP) can be exploited to reformulate problem (P1) as an MINLP. Alternatively we can develop a special branch and bound search procedure as will be shown later in the paper. It should be noted that in problem (CRP), ν^j are the disaggregated variables of the vector of continuous variables x, while λ_j are weights that measure the "closeness" by which each term of the disjunction is satisfied ($x \to \nu^j$ as $\lambda_j \to 1$). Generally, solving an optimization problem with problem (CRP) yields a solution λ_j with fractional values. However, when one of the λ_j becomes 1 and the other weights are zero, problem (CRP) becomes problem (P1) with fixed $Y_j = true$ and all the other $Y_{j', j' \neq j} = false$ in the disjunction.

EXAMPLE 1 CONTINUED

If we apply the (CRP) model to the GDP problem (2), the convex NLP relaxation problem is as follows:

$$\begin{aligned}
\min Z^L &= (x_1-3)^2 + (x_2-2)^2 + 2\lambda_1 + \lambda_2 + 3\lambda_3 \\
s.t. \quad & x_1 = \nu_1^1 + \nu_1^2 + \nu_1^3 \\
& x_2 = \nu_2^1 + \nu_2^2 + \nu_2^3 \\
& 0 \leq \nu_i^j \leq 8\lambda_j, i = 1,2; j = 1,2,3. \\
& \lambda_1 + \lambda_2 + \lambda_3 = 1 \\
& (\lambda_1+\varepsilon)[(\nu_1^1/(\lambda_1+\varepsilon))^2 + (\nu_2^1/(\lambda_1+\varepsilon))^2 - 1] \leq 0 \\
& (\lambda_2+\varepsilon)[(\nu_1^2/(\lambda_2+\varepsilon)-4)^2 + (\nu_2^2/(\lambda_2+\varepsilon)-1)^2 - 1] \leq 0 \\
& (\lambda_3+\varepsilon)[(\nu_1^3/(\lambda_3+\varepsilon)-2)^2 + (\nu_2^3/(\lambda_3+\varepsilon)-4)^2 - 1] \leq 0 \\
& 0 \leq x_1, x_2 \leq 8,\ 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1,\ \varepsilon = 0.0001 \\
& \nu_i^j \geq 0, i = 1,2; j = 1,2,3.
\end{aligned} \tag{6}$$

To avoid division by zero in the nonlinear constraints, ε is introduced as a small tolerance (ε = 0.0001). The optimal solution of (6) is Z^L = 1.154 and x^L = (3.195,1.797). Notice that the lower bound (1.154) is rather tight compared to the optimal solution (1.172). Also, the lower bound of the relaxed big-M MINLP problem (3) is lower than the lower bound of CRP problem (1.031 vs. 1.154). In fact, the relaxation gap of the CRP problem (6) is 1.54%, while the relaxation gap of relaxed MINLP problem (3) is 12.0%.

By letting $z_j = \nu^j/\lambda_j$ from the solution in (6), x^L can be expressed as a convex combination of z_j

with weight λ_j as shown in Figure 1 ($x = \sum_j \lambda_j z_j$). Two important points are noted. (i) Each z_j lies at the boundary of each feasible region S_j when λ_j is nonzero. This means that all the relaxed nonlinear constraints in (6) are active. The objective function value at each z_j, $f(\nu^j/\lambda_j) + \gamma_j$, yields an upper bound to the local solution of that feasible region (see Table 1). (ii) Each λ_j shows how close the optimal point x^L is to each feasible region S_j (the larger λ_j is, the closer x^L is to S_j). From this information, a good guess is that the global optimal solution of GDP problem is in S_j which has the largest λ_j (see z_2 in Figure 1). Therefore, when we apply a branch and bound method to GDP problem, λ_j can be used as an indicator showing which Boolean variable should be selected as a branching variable at the next node in the search tree.

SOLUTION METHODS

Having derived the nonlinear convex relaxation problem (CRP), there are two major solution approaches one can take. The simplest and most direct one is to reformulate (CRP) as the following MINLP problem (|K| = 1).

$$
\begin{aligned}
\min Z = \sum_{j \in J} \gamma_j \lambda_j + f(x) \\
s.t. \quad r(x) \le 0 \\
x = \sum_{j \in J} \nu^j \\
\sum_{j \in J} \lambda_j = 1 \\
(\lambda_j + \varepsilon) g_j(\nu^j / (\lambda_j + \varepsilon)) \le 0, \; j \in J \\
0 \le \nu^j \le \lambda_j U_j, \; j \in J \\
A\lambda \le a \\
x, \nu^j \ge 0, \lambda_j \in \{0,1\}, \; j \in J
\end{aligned}
\tag{P2}
$$

Again, the tolerance ε (e.g. 0.0001) is introduced in the constraints to avoid division by zero, and an additional inequality in terms of a valid upper bound U_j has been introduced to ensure that $\nu^j = 0$ when $\lambda_j = 0$. Lee and Grossmann (1999) proved that the lower bound predicted by the relaxation of problem (P2) is greater than or equal to the lower bound predicted by the relaxation of the MINLP counterpart as given by problem (BM).

Problem (P2) can be solved with any standard method for MINLP problem discussed in the introduction section (e.g., Branch and Bound, Outer-Approximation, Generalized Benders Decomposition, and Extended Cutting Plane). The other alternative is to develop a specific branch and bound method that exploits more directly the property of the convex hull as will be discussed in the next section.

A BRANCH AND BOUND ALGORITHM WITH CONVEX RELAXATION

A branch and bound algorithm with the proposed nonlinear convex relaxation in (CRP) is outlined in this section for the case of only one disjunction ($|K| = 1$).

First, the CRP problem of the given GDP problem is solved. The branching rule that can be used is to select the variable λ_j closest to 1 because this corresponds to the disjunctive term that is closest to being feasible. By solving the corresponding NLP subproblem, this generally yields a good upper bound as was shown in the illustrative example 1.

After branching on one term of disjunction, we propose to select the convex hull of the remaining terms of the disjunction $j' \in J, j' \neq j$, which have not been examined yet. For the case of one disjunction ($|K| = 1$), this corresponds to the dichotomy:

- *either fix* S_j *or fix* $\text{conv}(\bigcup_{\substack{j' \in J \\ j' \neq j}} S_{j'})$

This means that either the solution is in the subregion S_j, or else somewhere in the convex hull of the remaining set of subregions $S_{j'}$, $j' \in J, j' \neq j$. As will be shown later with the results, this branching rule is generally very effective.

Based on the above idea, the main steps of the proposed branch and bound algorithm for $|K| = 1$ are as follows (see Figure 3):

Branch and bound algorithm for Generalized Disjunctive Programming, $|K| = 1$.

Step 0. Initialization

(a) Set $Z^* = \infty$.

(b) Select ε.

(c) Set $T = J$.

Step 1. CRP problem

(a) Solve the CRP problem with the convex hull of S_j, $j \in T$.

(b) Obtain the optimal objective value Z^L and the optimal point x^L.

(c) If all λ_j are 0 or 1, then x^L is a feasible solution to the GDP problem (P1). The global optimal solution is Z^L and x^L. Set $Z^* = Z^L$ and $x^* = x^L$. Exit.

(d) Otherwise, x^L lies outside all S_j (j=1,2,…,m). Go to step 2.

Step 2. Branch on one term

(a) Select Y_j with the largest λ_j ($\neq$ 0, 1) in the solution of CRP problem.

(b) Set $Y_j = true$ and $Y_{j'} = false, j' \neq j$ (Fix λ_j as 1).

(c) Solve GDP problem (P1) with fixed Y_j.

(d) Obtain the optimal objective value Z^U and optimal point x^U.

(e) Set $T = T \setminus j$.

(f) If $Z^U \leq Z^*$, then set $Z^* = Z^U$ and $x^* = x^U$.

Step 3. Check the remaining terms

(a) If T is empty, then exit.

(b) Else if T is not empty, then go to step 4.

Step 4. Branch on the remaining terms

(a) Fix λ_j as 0 (remove S_j from convex hull).

(b) Solve the CRP problem with the convex hull of remaining feasible regions ($S_{j'}$, $j' \neq j$, $j' \in T$).

(c) Obtain the optimal objective value Z^L and the optimal point x^L.

(d) If $Z^L \geq Z^U$, then exit. The global optimal solution is Z^U, x^U.

(e) Else if $Z^L < Z^U$, then go to step 2.

REMARKS

The above algorithm has obviously finite convergence since the number of terms in the disjunction is finite. Also, since the nonlinear functions are convex, the subproblems have a unique optimal solution. Hence, the rigorous validity of the bounds can be guaranteed, with which the branch and bound method is in turn guaranteed to obtain the global optimum. Furthermore, given the strength of the relaxation one can also in general expect the enumeration of fewer nodes.

An important point that is worth noting in the proposed branch and bound enumeration is the case when the proposed branching rule does not yield a true partition. This may arise as follows. After searching one particular feasible subregion j, the convex hull of the remaining feasible subregions ($S_{j',\, j' \neq j}$) generally yields an increase to the lower bound. However, if this lower bound is the same as before, there is the need to verify whether partitioning has in fact taken place. This can be done by the following test. If $x^L \notin \text{conv}(\cup S_{j',\, j' \neq j})$, where x^L is the optimal solution of the parent node, then this is a 'partitionable set' (see Figure 4(a)). If $x^L \in \text{conv}(\cup S_{j',\, j' \neq j})$, then the set of subregions is a 'non-partitionable set' (see Figure 4(b)) because the point x^L remains feasible in the convex hull of the subregion j' and hence the lower bound remains the same after branching. This test step can be used in the proposed algorithm to accelerate the search by avoiding repeated identical lower bounds.

