# A Global Optimization Algorithm for Nonconvex Generalized Disjunctive Programming and Applications to Process Systems

**Sangbum Lee and Ignacio E. Grossmann***

**Department of Chemical Engineering**

**Carnegie Mellon University**

**Pittsburgh, PA 15213**

## Abstract

A global optimization algorithm for nonconvex Generalized Disjunctive Programming (GDP) problems is proposed in this paper. By making use of convex underestimating functions for bilinear, linear fractional and concave separable functions in the continuous variables, the convex hull of each nonlinear disjunction is constructed. The relaxed convex GDP problem is then solved in the first level of a two-level branch and bound algorithm, in which a discrete branch and bound search is performed on the disjunctions to predict lower bounds. In the second level, a spatial branch and bound method is used to solve nonconvex NLP problems for updating the upper bound. The proposed algorithm exploits the convex hull relaxation for the discrete search, and the fact that the spatial branch and bound is restricted to fixed discrete variables in order to predict tight lower bounds. Application of the proposed algorithm to several example problems is shown, as well as comparisons with other algorithms.

**Keywords**: Nonconvex GDP, nonconvex MINLP, convex hull relaxation, branch and bound, global optimization.

* Author to whom correspondence should be addressed. E-mail address: grossmann@cmu.edu.

## Introduction

Nonlinear discrete/continuous optimization problems can be formulated as Generalized Disjunctive Programming (GDP) models as proposed by Raman and Grossmann (1994). The GDP model involves disjunctions for representing discrete decisions in the continuous space, and logic propositions for the decisions in the discrete space. Lee and Grossmann (2000) have proposed a convex hull relaxation for the GDP model, and solution algorithms based on branch and bound or reformulation. Convex GDP problems can often be reformulated into tight Mixed-Integer Non-Linear Programming (MINLP) problems, which can be solved with a number of MINLP algorithms (Grossmann and Kravanja, 1997). The methods by Lee and Grossmann (2000) for convex GDP problems can be applied to problems involving multiple terms in each disjunction.

The solution of MINLP models involving nonconvex functions has been receiving increased attention due to its practical importance in engineering and many other areas. Due to the nonconvexites, conventional MINLP algorithms are often trapped in suboptimal solutions. There has recently been significant progress in the global optimization of nonconvex NLP problems (for a review, see Floudas, 2000; Horst and Tuy, 1996). Most of the methods proposed for solving these problems rely on the spatial branch and bound method, which is a deterministic algorithm that divides the feasible region of continuous variables and compares the lower bound and upper bound for fathoming each subregion. The subregion that contains the optimal solution is found by eliminating subregions that are proved not to contain the optimal solution. An example of such a method for nonconvex NLP problems is the one by Quesada and Grossmann (1995) who proposed a spatial branch and bound algorithm for concave separable, linear fractional and bilinear programs, and making use of linear and nonlinear underestimating functions.

As for methods for nonconvex MINLP, Ryoo and Sahinidis (1995), and later Tawarmalani and Sahinidis (2000a) have developed a branch and bound method that branches on both the continuous and discrete variables. This method, which relies on bounds reduction and the use of underestimators, has been implemented in BARON. Adjiman et al. (1997; 2000) proposed the SMIN-αBB and GMIN-αBB algorithms for twice-differentiable nonconvex MINLPs. By using a valid convex underestimation of general functions, as well as for special functions, Adjiman and Floudas (1996) developed the αBB method which is a branch and bound procedure that branches on both the continuous and discrete variables according to specific options. The branch-and-contract method (Zamora and Grossmann, 1998b; 1999) for global optimization of process models, which have bilinear, linear fractional, and concave separable functions in the continuous variables and linear 0-1 variables, uses bound contraction and applies the outer-approximation (OA) algorithm at each node of the tree for the spatial search. Kesavan and Barton (2000) developed a generalized branch-and-cut (GBC) algorithm, and showed that their earlier decomposition algorithm (Kesavan and Barton, 1999) is a specific instance of the GBC algorithm with a set of heuristics.

In this paper, we propose a global optimization algorithm for nonconvex GDP problems in which we consider bilinear, linear fractional, and concave separable functions for the continuous variables, and linear functions for the discrete variables. Using valid underestimators from McCormick (1976) and Quesada and Grossmann (1995), the relaxed GDP is reformulated as a convex NLP by the convex hull relaxation as proposed by Lee and Grossmann (2000). To exploit the tight relaxation of this problem, a two-level branch and bound algorithm is proposed. In the first level, a discrete branch and bound is performed on the disjunctions to update the lower bound. In the second level, a spatial branch and bound search is performed for fixed discrete variables to update the upper bound. The proposed method is applied to nonconvex GDP problems that arise in process networks, heat exchanger networks, and the design of batch processes. Numerical results and comparisons with other solution methods are presented.

## Nonconvex GDP Model

Consider the following Generalized Disjunctive Programming problem (Raman and Grossmann, 1994), which includes Boolean variables, disjunctions and logic propositions as shown in problem (P),

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

$f$: $R^n \to R^1$ is the term for continuous variables $x$ in the objective function and $r$: $R^n \to R^q$ are common constraints that hold regardless of the discrete decisions. $f(x)$, $r(x)$, and $g(x)$ are assumed to be nonconvex functions of the general form, $h(x) + \sum_i \sum_j a_{ij} x_i x_j + \sum_i \sum_j b_{ij} \frac{x_i}{x_j} + \sum_i l_i(x)$, where $h(x)$ is a convex function and $l_i(x)$ is a concave separable function, and the second and the third terms involve bilinear and linear fractional functions, respectively (see Quesada and Grossmann, 1995; Zamora and Grossmann, 1998b).

The disjunctions $k \in K$ are composed of a number of terms $j \in J_k$ that are connected by the OR operator ($\vee$). In each term, there is a Boolean variable $Y_{jk}$, a set of nonconvex inequalities $g_{jk}(x) \le 0$, $g^i_{jk}$: $R^n \to R^1$, and a cost variable $c_k$. If $Y_{jk}$ is true, then $g_{jk}(x) \le 0$ and $c_k = g_{jk}$ are enforced. Otherwise, these constraints are ignored. We assume here that each term in the disjunctions gives rise to a non-empty feasible region which is generally nonconvex. Also, $\Omega(Y) = True$ are logic propositions for the Boolean variables. Continuous variables $x$ are assumed to have lower and upper bounds.

The overall procedure of the proposed two-level branch and bound algorithm is as follows (see Figure 2). We first introduce convex underestimators in the nonconvex GDP problem (P), and construct the underestimating problem (R). This convex GDP problem is then reformulated as the convex NLP problem (CRP) by using the convex hull relaxation of each disjunction. Since all the disjunctions are relaxed, the convex NLP problem yields a valid lower bound. An initial upper bound is obtained by solving a nonconvex MINLP reformulation of the nonconvex GDP by a standard MINLP method such as DICOPT++ (Viswanathan and Grossmann, 1990). The upper bound is used for bound contraction to reduce the feasible region (Zamora and Grossmann, 1997). The discrete branch and bound method by Lee and Grossmann (2000) is applied at the first level of the branch and bound to solve the convex GDP problem. When all the Boolean variables are fixed, a spatial branch and bound method is used at the second level for solving the corresponding nonconvex NLP problem to yield an upper bound. The application of the proposed algorithm is illustrated with several example problems.

## Convex Relaxation of GDP

Problem (P) is first reformulated into a convex GDP problem by introducing valid convex underestimating functions as shown below,

$$
\begin{aligned}
&\min Z^R = \sum_{k\in K} c_k + \bar{f}(x) \\
&s.t. \qquad \bar{r}(x) \le 0 \\
&\bigvee_{j\in J_k} \begin{bmatrix} Y_{jk} \\ \bar{g}_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{bmatrix},\ k\in K \\
&\Omega(Y) = True \\
&0 \le x \le U,\ 0 \le c_k,\ Y_{jk} \in \{true, false\} \\
&x \in R^n,\ c \in R^K
\end{aligned}
\tag{R}
$$

The functions $\bar{f}, \bar{r},$ and $\bar{g}$ are valid convex underestimators such that $\bar{f}(x) \le f(x)$ and the inequalities $\bar{r}(x) \le 0$ and $\bar{g}(x) \le 0$ are satisfied if $r(x) \le 0$ and $g(x) \le 0$ (see Figure 1). Hence, the optimal solution $Z^{R*}$ of problem (R) provides a valid lower bound to the global optimal solution of problem (P). The specific underestimators for the bilinear, linear fractional, and concave separable terms are given in Appendix A. A recent review of these functions and some of its properties can be found in Tawarmalani and Sahinidis (2000a).

Since problem (R) is a convex GDP as described in Lee and Grossmann (2000), the feasible region of problem (R) can be relaxed by replacing each disjunction by its convex hull. This relaxation yields the following convex NLP model:

$$
\begin{aligned}
&\min Z^{L} = \sum_{k\in K}\sum_{j\in J_k} g_{jk}\, l_{jk} + \bar{f}(x) \\
&s.t. \qquad \bar{r}(x) \le 0 \\
&x = \sum_{j\in J_k} ?^{jk}, \quad \sum_{j\in J_k} l_{jk} = 1, \quad k\in K \qquad\qquad \text{(CRP)} \\
&0 \le ?^{jk} \le l_{jk} U_{jk}, \ j\in J_k, \quad k\in K \\
&l_{jk}\bar{g}_{jk}(?^{jk} / l_{jk}) \le 0, \ j\in J_k, \quad k\in K \\
&A\, l \le a \\
&0 \le x, ?^{jk} \le U, 0 \le l_{jk} \le 1, \ j\in J_k, k\in K
\end{aligned}
$$

where $\nu^{jk}$ is the disaggregated continuous variable for the $j$-th term in the $k$-th disjunction, and $\lambda_{jk}$ is the corresponding multiplier for each term $j \in J_k$ in a given disjunction $k \in K$. Note that problem (CRP) does not involve the Boolean variables $Y_{jk}$ since they are replaced by the continuous variables $\lambda_{jk}$, $0 \le \lambda_{jk} \le 1$. The constraints $l_{jk}\overline{g}_{jk}(v^{jk} / l_{jk})$ are convex if $g_{jk}(x)$ is convex (Hiriart-Urruty and Lemaréchal, 1993). Note that the logic propositions $\Omega(Y)$ = *True* are replaced by the linear constraints $A\lambda \le a$. Give that problem (R) yields a lower bound, and problem (CRP) is a relaxation of problem (R), the following proposition can be trivially established for problem (CRP):

**Proposition 1** *The optimal solution $Z^{L*}$ of problem (CRP) yields a lower bound to the optimal solution $Z^{P*}$ of problem (P).*

For implementation, the inequalities in the disjunctions are replaced by $(l_{jk} + e)\overline{g}_{jk}(v^{jk} / (l_{jk} + e)) \le 0$ where $e$ is a small tolerance (e.g. $\varepsilon = 0.0001$). These reformulated constraints are convex if $\overline{g}_{jk}(x) \le 0$ is convex. One can reformulate problem (CRP) as an MINLP by restricting the variables $\lambda_{jk}$ to binary values. For detailed properties of problem (CRP), see Lee and Grossmann (2001). In this paper, we use problem (CRP) within a two-level branch and bound method that will be explained in later sections.