EXAMPLE 1 CONTINUED

Figure 5 shows the corresponding search tree when we apply the proposed branch and bound algorithm to Example 1. At the root node, the search set is $T = \{1, 2, 3\}$ and CRP problem (6) yields a lower bound $Z^L = 1.154$. This optimal point x^L lies outside the feasible region of GDP problem (2) since x^L does not satisfy any term in the disjunction (see Figure 1). Hence, this solution is infeasible for problem (2). Among the weights, λ_2 has the largest value as seen in Table 1, so we select Y_2 and set Y_2 as true. At the first node, the GDP problem is solved with fixed $Y = (false, true, false)$. Only Y_2 is set as true and the other $Y_{j'}$ are set as false. It means that we fix λ_2 as 1 and other λ_j as 0 in problem (6). Therefore, the feasible region is restricted to S_2 only. Solving problem (2) with $Y_2 = true$ yields an upper bound $Z^U = 1.172$. Since S_2 has been examined, it is removed from the search set, $T = \{1,2,3\} \backslash \{2\} = \{1,3\}$. At the second node, we consider the convex hull of S_1 and S_3. The CRP problem is then problem (6) without the constraints and variables for S_2. By solving this CRP problem, a lower bound $Z^L = 3.327$ is obtained. Since this lower bound 3.327 is greater than the upper bound $Z^U = 1.172$, the feasible solution of S_1 and S_3 will be greater than $Z^L = 3.327 > Z^U = 1.172$. Hence, the global optimal solution is $Z^U = 1.172$ and the search ends after 3 nodes.

Table 2 shows the comparison among the standard BB, OA, GBD, and ECP algorithms applied to the big-M MINLP formulation (3), and the reformulated MINLP (PR) in (6). The standard BB finds the optimal solution in 5 nodes (see Figure 6), and other algorithms show almost the same number of major iterations in solving both the big-M and the CRP formulations. However, the convex hull predicts tighter lower bound than the relaxation of the big-M formulation. Note that OA requires 2 MILP and 2 NLP subproblems to solve (PR) compared with 3MILP and 3NLP subproblems to solve (BM). GBD solves 3 MILP and 3 NLP subproblems for both formulations. ECP solves 19 and 20 MILPs, respectively.

GENERALIZATION TO MULTIPLE DISJUNCTIONS

For the case of multiple disjunctions ($|K| > 1$) as in problem (P), the MINLP reformulation of (P2) can be readily generalized as follows:

$$
\begin{aligned}
\min Z = & \sum_{k\in K}\sum_{j\in J_k}\gamma_{jk}\lambda_{jk} + f(x) \\
s.t. \quad & r(x) \le 0 \\
& x = \sum_{j\in J_k} \nu^{jk}, \quad k \in K \\
& \sum_{j\in J_k} \lambda_{jk} = 1, \quad k \in K \\
& (\lambda_{jk}+\varepsilon) g_{jk}(\nu^{jk}/(\lambda_{jk}+\varepsilon)) \le 0, \; j\in J_k, \; k \in K \\
& 0 \le \nu^{jk} \le \lambda_{jk} U_{jk}, \; j \in J_k, \; k \in K \\
& A\lambda \le a \\
& x, \nu^{jk} \ge 0, \lambda_{jk} \in \{0,1\}, \; j \in J_k, \; k \in K
\end{aligned}
\qquad \text{(PR)}
$$

where the tolerance ε is also introduced in the nonlinear inequalities as in (6). The dimension of the variables in (PR) increases due to the double indices in ν^{jk} and λ_{jk}. Similarly, as in the case of (P2), MINLP methods such as Branch and Bound, Outer-Approximation, Generalized Benders Decomposition, and Extended Cutting Plane can be applied to solve problem (PR). Also, the relaxation of this problem yields stronger lower bounds than the relaxation of problem (BM).

As for the proposed branch and bound, the solution procedure of the GDP problem with a number of disjunctions, $|K| \ge 1$, can be easily generalized (see Lee and Grossmann, 1999). In this algorithm, we solve the relaxation of problem (PR), which allows λ_{jk} to be continuous between 0 and 1. This relaxed problem of (PR) corresponds to the CRP problem of GDP problem (P). The global optimal solution is the best upper bound after termination of the branch and bound enumeration.

Relation to logic-based Outer-Approximation Method

Türkay and Grossmann (1996) proposed a Logic-Based Outer-Approximation algorithm when the GDP problem (P) is applied to process networks. In this case the GDP has the following form,

$$\min Z = \sum_{k \in K} c_k + f(x)$$
$$s.t. \quad r(x) \le 0$$
$$\begin{bmatrix} Y_k \\ g_k(x) \le 0 \\ c_k = \gamma_k \end{bmatrix} \vee \begin{bmatrix} \neg Y_k \\ B^k x = 0 \\ c_k = 0 \end{bmatrix}, k \in K \qquad \text{(DP)}$$
$$\Omega(Y) = True$$
$$x,\ c_k \ge 0,\ k \in K$$

which in contrast to (P), has only two terms in each disjunction to denote the existence (Y_k) or non-existence ($\neg Y_k$) of units. In problem (DP) B^k is a matrix which forces the subset of variables x_Z to zero when Y_k is *false*. As shown in Appendix B, an interesting point is that applying the outer-approximation method to the MINLP reformulation (PR) reduces to the logic-based outer-approximation method by Türkay and Grossmann (1996). The reason is that the master problem for both methods becomes identical.

NUMERICAL RESULTS

In this section, we present the comparison of the proposed branch and bound algorithm with standard branch and bound algorithm. Both algorithms use a depth-first search rule. All the example problems were solved with GAMS (Brooke et al., 1997) on a 300MHz Pentium II PC. The GAMS/CONOPT NLP solver was used in both algorithms and comparisons were also performed with GAMS/DICOPT++.

EXAMPLE 2

The corresponding GDP problem taken from Grossmann and Kravanja (1997) is given by (7). The global optimal solution is $Y^* = (false, true, false)$, $x^* = (1,1)$, and $Z^* = 3.5$.

$$
\begin{aligned}
&\min Z = c + x_1^2 + x_2^2 \\
&s.t. \quad (x_1 - 2)^2 - x_2 \le 0 \\
&\begin{bmatrix} Y_1 \\ x_1 - 2 \ge 0 \\ x_1 - x_2 \le 4 \\ c = 1 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ x_1 - x_2 \le 0 \\ x_1 - 1 \ge 0 \\ x_2 - 1 \ge 0 \\ c = 1.5 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ x_1 - x_2 \le 4 \\ x_1 + x_2 \ge 3 \\ x_1 - 1 \ge 0 \\ c = 0.5 \end{bmatrix} \\
&0 \le x_1 \le 4,\ 0 \le x_2 \le 4,\ c \ge 0 \\
&Y_j \in \{true, false\},\ j = 1,2,3.
\end{aligned} \tag{7}
$$

As seen in Table 3, the solution of problem (7) with a standard branch and bound algorithm applied to the big-M MINLP formulation (Grossmann and Kravanja, 1997) predicts a lower bound of 2.532 and requires 5 nodes for the termination. In contrast, by applying the proposed specialized branch and bound method to the GDP in (7) only 3 nodes are required, which is largely due to the improved lower bound of 2.906 that is predicted.

EXAMPLE 3: JOBSHOP SCHEDULING PROBLEM

The next example is scheduling problem with 3 jobs and 3 stages (Raman and Grossmann, 1994). The objective of the following GDP model is to minimize the makespan:

$$
\begin{aligned}
&\min Z = T \\
&s.t. \quad \mathrm{T} \ge x_1 + 8 \\
&\qquad \mathrm{T} \ge x_2 + 5 \\
&\qquad \mathrm{T} \ge x_3 + 6 \\
&\begin{bmatrix} Y_1 \\ x_1 - x_3 + 5 \le 0 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ x_3 - x_1 + 2 \le 0 \end{bmatrix} \\
&\begin{bmatrix} Y_2 \\ x_2 - x_3 + 1 \le 0 \end{bmatrix} \vee \begin{bmatrix} \neg Y_2 \\ x_3 - x_2 + 6 \le 0 \end{bmatrix} \\
&\begin{bmatrix} Y_3 \\ x_1 - x_2 + 5 \le 0 \end{bmatrix} \vee \begin{bmatrix} \neg Y_3 \\ x_2 - x_1 \le 0 \end{bmatrix} \\
&T, x_1, x_2, x_3 \ge 0 \\
&Y_k \in \{true, false\},\ k = 1,2,3.
\end{aligned} \tag{8}
$$

The optimal solution has a makespan of Z^* = 11 hours, Y^* = (*false,true,false*), and x^* = (3,0,1). This optimal schedule is shown in Figure 7. When the model in (8) is converted into a big-M MILP (BM), a standard BB solves this problem in 13 nodes. In contrast, the proposed BB solves the GDP in (8) in only 5 nodes. In this case the lower bound from the linear convex hull (8.162) is not much tighter than that of big-M relaxation (8.000). However, the proposed algorithm reduces the number of nodes significantly.