As stated by the above proposition, a rigorous lower bound to problem (P) is obtained by solving problem (CRP) which has a unique local optimal solution (Bazaraa et al., 1993). This objective value is used as the initial value of the Global Lower Bound (GLB) when solving the nonconvex GDP problem (P). This lower bound is updated when fixing the Boolean variable $Y_{jk}$ in problem (R) (or corresponding $\lambda_{jk}$ in (CRP)) in the discrete branch and bound search. When all $\lambda_{jk}$ are either 0 or 1 (feasible to problem (R)), and there is no gap between the convex underestimators and the nonconvex functions at the solution point of problem (CRP), then the optimal objective value is a valid upper bound to problem (P) since this solution is also feasible to problem (P).

**Global Upper Bound Subproblem**

A valid upper bound for problem (P) can be obtained by applying an algorithm, such as the Augmented Penalty/Outer Approximation/Equality Relaxation implemented in DICOPT++ (Viswanathan and Grossmann, 1990), to the MINLP reformulation of (P). This yields the following nonconvex MINLP:

$$
\begin{aligned}
&\min Z = \sum_{k\in K}\sum_{j\in J_k} g_{jk} y_{jk} + f(x) \\
&s.t. \quad r(x) \le 0 \\
&g_{jk}(x) \le M_{jk}(1-y_{jk}),\ j\in J_k, k\in K \\
&\sum_{j\in J_k} y_{jk} = 1,\ k\in K \\
&Ay \le a \\
&0 \le x \le U,\ y_{jk}\in\{0,1\},\ j\in J_k, k\in K
\end{aligned}
\qquad \text{(P - MIP)}
$$

where $M_{jk}$, the "big-M" parameter, is a valid upper bound to the violation of the inequality $g_{jk}(x) \le 0$ and parameter $U$ is an upper bound to $x$.

In our experience, we have usually found a very good upper bound by using DICOPT++ to solve problem (P-MIP). Since this problem is nonconvex, the lower bound predicted by the MILP master problem is not valid, and therefore the heuristic termination criterion is used in DICOPT++ which stops when no further improvement is found in the NLP subproblem. The solution of (P-MIP) yields a Global Upper Bound (GUB) which is useful in pruning non-optimal nodes in the discrete branch and bound search.

**Bound Contraction Procedure**

Usually considerable computational work is required in both the discrete and the spatial branch and bound search for finding the subregion which contains the global optimal solution. The difference between the lower and the upper bounds largely depends on the variable bounds. Since elimination of non-optimal subregions is crucial in accelerating the search, we consider a bound contraction scheme to tighten the lower and upper bound of a given continuous variable $x_{i,}$ $i$ = 1,2,3,…,$n$ by solving the following NLP problem:

$$
\begin{aligned}
&\text{min/max } x_i \\
&s.t. \sum_{k\in K}\sum_{j\in J_k} g_{jk}\, l_{jk} + \bar{f}(x) \le GUB \\
&\bar{r}(x) \le 0 \\
&x = \sum_{j\in J_k} ?^{jk}, \quad \sum_{j\in J_k} l_{jk} = 1, \quad k \in K \\
&0 \le ?^{jk} \le l_{jk} U_{jk}, \; j \in J_k, \quad k \in K \\
&l_{jk}\bar{g}_{jk}(?^{jk} / l_{jk}) \le 0, \; j \in J_k, \quad k \in K \\
&A\, l \le a \\
&x^L \le x \le x^U \\
&0 \le v^{jk} \le U, 0 \le l_{jk} \le 1, \; j \in J_k, \quad k \in K
\end{aligned}
\qquad \text{(BCP)}
$$

If the solution point $x_i$ of problem (CRP) does not lie at its bound, then we solve the NLP problem (BCP) and update the upper and lower bounds (see Figure 3). Another way of updating the bounds is through range reduction, where cuts are generated based on the active constraints in the relaxed solution (Ryoo and Sahinidis, 1995; Tawarmalani and Sahinidis, 2000a). In this paper, we follow the bound contraction operation which was proposed by Zamora and Grossmann (1999). In the relaxed solution of problem (CRP), one continuous variable $x_i$, which is not at its bound, is selected. Then the direction of bound contraction (min or max) is decided based on the relative distance from the solution to each bound. Bound contraction is applied to only the continuous variables. The discrete variables are not fixed and they are relaxed as continuous variables in solving problem (BCP). The iteration of bound contraction continues until the solution of subproblem does not yield a reduction greater than a specified tolerance.

### Branch and Bound on Boolean Variables

In this step, a branch and bound method is applied in the space of the terms of the disjunctions by solving the relaxed convex NLP problem (CRP) at each node (for detailed description, see Lee and Grossmann, 2000). The branching rule is to select the variable $\lambda_{jk}$ that has the largest fractional value in the solution. Two child nodes are created by fixing $\lambda_{jk} = 1$ and $\lambda_{jk} = 0$, which means that we fix $Y_{jk}$ as true and as false in problem (R), respectively. For the case when we fix $Y_{jk}$ = true, we simply fix at that node the corresponding $j$-th term of disjunction $k$. When we fix $Y_{jk}$ = false, we consider at that node the convex hull relaxation of all terms $j' \neq j$. Since the number of Boolean variables in problem (R) is finite, the search in the discrete space requires a finite number of nodes in the branch and bound tree. The global lower bound, GLB, increases monotonically as the variables $\lambda_{jk}$ are fixed in the branch and bound tree. When all the $\lambda_{jk}$ are either 0 or 1, the solution for Boolean variables is feasible for the GDP problem (R). If there is a gap between the solution of this problem and the original nonconvex GDP problem (P), we need to update the upper bound of the objective value so that the convex approximation is small enough within a given tolerance.

At a node where a feasible solution to problem (R) is obtained, and the gap between every nonconvex term in problem (P) and its convex underestimator in problem (R) is nonzero, we fix all Boolean variables and switch to a spatial branch and bound method. At this node we solve a nonconvex NLP problem to global optimality, and obtain a feasible solution to problem (P) (if one exists). If the solution is lower than the global upper bound, GUB, it is updated. After the spatial branch and bound is completed, we return to the current node of the discrete branch and bound tree and add a cut for the Boolean variables to the discrete branch and bound to exclude the previous choice of the fixed Boolean variables. By solving problem (R) at the current node with this cut, a new solution of $Y_{jk}$ is generated. If the solution is infeasible, then we close the current node and backtrack. If the solution is feasible and there is a gap between GUB and GLB, which is the lowest objective value among the open nodes, then we keep branching. The search stops when there are no open nodes with an objective value less than GUB in the discrete branch and bound tree.

**Spatial Branch and Bound (SBB) Method**

When all the Boolean variables are fixed, problem (CRP) reduces to a discrete feasible GDP problem (R) since each disjunction is satisfied. If there is a gap between the convex underestimators and the nonconvex functions, we need to branch on the continuous variables *x* to reduce the feasible region by contracting the upper and lower bounds of *x*.

The spatial branch and bound described is essentially the one by Quesada and Grossmann (1995), and has three main steps: the branching variable selection, the branching point selection and the node selection. After solving problem (CRP), the nonconvex term with the maximum gap between its convex underestimator and the nonconvex term is chosen. If that function has more than one variable as its arguments, then the variable with the largest difference in the variable bounds, $|x^U - x^L|$, is selected since we expect that the variable with largest variable bound would have the most effect on the reduction of the gap. The branching point used in this work is the middle point of the variable bound, $(x^U + x^L)/2$ (bisection rule). Alternatively, one can use the solution point *x* of problem (CRP) at the current node (omega rule). Among the open nodes, we select the node with lowest objective value. If the objective value of the current node is greater than GUB, then the current node is fathomed. Since branching is applied to continuous variables, the spatial branch and bound search is theoretically an infinite process although it has finite termination for ε-convergence. The performance, however, heavily depends on the actual value selected for ε.

An optional step in the spatial branch and bound is to use the bound contraction operation whenever a new GUB is found. It can help to reduce the variable bounds, but additional effort is spent in solving the contraction subproblem. This option was not used in this paper.

### Solution Algorithm for Nonconvex GDP Problems

In this section, we describe the global optimization algorithm for nonconvex GDP problems (see Figure 2). In this algorithm, we define the gap as the difference between a given nonconvex function and its convex underestimator at the solution point *x** of problem (CRP) (i.e. gap = $f(x^*) - \overline{f}(x^*)$).

Step 0. Heuristic Search (Nonconvex MINLP)

Solve the nonconvex problem (P-MIP) with an MINLP solver such as DICOPT++ (Viswanathan and Grossmann, 1990).

Set GUB as the best upper bound obtained by DICOPT++; let ($Y^*$, $x^*$) be the solution.

Step 1. Bound Contraction (Convex NLP)

1.1 Set the maximum number of major iterations, NC. Set BC = 0 and the tolerance $\delta$.

Set the minimum value $SP_m$ for a successful contraction step. (e.g., 0.2)

Initialize the Relative Distance (RD) from $x_i$* to each bound:

$$RDL_i = \frac{x_i * - x_i^L}{x_i^U - x_i^L}, RDU_i = \frac{x_i^U - x_i *}{x_i^U - x_i^L} \quad i = 1,2,3,...n$$

1.2 Increase the iteration count, BC = BC + 1.

For $i = 1,2,3,\ldots,n$

Solve problem (BCP).

- Minimize $x_i$ if $RDL_i > RDU_i$ and $RDL_i > \delta$.

- Maximize $x_i$ if $RDL_i \leq RDU_i$ and $RDU_i > \delta$.

If contraction is greater than $SP_m$, update the bound and continue with $x_i$.

Otherwise, select next variable $x_i$.

1.3 Return to step 1.2 and repeat while the iteration count BC ≤ NC.

Step 2. Branch and Bound on Discrete Variables (Convex NLP)

2.1 Set tolerance $\alpha$ for difference in $Z^L$ and GUB. Set $\varepsilon$ for maximum gap.

2.2 Solve problem (CRP) to obtain lower bound $Z^L$. Update lowest lower bound as GLB.

a) If the solution is feasible and $Z^L \geq$ GUB - $\alpha$, then fathom the node.

b) If the solution is infeasible, then fathom the node.

c) If the solution is feasible and all $\lambda_{jk}$ are either 0 or 1, then

If for all functions gap ≤ $\varepsilon$ and $Z^L <$ GUB, then GUB = $Z^L$. Backtrack.

Else if any gap > $\varepsilon$, then go to Step 3 with fixed $\lambda_{jk}$.

After GUB is updated from Step 3, add a cut for the current Boolean variables, $\neg\left[\underset{i\in T_n}{\wedge} Y_i \underset{i\in F_n}{\wedge} \left(\neg Y_i\right)\right]$ where $T_n$ and $F_n$ are the set of Boolean variables that are true and false at the current node.

d) If the solution is feasible and $Z^L <$ GUB - $\alpha$, then

Branch on $\lambda_{jk}$ which is closest to one.

Create two child nodes ($\lambda_{jk} = 1$ and $\lambda_{jk} = 0$).

2.3 Return to step 2.2 according to a specified search strategy (e.g. depth first or breadth first) while GLB < GUB - $\alpha$.

Step 3. Spatial Branch and Bound (Convex NLP)

3.1 Fix all $\lambda_{jk}$ as either 0 or 1 according to the solution from Step 2.

3.2 Solve problem (CRP).

a) If the solution is feasible and $Z^L \geq$ GUB - $\alpha$, then fathom the node.

b) If the solution is infeasible, then fathom the node.

c) If the solution is feasible and for all functions gap $\leq \varepsilon$, update GUB when $Z^L <$ GUB.

d) If the solution is feasible and any gap $> \varepsilon$, then

Select the nonconvex function $f_r$ with maximum gap.