EXAMPLE 4: PROCESS NETWORK SUPERSTRUCTURE

This example was originally proposed by Duran and Grossmann (1986) as an MINLP problem. The model is given as a GDP problem by Türkay and Grossmann (1996).

$$\min Z = \sum_{k=1}^{8} c_k + a^T x + 122$$

$$s.t.$$

$$x_1 - x_2 - x_4 = 0,\ x_6 - x_7 - x_8 = 0,\ \ x_3 + x_5 - x_6 - x_{11} = 0$$

$$x_{13} - x_{19} - x_{21} = 0,\ x_{17} - x_9 - x_{16} - x_{25} = 0,\ \ x_{11} - x_{12} - x_{15} = 0$$

$$x_{23} - x_{20} - x_{22} = 0,\ x_{23} - x_{14} - x_{24} = 0$$

$$x_{10} - 0.8x_{17} \le 0, \quad x_{10} - 0.4x_{17} \ge 0$$

$$x_{12} - 5x_{14} \le 0, \quad x_{12} - 2x_{14} \ge 0$$

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

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

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

$$\begin{bmatrix} Y_4 \\ 1.25(x_{12} + x_{14}) - x_{13} = 0 \\ c_4 = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_4 \\ x_{12} = x_{13} = x_{14} = 0 \\ c_4 = 0 \end{bmatrix}$$

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

$$\begin{bmatrix} Y_6 \\ \exp(x_{20} / 1.5) - 1 - x_{19} = 0 \\ c_6 = 7 \end{bmatrix} \vee \begin{bmatrix} \neg Y_6 \\ x_{19} = x_{20} = 0 \\ c_6 = 0 \end{bmatrix} \qquad (9)$$

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

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

$$a^T = [0,1,-10,1,-15,0,0,0,-40,15,0,0,0,15,0,\ 0,80,-65,25,-60,35,-80,0,0,-35]$$

$$x_j, c_k \ge 0,\ \ Y_k \in \{true, false\}, k = 1,2,...,8;\ j = 1,2,...,25.$$

Logic Propositions:

$$
\begin{aligned}
&Y_1 \Rightarrow Y_3 \vee Y_4 \vee Y_5 \\
&Y_2 \Rightarrow Y_3 \vee Y_4 \vee Y_5 \\
&Y_3 \Rightarrow Y_1 \vee Y_2 \\
&Y_3 \Rightarrow Y_8 \\
&Y_4 \Rightarrow Y_1 \vee Y_2 \\
&Y_4 \Rightarrow Y_6 \vee Y_7 \\
&Y_5 \Rightarrow Y_1 \vee Y_2 \\
&Y_5 \Rightarrow Y_8 \\
&Y_6 \Rightarrow Y_4 \\
&Y_7 \Rightarrow Y_4 \\
&Y_8 \Rightarrow Y_3 \vee Y_5 \vee (\neg Y_3 \wedge \neg Y_5)
\end{aligned} \tag{10}
$$

Specifications:

$$
\begin{aligned}
&Y_1 \veebar Y_2 \\
&Y_4 \veebar Y_5 \\
&Y_6 \veebar Y_7
\end{aligned} \tag{11}
$$

The optimal solution is $Z^* = 68.01$, $Y^* = (false,true,false,true,false,true,false,true)$, and $x^* = (4.294,0,0,4.294,2,0.667,0.203,0.464,0,0.464,1.333,1.333,2,0.267,0,0,0.58,0.715,2,1.648,0,0,1.648,1.381,0.58)$. Figure 8 shows the superstructure of the process, and Figure 9 shows its optimal configuration. The Boolean variables Y_k denote the existence or non-existence of process 1-8.

As seen in Table 4, the proposed algorithm applied to the GDP in (9)-(11) finds the optimal solution in only 5 nodes. Using the big-M formulation reported by Duran and Grossmann (1986), 17 nodes were required with the standard branch and bound method, and 12 nodes with the branch and cut by Stubbs and Mehrotra (1999). As seen in Figure 10, we first consider the convex hull of each of the eight disjunctions and the weight of the first term in each disjunction is shown. The relaxed optimum objective obtained at the root node, 62.48, is quite close to the optimal solution 68.01 of GDP problem. Since the second term in each disjunction sets a subset of the variables to zero, we consider only the weight of the first term in each disjunction (λ_{1k}). The eight weights shown in Figure 10 correspond to each Boolean variable Y_k. At the root node, λ_{12} has the largest fractional value. At the first node, we fix the first term of the second disjunction (set Y_2 as true) and consider the convex hull in each of the remaining seven disjunctions, k = 1,3,4,5,6,7,8. After fixing $\lambda_{12} = 1$, the optimal solution at the first node has only one fractional weight, λ_{13}. So λ_{13} is selected and fixed as 1 at the second node. The solution of the second node yields the upper bound 71.79. After backtracking, the global optimal solution, 68.01, is obtained at the third node. At the fourth node, a lower bound 75.01 is obtained with fractional λ_{1k}. Since this lower bound is greater than the current upper bound of 68.01, the search stops.

Table 5 shows the comparison with other algorithms when the problem (9)-(11) is reformulated as the MINLP problem (PR) with the convex hull representation for the disjunctions. 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 of (PR). The difference in the number of nodes lies in the branching rules. The OA method requires 3 major iterations and the first relaxed solution is lower than that of BB method. OA, GBD and ECP start with initial guess Y^0 = [1,0,1,1,0,0,1,1]. Note that in the GBD and OA methods, one major iteration consists of one NLP subproblem and one MILP master problem. Again, the proposed algorithm yields the tightest lower bound and requires the fewest number of subproblems.

For comparison, the logic-based OA method by Türkay and Grossmann (1996) yields the lower bound 67.9 with 3 initial NLP subproblems (NLPS) and finds the optimal solution (68.01) in one major iteration requiring a total of 4 NLP and 2 MILP's. The proposed method requires 5 NLP's.

EXAMPLE 5: OPTIMAL POSITIONING OF A NEW PRODUCT

The fifth example problem consists in determining the optimal positioning of a new product in a multiattribute space (Duran and Grossmann, 1986). Here we consider a market with a set of existing products (M) and a set of consumers (N). The existing products can be located in a multiattribute space of dimension K with coordinates δ_{jk}, $j = 1,\ldots, M$, $k = 1,\ldots, K$. Each consumer is characterized by an ideal point z_{ik}, and a set of weights w_{ik}, $i = 1,\ldots, N$, $k = 1,\ldots, K$ both representing consumer's concept of an ideal product. A region which defines closeness to the ideal point for each consumer can be determined in terms of the existing products. Based on these criteria, a consumer is assumed to select a product closest to the ideal point. The objective is to design the optimal location of the product, x_k, $k = 1,\ldots, K$ to maximize the profit. The revenue of the new product from sales to consumer i is given c_i, and $f(x)$ is the cost of reaching locations of the new product within an attribute space. This example was formulated as an MINLP problem by Duran and Grossmann (1986), and can be expressed as the following GDP problem.

$$\min Z = -\sum_{i=1}^{25} c_i - f(x)$$

$$
\begin{aligned}
s.t. \quad & x_1 - x_2 + x_3 + x_4 + x_5 \le 10 \\
& 0.6x_1 - 0.9x_2 - 0.5x_3 + 0.1x_4 + x_5 \le -0.64 \\
& x_1 - x_2 + x_3 - x_4 + x_5 \ge 0.69 \\
& 0.157x_1 + 0.05x_2 \le 1.5 \\
& 0.25x_2 + 1.05x_4 - 0.3x_5 \ge 4.5 \\
& \begin{bmatrix} Y_i \\ \sum_{k=1}^{5} w_{ik}(x_k - z_{ik})^2 \le R_i^2 \\ c_i = p_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ x_k \ge 0, k = 1,2,...,5 \\ c_i = 0 \end{bmatrix}, i = 1,...,25 \\
& f(x) = -0.6x_1^2 + 0.9x_2 + 0.5x_3 - 0.1x_4^2 - x_5 \\
& Y_i \in \{true, false\}, i = 1,...,25 \\
& a \le x \le b, a = [2,0,3,0,4], b = [4.5,8,9,5,10] \\
& R_i^2 = \min_{j=1,...,10} \{\sum_{k=1}^{5} w_{ik}(\delta_{jk} - z_{ik})^2\}, i = 1,...,25 \\
& p^T = [1,0.2,1,0.2,0.9,0.9,0.1,0.8,1,0.4,1,0.3,0.1,0.3,0.5,0. \\
& \quad 9,0.8,0.1,0.9,1,1,1,0.2,0.7,0.7]
\end{aligned} \tag{12}
$$

The data for 10 existing products, 25 consumers, 5 attributes, and the GDP problem are shown in Appendix C. The optimal solution is Z^* = -8.064, $Y^*_{1,6,8,15,17,20,25}$ = *true*, and x^* = (2,7.792,6.056,3.573,4).

The lower bound obtained from the relaxation of model (PR) (-8.685) is much closer to the optimal solution (-8.064) than that of the big-M formulation (BM) by Duran and Grossmann (1986) (-19.10) (see Table 6). The tightness of the lower bound substantially reduces the number of nodes in the BB algorithm (89 vs. 11). Also, model (PR) is solved in 3 major iterations by the OA algorithm compared with 9 of model (BM).