Select the continuous variable $x_s$ among the arguments of $f_r$ with maximum variable bound difference, $|x^U - x^L|$.

Set the branching point $x_s^B$ according to the branching rule.

$$\text{Omega Rule}: x_s^B = x_s\,*$$
$$\text{Bisection Rule}: x_s^B = (x_s^L + x_s^U)/2$$

Create two child nodes ($x_s \leq x_s^B$ and $x_s^B \leq x_s$).

3.3 Select a node and return to step 3.2 and continue until there is no open node with $Z^L <$ GUB - $\alpha$.

3.4 Go to Step 2.

It should be pointed out that the above algorithm finds the optimal solution in a finite number of steps. This follows from the finiteness of the discrete branch and bound tree (step 2), and the finite termination for ε-convergence in the global optimization subproblem (step 3). It should be noticed that step 3 is included in the branch and bound tree of step 2, giving rise to an embedded tree structure. The proposed algorithm is in that sense similar to the work by Adjiman et al. (2000) who used as an option a two-level sequential branching strategy for nonconvex MINLP problems in which binary variables are selected for branching in the first level and then continuous variables are selected in the second level of the SMIN-αBB algorithm. Finally, it is worth noting that fixing all the discrete variables in step 3 may in special cases simplify the problem to a convex NLP problem which can then simply be solved with a local optimizer (see examples 5 and 6).

## Illustrative Example

We describe the solution procedure with a small nonconvex GDP example, which was originally proposed as a nonconvex MINLP by Kocis and Grossmann (1989) for optimizing a small superstructure consisting of two reactors. This problem can be reformulated as the following nonconvex GDP problem:

$$
\begin{gathered}
\min Z = c + 5x + p \\
s.t. \\
\begin{bmatrix} Y_1 \\ 10 = 0.9[1 - \exp(-0.5v)]x \\ p = 7.0v \\ c = 7.5 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ 10 = 0.8[1 - \exp(-0.4v)]x \\ p = 6.0v \\ c = 5.5 \end{bmatrix} \\
0 \le v \le 10;\ 0 \le x \le 20;\ 0 \le c, p
\end{gathered}
\tag{E0}
$$

The optimal solution is 99.2 with $Y^*$ = (true, false), $x^*$ = 13.4 and $v^*$ = 3.5. Note that problem (E0) has only one disjunction which has nonlinear equality constraints which are nonconvex. In step 0 of the proposed algorithm, a nonconvex MINLP reformulation (P-MIP) is solved with the OA method. An initial upper bound of 99.2 is obtained after three major iterations. The GUB is initialized as 99.2 and is used in the remaining steps. To derive the convex relaxation we first substitute [1-exp(-0.5$v$)] in the first term with the continuous variable $\alpha$, resulting in bilinear terms. The nonlinear equality $\alpha$ = [1-exp(-0.5$v$)] is replaced by two nonlinear inequalities. The second term in the disjunction follows the same substitution, which leads to:

$$
\begin{gathered}
\min Z = c + 5x + p \\
s.t. \\
\begin{bmatrix} Y_1 \\ 10 = 0.9ax \\ a \le [1 - \exp(-0.5v)] \\ a \ge [1 - \exp(-0.5v)] \\ p = 7.0v \\ c = 7.5 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ 10 = 0.8ax \\ a \le [1 - \exp(-0.4v)] \\ a \ge [1 - \exp(-0.4v)] \\ p = 6.0v \\ c = 5.5 \end{bmatrix} \\
0 \le v \le 10;\ 0 \le x \le 20;\ 0 \le c, p, a
\end{gathered}
\tag{E0′}
$$

The bilinear term $\alpha x$ is replaced by linear under and overestimators (see Appendix A). In the first term of the disjunction, the inequality $\alpha \le$ [1-exp(-0.5$v$)] is convex while the inequality $\alpha \ge$

[1-exp(-0.5*v*)] is concave. We underestimate the concave term by a secant line which matches the concave term at the lower and upper bound of *v*. The convex underestimating problem of (E0) is then as follows:

$$\min Z = c + 5x + p$$

$$s.t.$$

$$\left[\begin{array}{c} Y_1 \\ 10/0.9 \geq a_1^U x + a x^U - a_1^U x^U \\ 10/0.9 \geq a_1^L x + a x^L - a_1^L x^L \\ 10/0.9 \leq a_1^U x + a x^L - a_1^U x^L \\ 10/0.9 \leq a_1^L x + a x^U - a_1^L x^U \\ a \leq [1 - \exp(-0.5v)] \\ a \geq a_1^L + (v - v^L)\dfrac{a_1^U - a_1^L}{v^U - v^L} \\ p = 7.0v \\ c = 7.5 \end{array}\right] \vee \left[\begin{array}{c} Y_2 \\ 10/0.8 \geq a_2^U x + a x^U - a_2^U x^U \\ 10/0.8 \geq a_2^L x + a x^L - a_2^L x^L \\ 10/0.8 \leq a_2^U x + a x^L - a_2^U x^L \\ 10/0.8 \leq a_2^L x + a x^U - a_2^L x^U \\ a \leq [1 - \exp(-0.4v)] \\ a \geq a_2^L + (v - v^L)\dfrac{a_2^U - a_2^L}{v^U - v^L} \\ p = 6.0v \\ c = 5.5 \end{array}\right] \qquad \text{(R0)}$$

$$0 \leq v \leq 10;\ 0 \leq x \leq 20;\ 0 \leq c, p, a$$

where $a_1^L = 1 - \exp(-0.5v^L), a_1^U = 1 - \exp(-0.5v^U), a_2^L = 1 - \exp(-0.4v^L), a_2^U = 1 - \exp(-0.4v^U)$ .