EXAMPLE 6: DESIGN OF A MULTI-PRODUCT BATCH PLANT

The last example is a batch plant design with multiple units in parallel and intermediate storage tanks (Ravemark, 1995; Vecchietti and Grossmann, 1999). This problem consists of determining the volume of the equipment, the number of units in parallel, and the volume and location of the intermediate storage tanks. The objective is to minimize the investment cost. The GDP model is as follows:

$$\min Z = \sum_{j=1}^{J} 250\exp(n_j + m_j + 0.6v_j) + \sum_{j=1}^{J-1} 150\exp(0.5\hat{v}_j)$$

$$s.t.$$

$$v_j \geq \log(S_{ij}) + b_{ij} - n_j, i \in I; j \in J$$

$$e_i \geq \log(P_{ij}) - b_{ij} - m_j, i \in I; j \in J$$

$$H \geq \sum_{i=1}^{I} Q_i \exp(e_i)$$

$$m_j = \sum_{z=1}^{4} \log(z) Y1_{zj}, \sum_{z=1}^{4} Y1_{zj} = 1, j \in J$$

$$n_j = \sum_{z=1}^{4} \log(z) Y2_{zj}, \sum_{z=1}^{4} Y2_{zj} = 1, j \in J$$

$$\begin{bmatrix} Y_j \\ \phi \geq b_{ij} - b_{ij+1} \geq -\phi, i \in I \\ \hat{v}_j \geq \log(10) + \mu(b_{ij} - b_{ij+1}) + b_{ij+1}, i \in I \\ \hat{v}_j \geq \log(10) + \mu(b_{ij+1} - b_{ij}) + b_{ij}, i \in I \end{bmatrix} \vee \begin{bmatrix} \neg Y_j \\ b_{ij} = b_{ij+1}, i \in I \\ \hat{v}_j = \log(100) \end{bmatrix}, j = 1, \ldots J-1 \qquad (13)$$

$$0 \leq m_j \leq \log(4),\ 0 \leq n_j \leq \log(4), j \in J$$

$$\log(V_j^L) \leq v_j \leq \log(V_j^U),\ \log(\hat{V}_j^L) \leq \hat{v}_j \leq \log(\hat{V}_j^U), j \in J$$

$$\log((B_{ij})^L) \leq b_{ij} \leq \log((B_{ij})^U), i \in I; j \in J$$

$$\phi = \log(3), \mu = 1.0$$

$$Y_j \in \{true, false\}, Y1_{zj}, Y2_{zj} \in \{0,1\}, j \in J; z = 1,2,3,4$$

$$J = \{1,2,3,4,5,6\}, I \in \{A, B, C, D, E\}$$

The disjunctions correspond to the storage tank volume and the batch size. The objective function is nonlinear and convex, and the convex hull is linear because the constraints in the disjunctions are linear. The data and the optimal solution are shown in Appendix C. The optimal structure, which has a cost of 261,883, is shown in Figure 11. The relaxation gap is 14.4% (PR) vs. 16.2% (BM). The proposed BB significantly reduces the number of nodes by 81% (391 vs. 73) compared to the standard BB (see Table 6). When the OA algorithm is applied to both model (BM) and (PR), it takes more CPU time to solve model (PR) than (BM) (40.91 vs. 13.47 sec).

CONCLUSION

A novel solution algorithm has been proposed for GDP problems which correspond to discrete/continuous optimization problems that involve disjunctions with nonlinear inequalities and logic propositions. A new nonlinear relaxation of the GDP problem and its properties have been presented. The proposed relaxation problem (CRP) of the GDP problem is based on the convex hull of each nonlinear disjunction, and is used for the reformulation of the GDP problem as the MINLP problem (PR), which can be solved with MINLP algorithms such as BB, OA, GBD, and ECP. The relation of problem (PR) with the logic based outer-approximation algorithm by Türkay and Grossmann was established. A special purpose branch and bound algorithm for the GDP problem was also proposed based on the CRP problem.

The numerical results of six GDP problems showed that the proposed branch and bound algorithm, which makes use of the relaxation (CRP), requires fewer nodes and less CPU time than the standard branch and bound method which makes use of the big-M relaxation. These GDP problems were also reformulated as the MINLP problem (PR), and solved by existing MINLP algorithms.

Acknowledgments-The authors would like to acknowledge financial support from the NSF Grant CTS-9710303 and partial support from Eastman Chemical Company.

REFERENCES

Balas E., Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM J. Alg. Disc. Meth.* **6**. 466-486, 1985.

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., D. Kendrick, A. Meeraus and R. Raman, GAMS Language Guide, Release 2.25, Version 92. GAMS Development Corporation, 1997.

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

Duran M.A. and I.E. Grossmann, An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Mathematical Programming,* **36**, 307-339, 1986.

Fletcher R. and S. Leyffer, Solving Mixed Nonlinear Programs by Outer Approximation.

Mathematical Programming, **66**(3), 327-349, 1994.

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

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

Grossmann I.E. and Z. Kravanja, Mixed-Integer Nonlinear Programming: A Survey of Algorithms and Applications*, Large-Scale Optimization with Applications, Part II: Optimal Design and Control* (eds. Biegler et al). Springer-Verlag, 73-100, 1997.

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

Hiriart-Urruty J. and C. Lemaréchal, Convex Analysis and Minimization Algorithms, Vol. 1. Springer-Verlag, 1993.

Lee S. and I.E. Grossmann, Generalized Disjunctive Programming: Nonlinear Convex Hull Relaxation and Algorithms, submitted to *Mathematical Programming*, 1999.

Leyffer S., Integrating SQP and branch-and-bound for Mixed Integer Nonlinear Programming, submitted to *Computational Optimization and Applications*, 1999.

Nemhauser G.L. and L.A. Wolsey, *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., 1988.

Quesada I. and I.E. Grossmann, An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers Chem. Engng.,* **16**(10/11), 937-947, 1992.

Raman R. and I.E. Grossmann, Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis. *Computers Chem. Engng.,* **15**(2), 73-84, 1991.

Raman R. and I.E. Grossmann, Modelling and Computational Techniques for Logic Based Integer Programming. *Computers Chem. Engng.,* **18**(7), 563-578, 1994.

Ravemark E., *Optimization models for design and operation of chemical batch processes*. Ph.D. Thesis. 1995.

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

Türkay M. and I.E. Grossmann, Logic-based MINLP Algorithms for the Optimal Synthesis of Process Networks. *Computers Chem. Engng.,* **20**(8), 959-978, 1996.

Vecchietti A and I.E. Grossmann, LOGMIP : A Disjunctive 0-1 Nonlinear Optimizer for Process

Systems Models. *Computers Chem. Engng.,* **23**, 555-565, 1999.

Westerlund T. and F. Petterson, An Extended Cutting Plane Method for Solving Convex MINLP Problems. *Computers Chem. Engng.,* **19**, supl., S131-S136, 1995.

Williams H.P., *Model building in mathematical programming*, John Wiley & Sons, Inc., 1985.

Yuan X, S. Zhang, L. Piboleau and S. Domenech, Une Methode d'optimization Nonlineare en Variables Mixtes pour la Conception de Porcedes. *Rairo Recherche Operationnele*, 22, 331, 1988.

APPENDIX A

Proof of Convex Hull

Theorem 2. The convex hull of the disjunction in (4) is given by

$$\begin{aligned} & x = \sum_{j \in J} \nu^j, \qquad c = \sum_{j \in J} \lambda_j \gamma_j \\ & 0 \le \nu^j \le \lambda_j U_j\,,\ j \in J \\ & \sum_{j \in J} \lambda_j = 1,\ 0 \le \lambda_j \le 1,\ j \in J \\ & \lambda_j g_j(\nu^j / \lambda_j) \le 0\,,\ j \in J \\ & x, \nu^j, c \ge 0\,,\ j \in J \end{aligned} \tag{A.1}$$

Proof. The convex hull of the disjunction in (4) can be expressed as a convex combination of multipliers λ_j that multiply the constraints in the disjunction.

$$\lambda_j g_j(x) \le 0\,, j \in J \tag{A.2}$$

$$c\lambda_j = \gamma_j \lambda_j, j \in J \tag{A.3}$$

$$\sum_{j \in J} \lambda_j = 1,\ 0 \le \lambda_j \le 1, j \in J \tag{A.4}$$

The equation (A.3) can be linearized by setting $c_j = c\lambda_j$, which leads to,

$$c_j = \gamma_j \lambda_j, j \in J \tag{A.5}$$

The inequality (A.2) is generally nonconvex due to the bilinearity that is introduced by the product. We can convexify, however, the inequality by defining the new variable $\nu^j = x\lambda_j$. From the convexity condition of λ_j, the following equations hold.

$$\sum_{j\in J} x\lambda_j = x = \sum_{j\in J} \nu^j \tag{A.6}$$

$$\sum_{j\in J} c\lambda_j = c = \sum_{j\in J} \gamma_j \lambda_j \tag{A.7}$$

Furthermore, rewriting (A.2) in terms of ν^j and λ_j, for $\lambda_j \geq 0$,

$$\lambda_j g_j(\nu^j / \lambda_j) \leq 0 \ , j \in J \tag{A.8}$$

From Hiriart-Urruty and Lemaréchal (1993), the above inequality is convex. From the assumption, ν^j is bounded by

$$0 \leq \nu^j \leq \lambda_j U_j \ , j \in J \tag{A.9}$$

where U_j is an upper bound for each ν^j. Hence, the convex hull is given by the equations and inequalities in (A.1). ■

APPENDIX B

Relation to logic-based Outer-Approximation Method for problem (DP)

In the algorithm by Türkay and Grossmann (1996), which addresses the solution of problem (DP), the NLP subproblem for fixed values of the Boolean variables Y_k^l at iteration l, is given by,

$$\begin{aligned} \min Z &= \sum_{k\in K} c_k + f(x) \\ s.t. \quad & r(x) \leq 0 \\ g_k(x) \leq 0,\ & c_k = \gamma_k \ \ for\ Y_k^l = true \\ B^k x = 0, \quad & c_k = 0 \ \ for\ Y_k^l = false \\ x,\ & c_k \geq 0,\ k \in K \end{aligned} \tag{FX - DP}$$

The outer-approximation master problem is given by the following disjunctive problem in which the nonlinear constraints are linearized at the optimal solutions of problem (FX-DP),

$$\min Z = \sum_{k \in K} c_k + \alpha$$

$$s.t. \left\{ \begin{array}{l} \alpha \geq f(x^l) + \nabla f(x^l)^T (x - x^l) \\ r(x^l) + \nabla r(x^l)^T (x - x^l) \leq 0 \end{array} \right\} for\ l = 1,2,...,L$$

$$\begin{bmatrix} Y_k \\ g_k(x^l) + \nabla g_k(x^l)^T (x - x^l) \leq 0, l \in K_L^k \\ c_k = \gamma_k \end{bmatrix} \vee \begin{bmatrix} \neg Y_k \\ B^k x = 0 \\ c_k = 0 \end{bmatrix}, k \in K \qquad \text{(MP)}$$

$$\Omega(Y) = True$$

$$K_L^k = \{l \mid Y_k^l = true, l = 1,2,...,L\},\ k \in K$$

$$\alpha, x, c_k \geq 0, Y_k \in \{true, false\},\ k \in K$$

The index $l = 1,2,\ldots,L$ corresponds to the iteration counter, while K_L^k is the set of those iterations in which the left term of the k-th disjunction in (DP) is active, thus yielding a linear approximation for the inequality $g_k(x) \leq 0$. Problem (MP) can be transformed into the following MILP problem by using the convex hull of each disjunction with linearized constraints (see equation (5)):

$$\min Z = \sum_{k \in K} \gamma_k y_k + \alpha$$

$$s.t. \left\{ \begin{array}{l} \alpha \geq f(x^l) + \nabla f(x^l)^T (x - x^l) \\ r(x^l) + \nabla r(x^l)^T (x - x^l) \leq 0 \end{array} \right\} for\ l = 1,2,...,L \qquad \text{(DP - MP)}$$

$$\nabla_{x_{Zk}} g_k(x^l)^T x_{Zk} + \nabla_{x_{Nk}} g_k(x^l)^T x_{Nk}^1 \leq [-g_k(x^l) + \nabla_x g_k(x^l)^T x^l] y_k, l \in K_L^k$$

$$x_{Nk} = x_{Nk}^1 + x_{Nk}^2$$

$$0 \leq x_{Nk}^1 \leq x_{Nk}^U y_k,\ 0 \leq x_{Nk}^1 \leq x_{Nk}^U (1 - y_k)$$

$$Ay \leq a$$

$$K_L^k = \{l \mid Y_k^l = true, l = 1,2,...,L\},\ k \in K$$

$$\alpha, x \geq 0, y_k \in \{0,1\},\ k \in K$$

where x_{Nk} is the vector of variables which are non-zero when Y_k is false, while x_{Zk} is the vector of variables that takes a value of zero. This partition of the continuous variables x is performed according to the definition of the matrix B^k in (DP).

Since each disjunction must have at least one linearization, several NLP subproblems must be solved initially. The fewest number of such NLP subproblems can be determined from a set covering problem (Türkay and Grossmann, 1996).

For the case of two terms in each disjunction in problem (DP), problem (PR) reduces to,

$$
\begin{aligned}
\min Z = \sum_{k\in K} \gamma_k \lambda_{1k} + f(x) \\
s.t. \qquad r(x) \le 0 \\
x_N = v_N^{1k} + v_N^{2k} \\
\lambda_{1k} + \lambda_{2k} = 1 \\
\lambda_{1k} g_k (x_Z / \lambda_{1k}, v_N^{1k} / \lambda_{1k}) \le 0 \\
0 \le x_Z \le U_{1k}\lambda_{1k} \\
0 \le v_N^{1k} \le U_{1k}\lambda_{1k},\ 0 \le v_N^{2k} \le U_{2k}\lambda_{2k} \\
A\lambda \le a \\
x, v_N^{1k}, v_N^{2k} \ge 0, \lambda_{1k}, \lambda_{2k} \in \{0,1\},\ k \in K
\end{aligned}
\qquad \text{(PRT)}
$$

where $x = [x_Z, x_N]$. Note that the above constraints have been simplified because $\nu_Z^{2k} = 0$ since the corresponding variables x_Z take a value of zero in the second term of the disjunction.

For fixed Y_k^l in (PRT) we have, $\lambda_{1k}^l = 1$ *if* Y_k^l *is true*, $\lambda_{1k}^l = 0$ *if* Y_k^l *is false*, with which problem (PRT) becomes:

$$
\begin{aligned}
\min Z = \sum_{k\in K} \gamma_k \lambda_{1k} + f(x) \\
s.t. \qquad r(x) \le 0 \\
\lambda_{1k}^l = 1 \ \ for\ Y_k^l = true \\
\lambda_{1k}^l = 0 \ \ for\ Y_k^l = false \\
g_k(x_Z, x_N) \le 0 \quad for\ Y_k^l = true \\
0 \le x_Z, x_N \le U_{1k} \ \ for\ Y_k^l = true \\
x_Z = 0, x_N = v_N^{2k} \quad for\ Y_k^l = false \\
0 \le v_N^{2k} \le U_{2k} \quad for\ Y_k^l = false \\
A\lambda \le a \\
x, v_N^{1k}, v_N^{2k} \ge 0,\ k \in K
\end{aligned}
\qquad \text{(FX - PRT)}
$$

It is clear that for fixed Y_k^l the NLP subproblem (FX-PRT) is identical to problem (FX-DP). Rather than linearizing the original constraints as in problem (MP), we linearize the nonlinear convex hull formulation in (PRT) to define the master problem of the outer-approximation algorithm. Then the corresponding MILP master problem is given as follows,

$$\min Z = \sum_{k \in K} \gamma_k \lambda_{1k} + \alpha$$

$$s.t. \begin{Bmatrix} \alpha \geq f(x^l) + \nabla f(x^l)^T (x - x^l) \\ r(x^l) + \nabla r(x^l)^T (x - x^l) \leq 0 \end{Bmatrix} for\ l = 1,2,...,L$$

$$x_N = v_N^{1k} + v_N^{2k}$$

$$\lambda_{1k} + \lambda_{2k} = 1 \qquad \text{(M - PRT)}$$

$$\nabla_{x_Z} g_k \left(\frac{x_Z^l}{\lambda_{1k}^l} \right)^T \left(\frac{x_Z}{\lambda_{1k}} \right) + \nabla_{x_N} g_k \left(\frac{v_N^{1k,l}}{\lambda_{1k}^l} \right)^T \left(\frac{v_N^{1k}}{\lambda_{1k}} \right) \leq$$

$$\left[-g_k \left(\frac{v^{1k,l}}{\lambda_{1k}^l} \right) + \nabla_x g_k \left(\frac{v^{1k,l}}{\lambda_{1k}^l} \right)^T \left(\frac{v^{1k,l}}{\lambda_{1k}^l} \right) \right] \lambda_{1k}$$

$$0 \leq x_Z \leq U_{1k} \lambda_{1k}$$

$$0 \leq v_N^{1k} \leq U_{1k} \lambda_{1k},\ 0 \leq v_N^{2k} \leq U_{2k} \lambda_{2k}$$

$$A\lambda \leq a$$

$$K_L^k = \{l \mid Y_k^l = true, l = 1,2,...,L\},\ k \in K$$

$$x, v_N^{1k}, v_N^{2k} \geq 0, \lambda_{1k}, \lambda_{2k} \in \{0,1\},\ k \in K$$

Note that if we let $x^l = (\nu^{1k,l}/\lambda^l_{1k})$ and treat λ_{1k} as binary variable y_k, then the linearized constraints (M-PRT) and the convex hull of the linear disjunction in problem (DP-MP) are the same. Also, the partition of x in non-zero and zero variables is used in the same way as in (DP-MP). Hence, the MILP problem in (M-PRT) is identical to the MILP master problem (DP-MP) of Türkay and Grossmann (1996). Thus, we can conclude that for the case of problem (DP), applying the outer-approximation method to the MINLP reformulation (PR) reduces to the logic-based outer-approximation method by Türkay and Grossmann (1996).