This relaxation problem (R0) is a convex nonlinear disjunctive problem for which we apply the convex hull relaxation to the disjunction, resulting in problem (CRP). In step 1, we solve problem (BCP) of (R0) for bound contraction of the continuous variable *x* and *v*. The discrete variables are relaxed in solving the bound contraction problem. Initially, the bounds are 0 ≤ *x* ≤ 20 and 0 ≤ *v* ≤ 10. After solving four NLPs, the new bounds are 11.1 ≤ *x* ≤ 18.7 and 1.6 ≤ *v* ≤ 5.1. The interval size of the bounds of *x* is reduced to 38 % of its original size, and for *v* it is reduced to 35 %. The percentage bound reduction of each variable is calculated from the ratio of the new bounds gap to the old bounds gap, $| x_{new}^U - x_{new}^L | / | x_{old}^U - x_{old}^L |$. With the new bounds, the discrete branch and bound is performed in step 2. Figure 4 shows the branch and bound tree and Table 1 shows the numerical results of each step of the proposed method. The first lower bound at the root node (GLB) is 97.5 and the gap between GUB and GLB is 1.7 %. At the root node, we relax $l$ in problem (CRP) as continuous variables between 0 and 1. Here $l_{jk}$ corresponds to the Boolean variable $Y_{jk}$ in problem (R) and $l_{jk}$ = 1 in the solution means $Y_{jk}$ = true. Solving problem (CRP) yields a discrete feasible solution of $l$ = (1,0). For convenience, we denote this value of $l$ as $Y^L$. This integer value is fixed as $Y^F$ = (1,0) and the bound GUB (99.2) is given to the spatial branch and bound step (node S1 in Figure 4). In step 3, the branching variables are *x* and *v*. The variable with the largest difference in the variable bounds is selected first. The branching variable and its branching point are shown on each node. At node S1, *x* is selected first and the branching point is the middle point of the variable bounds. At node S3, the objective value 99.3 is higher than GUB, so it is fathomed. Node S4 is infeasible and node S5 yields the optimal solution. Five NLPs are solved with a relative tolerance of 0.1 % and the optimality of the upper bound 99.2 is again verified for fixed $Y^F$ = (1,0). A cut is added to node 1 of step 2 and problem (CRP) is resolved since the gap between GUB and GLB is not closed yet at the node 1. Node 2 yields a solution $Y^L$ = (0.5,0.5) and $Z^L$ = 101.6, and hence it is fathomed by GUB and the search is stopped. Therefore, a total of seven NLPs are solved in the branch and bound procedure. The problem size is too small for the computing time to be of any significance.

**Numerical Examples**

In this section, we apply the proposed algorithm to seven nonconvex GDP example problems. All of them were solved with GAMS (Brooke et al., 1997) on a 300 MHz Pentium II PC with 128MB of memory. The GAMS/DICOPT++ MINLP solver (AP/OA/ER) and GAMS/CONOPT NLP solver are used. The tolerances selected were $\delta$ = 0.01, $\alpha$ = 0.01 % (of GUB), $SP_m$ = 0.2, and $\varepsilon$ = 0.0001. Table 2 shows the size and numerical results of the nonconvex GDP problems.

**Example 1**

This example is adapted from Lee and Grossmann (2001). The model involves a disjunction of convex feasible sets and a concave objective function.

$$\begin{gathered}\min Z = -(x_1-3)^2-(x_2-2)^2+c \\ s.t. \\ \begin{bmatrix} Y_1 \\ (x_1)^2+(x_2)^2-1\le 0 \\ c=2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1-4)^2+(x_2-1)^2-1\le 0 \\ c=1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ (x_1-2)^2+(x_2-4)^2-1\le 0 \\ c=3 \end{bmatrix} \quad \text{(E1)} \\ 0\le x_1, x_2\le 8,\ 0\le c,\ Y_j\in\{true,\ false\},\ j=1,2,3.\end{gathered}$$

The optimal solution is –11.0 with $x^*$ = (0,0) and $Y^*$ = (true, false, false). Underestimators for the concave terms $-x_1^2$ and $-x_2^2$ are secant lines that match the concave terms exactly at the lower and upper bounds of $x$. Problem (E1) is then replaced by the convex GDP:

$$\min Z = a_1 + a_2 + 6x_1 + 4x_2 - 13 + c$$

$$s.t. \quad a_1 \geq -\left(x_1^L\right)^2 + (x_1 - x_1^L)\frac{-\left(x_1^U\right)^2 + \left(x_1^L\right)^2}{(x_1^U - x_1^L)}$$

$$a_2 \geq -\left(x_2^L\right)^2 + (x_2 - x_2^L)\frac{-\left(x_2^U\right)^2 + \left(x_2^L\right)^2}{(x_2^U - x_2^L)}$$

$$\begin{bmatrix} Y_1 \\ (x_1)^2 + (x_2)^2 - 1 \leq 0 \\ c = 2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c = 1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \leq 0 \\ c = 3 \end{bmatrix} \quad \text{(R1)}$$

$$0 \leq x_1, x_2 \leq 8, 0 \leq c, \; Y_j \in \{true, \; false\}, \; j = 1,2,3.$$

where $x_n^L$ and $x_n^U$ are lower and upper bounds of $x_n$. In step 0 of the proposed method, DICOPT++ finds the global optimal solution in four major iterations by solving the nonconvex MINLP reformulation of (E1). GUB = -11.0 is used in the bound contraction problem (BCP) in step 1. When we solve problem (BCP), no $\lambda_j$ is fixed and the bound contraction is applied to $x_1$ and $x_2$ only. In step 1, nine iterations of bound contraction yield 69.5 % of reduction in the feasible region. Due to the reduced feasible region of $x$, the first lower bound by solving problem (CRP) in step 2 is –11.25, which has 2.3% optimality gap from GUB. For comparison, if we use the original bound $0 \leq x \leq 8$ in step 2, then the first lower bound from problem (CRP) is –34.4. Therefore, there is a significant improvement of the lower bound by contracting the feasible region. The discrete branch and bound step searched 3 nodes and one feasible choice of $Y$ = (true, false, false) was found during the search. At that discrete choice, the spatial branch and bound step required three nodes, again verifying the best upper bound of –11.0. A total six NLP subproblems were solved in the branch and bound steps. For detailed results in the solution procedure, see Table 2.