APPENDIX C

Data for GDP example problems 5 and 6

EXAMPLE 5

i	***Ideal points (z_{ik})***					***Attribute weights (w_{ik})***				
	K =1	***2***	***3***	***4***	***5***	***k =1***	***2***	***3***	***4***	***5***
1	2.26	5.15	4.03	1.74	4.74	9.57	2.74	9.75	3.96	8.67
2	5.51	9.01	3.84	1.47	9.92	8.38	3.93	5.18	5.2	7.82
3	4.06	1.80	0.71	9.09	8.13	9.81	0.04	4.21	7.38	4.11
4	6.30	0.11	4.08	7.29	4.24	7.41	6.08	5.46	4.86	1.48
5	2.81	1.65	8.08	3.99	3.51	9.96	9.13	2.95	8.25	3.58

6	4.29	9.49	2.24	9.78	1.52	9.39	4.27	5.09	1.81	7.58
7	9.76	3.64	6.62	3.66	9.08	1.88	7.2	6.65	1.74	2.86
8	1.37	6.99	7.19	3.03	3.39	4.01	2.67	4.86	2.55	6.91
9	8.89	8.29	6.05	7.48	4.09	4.18	1.92	2.60	7.15	2.86
10	7.42	4.60	0.3	0.97	8.77	7.81	2.14	9.63	7.61	9.17
11	1.54	7.06	0.01	1.23	3.11	8.96	3.47	5.49	4.73	9.43
12	7.74	4.4	7.93	5.95	4.88	9.94	1.63	1.23	4.33	7.08
13	9.94	5.21	8.58	0.13	4.57	0.31	5	0.16	2.52	3.08
14	9.54	1.57	9.66	5.24	7.90	6.02	0.92	7.47	9.74	1.76
15	7.46	8.81	1.67	6.47	1.81	5.06	4.52	1.89	1.22	9.05
16	0.56	8.1	0.19	6.11	6.40	5.92	2.56	7.74	6.96	5.18
17	3.86	6.68	6.42	7.29	4.66	6.45	1.52	0.06	5.34	8.47
18	2.98	2.98	3.03	0.02	0.67	1.04	1.36	5.99	8.10	5.22
19	3.61	7.62	1.79	7.8	9.81	1.40	1.35	0.59	8.58	1.21
20	5.68	4.24	4.17	6.75	1.08	6.68	9.48	1.6	6.74	8.92
21	5.48	3.74	3.34	6.22	7.94	1.95	0.46	2.9	1.79	0.99
22	8.13	8.72	3.93	8.8	8.56	5.18	5.1	8.81	3.27	9.63
23	1.37	0.54	1.55	5.56	5.85	1.47	5.71	6.95	1.42	3.49
24	8.79	5.04	4.83	6.94	0.38	5.4	3.12	5.37	6.1	3.71
25	2.66	4.19	6.49	8.04	1.66	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	3.1	6.66
5	1.26	6.8	2.3	1.75	6.65
6	3.74	9.06	9.8	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	6.8	9.75
10	6.49	1.92	0.05	4.89	6.43

EXAMPLE 6

$$m_j = \log(M_j), n_j = \log(N_j), v_j = \log(V_j), \hat{v}_j = \log(\hat{V}_j), j \in J; \quad b_{ij} = \log(B_{ij}), i \in I; j \in J$$

i = products; j = stages; H = horizon time = 6000 h

Q_i = production rate of product i: A = 250000, B = 150000, C = 180000, D = 160000, E = 120000

S_{ij} = size factor for product i at stage j

i* \ *j	**1**	**2**	**3**	**4**	**5**	**6**
A	7.9	2.0	5.2	4.9	6.1	4.2
B	0.7	0.8	0.9	3.4	2.1	2.5

C	0.7	2.6	1.6	3.6	3.2	2.9
D	4.7	2.3	1.6	2.7	1.2	2.5
E	1.2	3.6	2.4	4.5	1.6	2.1

P_{ij} = processing time of product *i* at stage *j*

i \ *j*	1	2	3	4	5	6
A	6.4	4.7	8.3	3.9	2.1	1.2
B	6.8	6.4	6.5	4.4	2.3	3.2
C	1.0	6.3	5.4	11.9	5.7	6.2
D	3.2	3.0	3.5	3.3	2.8	3.4
E	2.1	2.5	4.2	3.6	3.7	2.2

Optimal Solution: Y^* = {*false,true,false,false,false*}

j	1	2	3	4	5	6
V_j	2491.9	1030	2127	2807	2495	2261
N_j	1	1	1	1	1	1
M_j	2	2	2	2	1	1
V_j (storage)	0	8637.8	0	0	0	0
Cost (Z^*)	261,883					

LIST OF TABLES

Table 1. Disaggregated variables and local optimal points of example 1.

Table 2. Comparison of the results for example 1.

Table 3. Comparison of the results for example 2.

Table 4. Comparison of branch and bound methods for example 4.

Table 5. Comparison of algorithms for formulation (PR) of example 4.

Table 6. Comparison of formulations (BM) and (PR) for GDP examples.

LIST OF FIGURES

Figure 1. Feasible region of example 1.

Figure 2. Convex hull of feasible region.

Figure 3. The proposed branch and bound algorithm, for $|K| = 1$.

Figure 4. Partitionable and Non-partitionable sets.

Figure 5. The proposed branch and bound tree: example 1.

Figure 6. Standard branch and bound tree: example 1.

Figure 7. The optimal schedule for example 3.

Figure 8. Superstructure for example 4.

Figure 9. The optimal structure of example 4.

Figure 10. The proposed branch and bound method for example 4.

Figure 11. The optimal plant structure of example 6.

Table 1. Disaggregated variables and local optimal points of example 1.

Feasible Region	λ_j	$z_j = \nu^j/\lambda_j$	$f(\nu^j/\lambda_j) + \gamma_j$	Local optimal point (x_1, x_2)	Local optimal value
S_1	0.016	(0.000, 1.000)	12.000	(0.832, 0.555)	8.7890
S_2	0.955	(3.306, 1.720)	1.1720	(3.293, 1.707)	1.1716
S_3	0.029	(1.306, 4.720)	13.268	(2.447, 3.106)	4.5279

Table 2. Comparison of the results for example 1.

Formulation	Opt. Solution	Lower Bound[a]	Method	Standard BB	Proposed BB	OA[b]	GBD[c]	ECP[c]
(BM)	1.172	1.013	Major Iter. /Nodes	5	-	3	3	19
(PR)	1.172	1.154	Major Iter. /Nodes	-	3	2	3	20

[a]NLP relaxation. [b]OA begins with NLP relaxation. [c]GBD and ECP begin with initial guess $y^0 = (1,0,0)$.

Table 3. Comparison of the results for example 2.

Method	No. of NLP Subproblems	Lower Bound
Standard BB-formulation (BM)	5	2.532
Proposed BB-formulation (PR)	3	2.906

Table 4. Comparison of branch and bound methods for example 4.

	Method	Standard BB	Branch & Cut	Proposed BB	Optimal Solution
Formulation (BM)	**Nodes**	17	12 (16 Cuts)	**-**	68.01
	Relaxed Opt.	15.08	15.08	**-**	
Formulation (PR)	**Nodes**	-	-	5	68.01
	Relaxed Opt.	-	-	62.48	

Table 5. Comparison of algorithms for formulation (PR) of example 4.

Method*	Standard BB	Proposed BB	OA (Major)	GBD (Major)	ECP	Logic-based OA*
No. of nodes / Iteration	11 (Nodes)	5 (Nodes)	3 (Iter.)	8 (Iter.)	7 (Iter.)	3 subproblem 1 Major Iter.
Relaxed Optimum	62.48	62.48	8.541	-551.4	-5.077	67.9

*All methods except logic-based OA solve the reformulated MINLP problem (PR).

Table 6. Comparison of formulations (BM) and (PR) for GDP examples.

Problem Number	GDP Global Opt.	OA				BB			
		Major It.	CPU sec	Major It.	CPU sec	Nodes	Lower Bound	Nodes	Lower Bound
		(BM)		(PR)		(BM)		(PR)	
2	3.500	3	0.641	3	1.060	5	2.532	3	2.906
4*	68.01	11	3.044	2	1.094	17	15.08	5	62.48
5	-8.064	9	5.817	3	2.863	89	-19.10	11	-8.685
6	261,883	10	13.47	16	40.91	391	219,335	73	224,165

(*with logic propositions)

Figure 1. Feasible region of example 1.

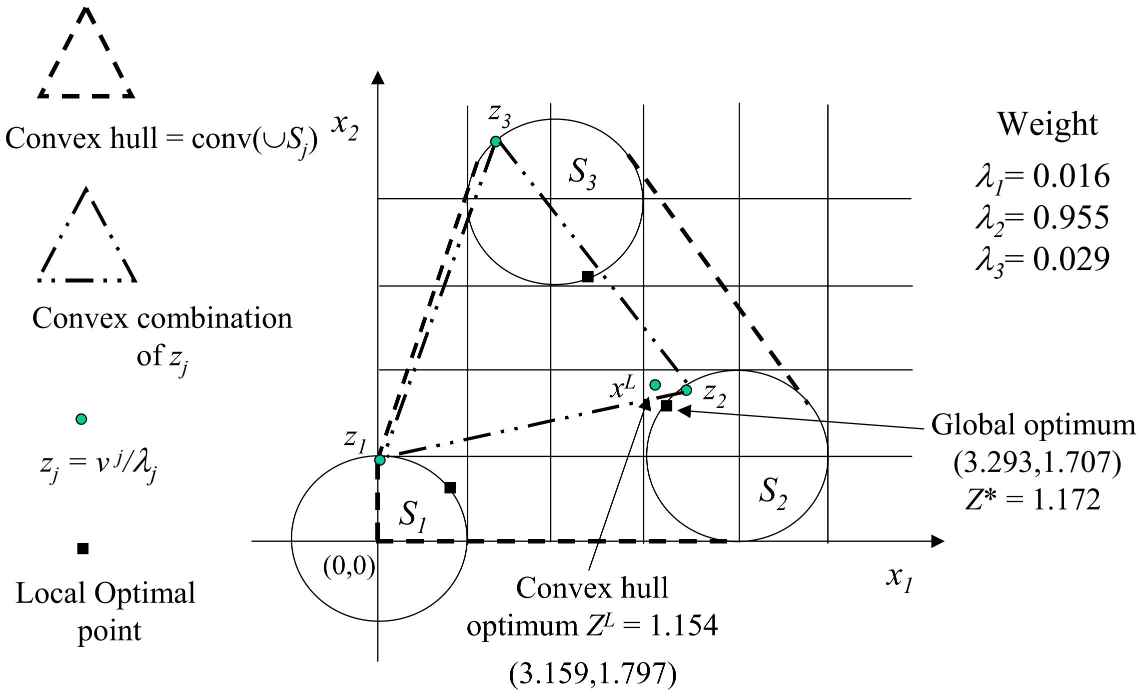


Figure 2. Convex hull of feasible region.

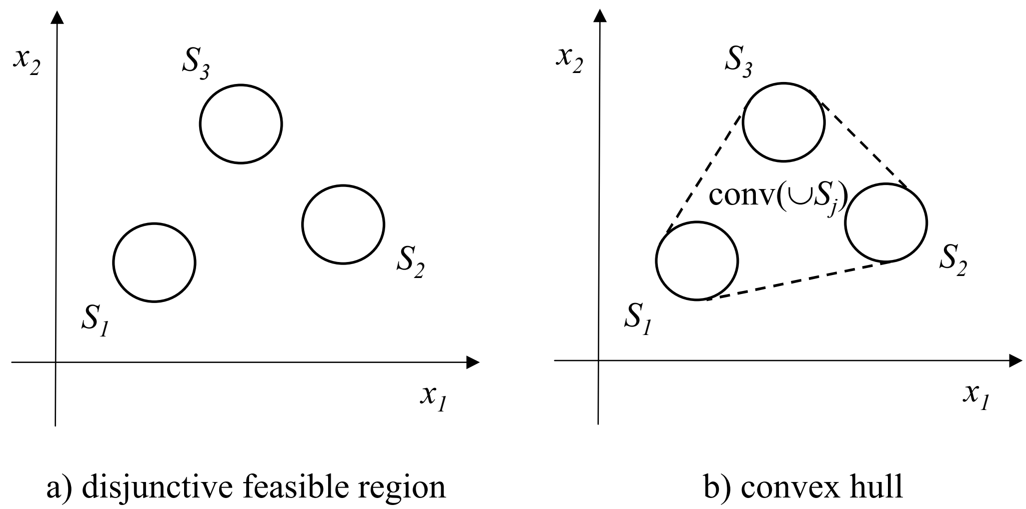


Figure 3. The proposed branch and bound algorithm, for $|K| = 1$.

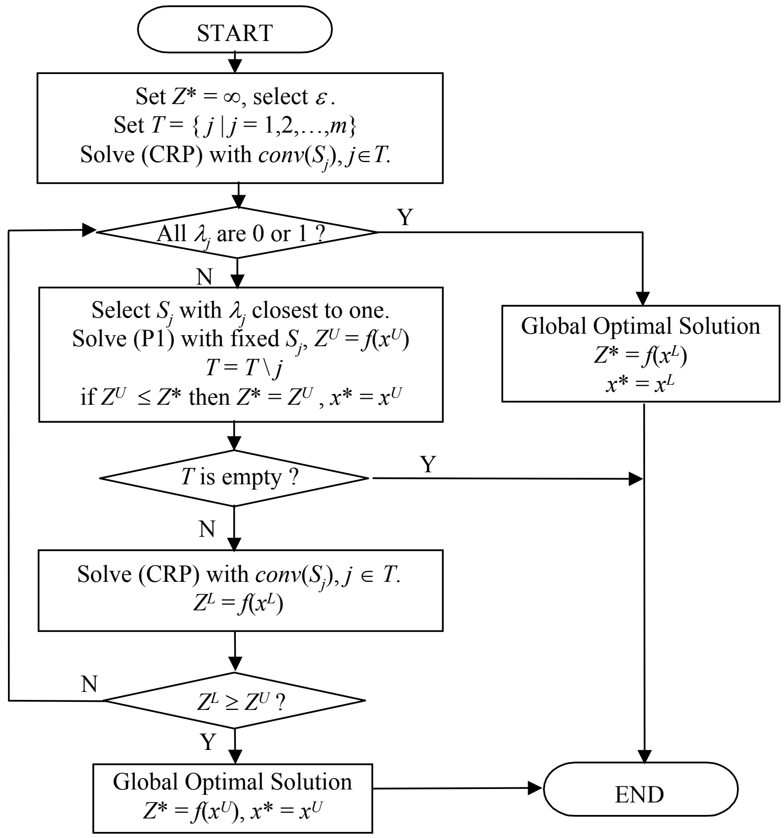
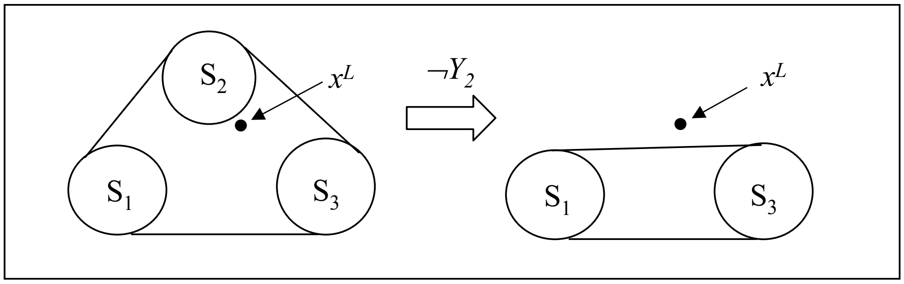
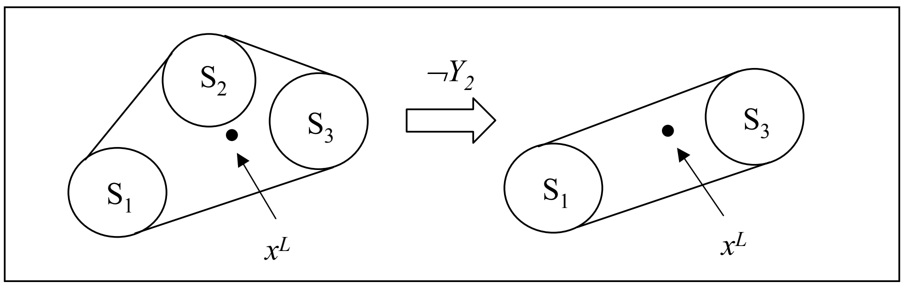


Figure 4. Partitionable and Non-partitionable sets.



a) Partitionable set



b) Non-partitionable set

Figure 5. The proposed branch and bound tree: example 1.

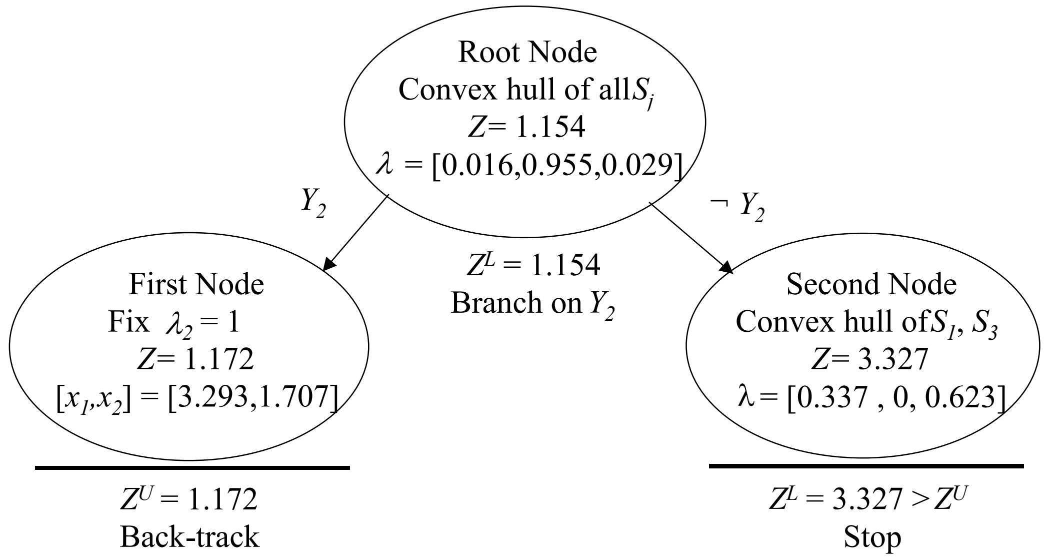


Figure 6. Standard branch and bound tree: example 1.