### Example 2

The second example has a nonconvex objective function with the same disjunctive feasible set as in example 1.

$$\min Z = x_1^4 - 14x_1^2 + 24x_1 - x_2^2 + c$$

$$s.t.$$

$$\begin{bmatrix} Y_1 \\ (x_1)^2 + (x_2)^2 - 1 \le 0 \\ c = 2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \le 0 \\ c = 1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \le 0 \\ c = 3 \end{bmatrix} \quad \text{(E2)}$$

$$0 \le x_1, x_2 \le 8,\ 0 \le c,\ Y_j \in \{true,\ false\},\ j = 1,2,3.$$

The optimal solution is –14.0 with $x^*$ = (2,5) and $Y^*$ = (false, false, true). Secant lines for the concave functions $-x_1^2$ and $-x_2^2$ are again used as convex underestimators. We replace the nonconvex terms in the objective function by the continuous variables, $\alpha_1$ and $\alpha_2$. The relaxed convex GDP problem is then as follows:

$$\min Z = x_1^4 + 14a_1 + 24x_1 + a_2 + c$$

$$s.t. \quad a_1 \ge -\left(x_1^L\right)^2 + (x_1 - x_1^L)\frac{-\left(x_1^U\right)^2 + \left(x_1^L\right)^2}{(x_1^U - x_1^L)}$$

$$a_2 \ge -\left(x_2^L\right)^2 + (x_2 - x_2^L)\frac{-\left(x_2^U\right)^2 + \left(x_2^L\right)^2}{(x_2^U - x_2^L)}$$

$$\begin{bmatrix} Y_1 \\ (x_1)^2 + (x_2)^2 - 1 \le 0 \\ c = 2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \le 0 \\ c = 1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \le 0 \\ c = 3 \end{bmatrix} \quad \text{(R2)}$$

$$0 \le x_1, x_2 \le 8,\ 0 \le c,\ Y_j \in \{true,\ false\},\ j = 1,2,3.$$

In step 0, DICOPT++ finds a feasible solution –14.0 in four major iterations. With this objective value as an initial GUB, the bound contraction operation is applied. In step 1, eight iterations of solving problem (BCP) result in an average reduction of 52.6 % in the bounds. By solving the convex hull relaxation problem (CRP) of (R2), the first lower bound is –37.36 (see Figure 5). At the root node, the relaxed value $\lambda_3$ is closest to one and it is selected as branching variable. By fixing $Y_3$ as true, we find a discrete feasible solution of $Y^L$ = (0,0,1). By fixing these Boolean variables according to $Y^F$ = (0,0,1), the spatial branch and bound method finds after 27 nodes an upper bound of –14.0, which is the optimal solution to problem (E2). When returning to the discrete branch and bound, the GUB is updated and the cut which excludes the choice $Y$ = (false, false, true) is added. The second node is resolved with this cut and the solution is infeasible. Therefore, we go to the third node by fixing $Y_3$ as false. The last two nodes are fathomed since their objective values are greater than GUB. A total of six nodes are searched in the discrete search of step 2.

**Example 3**

The next example is taken from Kocis and Grossmann (1989). It involves a nonconvex MINLP problem which consists in selecting the optimal structure for separating a multicomponent process streams into a set of product streams with given purity specifications (see Figure 6). We formulate this problem with the following nonconvex GDP model:

$$
\begin{aligned}
\min Z = & -35P1A - 30P2B + 10F1 + 8F2 + F4A + F4B \\
& + 4F5A + 4F5B + 2CF + 50CD \\
s.t. \quad & F3A = 0.55F1 + 0.50F2 \\
& F3B = 0.45F1 + 0.50F2 \\
& P1A = F8A + F10A + F6A \\
& P1B = F8B + F10B + F6B \\
& P2A = F9A + F11A + F7A \\
& P2B = F9B + F11B + F7B \\
& F6A = E6F3A, F6B = E6F3B \\
& F7A = E7F3A, F7B = E7F3B \\
& E4 + E5 + E6 + E7 = 1 \\
& P1A \geq 4.0P1B,\ \ P2B \geq 3.0P2A \\
& P1A + P1B \leq 15, P2A + P2B \leq 18
\end{aligned}
$$

$$
\begin{bmatrix} YF \\ F4A = E4F3A, F4B = E4F3B \\ 2.5 \leq F4A + F4B \leq 25 \\ F8A = 0.85F4A, F8B = 0.20F4B \\ F9A = 0.15F4A, F9B = 0.80F4B \\ CF = 2 \end{bmatrix} \vee \begin{bmatrix} \neg YF \\ F4A = F4B = 0 \\ F8A = F8B = 0 \\ F9A = F9B = 0 \\ E4 = 0 \\ CF = 0 \end{bmatrix} \tag{E3}
$$

$$
\begin{bmatrix} YD \\ F5A = E5F3A, F5B = E5F3B \\ 2.5 \leq F5A + F5B \leq 25 \\ F10A = 0.975F5A, F10B = 0.050F5B \\ F11A = 0.025F5A, F11B = 0.950F5B \\ CD = 25 \end{bmatrix} \vee \begin{bmatrix} \neg YD \\ F5A = F5B = 0 \\ F10A = F10B = 0 \\ F11A = F11B = 0 \\ E5 = 0 \\ CD = 0 \end{bmatrix}
$$

$$
0 \leq CF, CD; F1, F2 \leq 25;\ 0 \leq E4, E5, E6, E7 \leq 1; Y \in \{true,\ false\}
$$