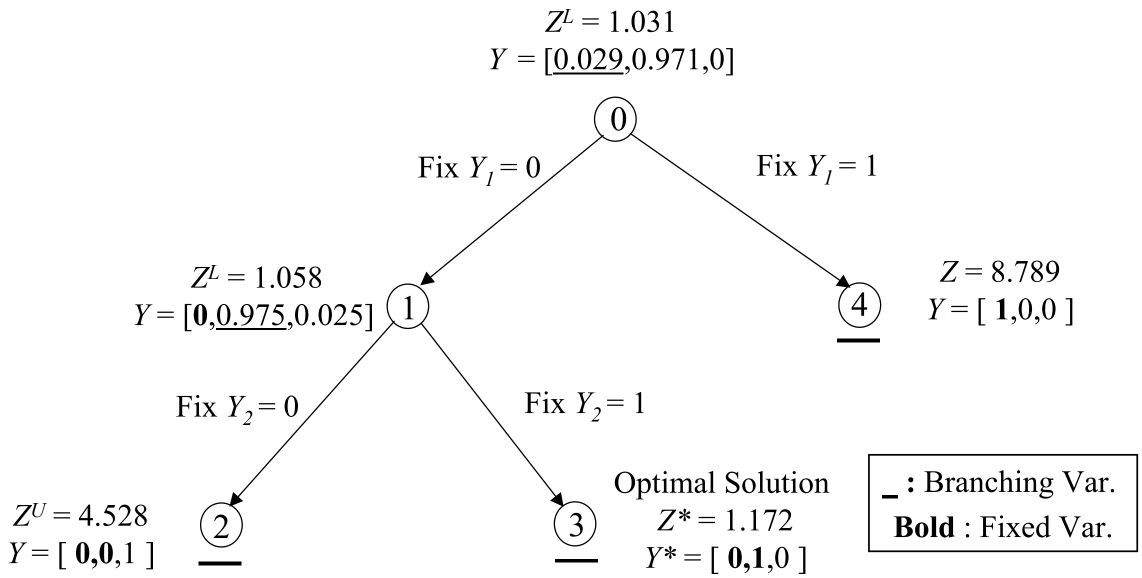


Figure 7. The optimal schedule for example 3.

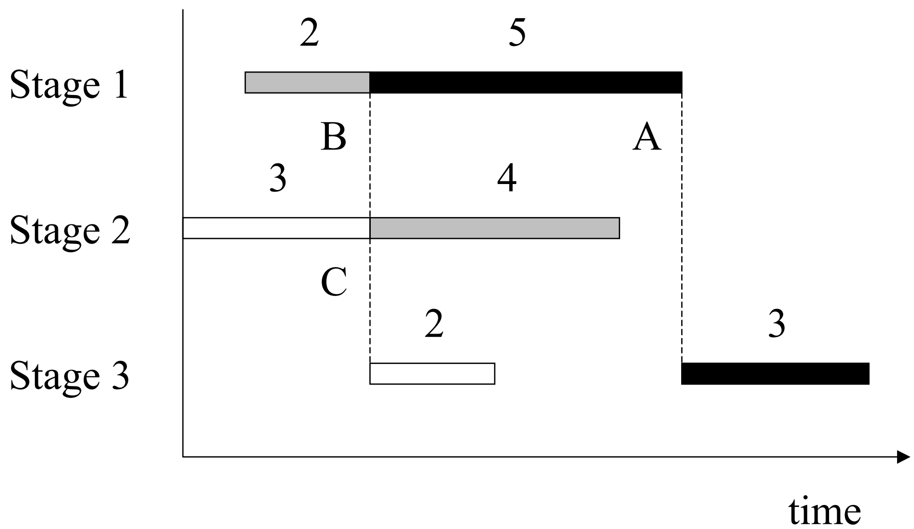


Figure 8. Superstructure for example 4.

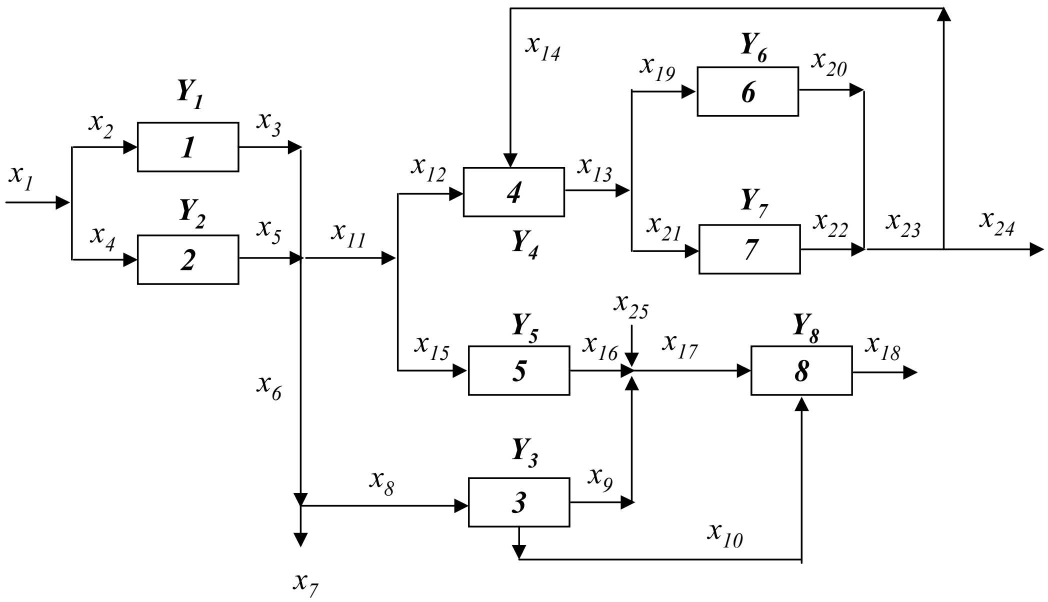


Figure 9. The optimal structure of example 4.

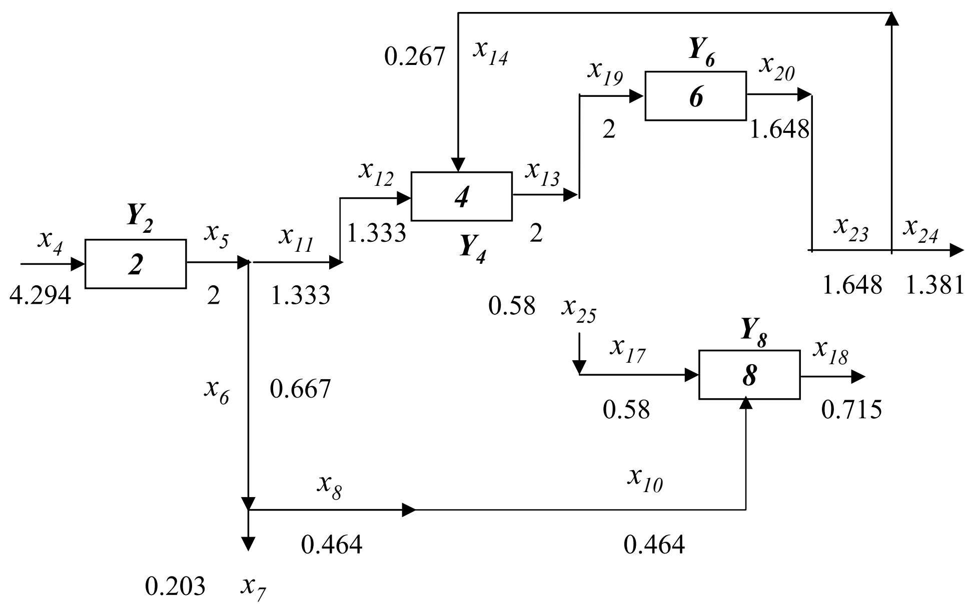


Figure 10. The proposed branch and bound method for example 4.

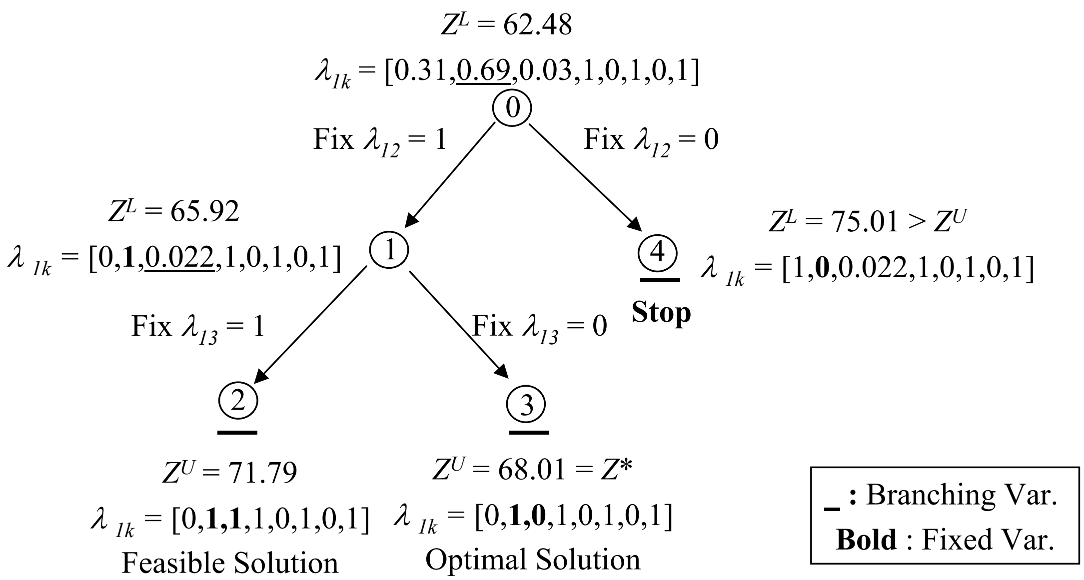


Figure 11. The optimal plant structure of example 6.