The optimal solution is –510.08 with *F*1* = 8, *F*2* = 25, *P*1* = *P*2* = 15, *E** = (0.108,0.758,0,0,134), and *Y** = (true, true). In this example, the bilinear terms are replaced by continuous variables and we add the linear underestimators and overestimators by McCormick (1976) as discussed by Quesada and Grossmann (1995) to construct the relaxed convex GDP problem (R). Using the heuristic search in step 0, DICOPT++ finds the trivial solution of 0.0 at the first NLP. The bound contraction step for continuous variables yields 10.8 % of reduction after solving fourteen LP subproblems in step 1. Performing the discrete branch and bound on problem (CRP) in step 2, a feasible solution to the relaxed GDP is found at the third node with $Y^L$ = (1,1) and objective value of –661.56. At this node, we switch to spatial branch and bound (step 3) with fixed $Y^F$ = (1,1). This means that we fix *Y* as (true, true) and selecting the corresponding terms in the disjunctions in (E3). Branching on the continuous variables, it takes 27 nodes to reduce the gap (27 LPs). An upper bound solution (-510.08) is found and this solution is used to update in step 2 the value of GUB. A cut to exclude *Y* = (true, true) is added to the previous node in step 2 and we resolve problem (CRP). Finding that the solution is infeasible, we close current node and backtrack. In step 2, two more nodes are searched and they are fathomed by the GUB. The optimal structure is shown in Figure 7.

**Example 4**

The fourth example is the nonconvex MINLP example 4.6 by Zamora (1997). The reformulated nonconvex GDP model is as follows:

$$\text{Min } Z = \sum_{k=1}^{3} c_k - 71x_1 - 60x_2 + 65x_3 + 57x_4 - 30x_5 - 65x_1x_4 - 56x_1x_3 - 85x_2x_3 - 87x_2x_5$$

$$\begin{aligned} s.t. \quad & 2.5x_1 - 1.8x_2 + 5x_4 - 5.6x_5 \le 296 \\ & x_2 + 4.6x_4 - 5x_5 + 1.5x_3x_4 + 2x_1x_2 \le 250 \\ & -3.5x_1 + 2.3x_2 + 4x_3 - 10x_5 - 6x_2x_4 + 2.2x_3x_5 \le 192.5 \\ & 1.9x_2 - 5x_3 - 1.4x_5 + 2.9x_2x_3 - 1.5x_3x_5 \le 134.5 \\ & 7.5x_1 + 5.8x_3 - 3x_5 + 1.5x_1x_2 - 3x_4x_5 = 55 \\ & -3.5x_2 - 10x_3 + 10.5x_5 - 3.5x_1x_2 \le 32 \end{aligned}$$

$$\begin{bmatrix} Y_1 \\ 2x_1 - x_2 + x_1x_3 \le 80 \\ c_1 = 54 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ 2x_1 - x_2 + x_1x_3 \le 0 \\ c_1 = 0 \end{bmatrix} \tag{E4}$$

$$\begin{bmatrix} Y_2 \\ 3.7x_2 - 7x_3 + 3.8x_4 - 5x_5 \\ +3.5x_1x_2 - 6x_2x_4 \le 75 \\ x_4 + 1.5x_5 \le 90 \\ c_2 = 58 \end{bmatrix} \vee \begin{bmatrix} \neg Y_2 \\ 3.7x_2 - 7x_3 + 3.5x_1x_2 \le 0 \\ x_4 = x_5 = 0 \\ c_2 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_3 \\ -7x_2 + 2x_3 - 2x_4 + 4x_5 \le 20 \\ x_1 + x_3 - x_5 \le 61.5 \\ c_3 = 30 \end{bmatrix} \vee \begin{bmatrix} \neg Y_3 \\ -7x_2 + 2x_3 - 2x_4 + 4x_5 \le 0 \\ x_1 + x_3 - x_5 \le 0 \\ c_3 = 0 \end{bmatrix}$$

$$0 \le x \le (53,85,40,64,68),\ 0 \le c,\ Y_k \in \{true,\ false\},\ k = 1,2,3$$

The optimal solution is –116,575 with $x^*$ = (2.952,52.77,0,1.736,24.46) and $Y^*$ = (false, true, true). Convex underestimators and overestimators for the bilinear terms are introduced and the bilinear terms are replaced by continuous variables, $\alpha_{ij}$. The convexification relaxes problem (E4) as the linear GDP problem (R4):

$$\min Z = \sum_{k=1}^{3} c_k - 71x_1 - 60x_2 + 65x_3 + 57x_4 - 30x_5 - 65a_{14} - 56a_{13} - 85a_{23} - 87a_{25}$$

$$\begin{aligned} s.t.\qquad & 2.5x_1 - 1.8x_2 + 5x_4 - 5.6x_5 \le 296 \\ & x_2 + 4.6x_4 - 5x_5 + 1.5a_{34} + 2a_{12} \le 250 \\ & -3.5x_1 + 2.3x_2 + 4x_3 - 10x_5 - 6a_{24} + 2.2a_{35} \le 192.5 \\ & 1.9x_2 - 5x_3 + 1.4x_5 + 2.9a_{23} - 1.5a_{35} \le 134.5 \\ & 7.5x_1 + 5.8x_3 - 3x_5 + 1.5a_{12} - 3a_{45} = 55 \\ & -3.5x_2 - 10x_3 + 10.5x_5 - 3.5a_{12} \le 32 \\ & a_{ij} \ge x_i^L x_j + x_j^L x_i - x_i^L x_j^L,\ a_{ij} \ge x_i^U x_j + x_j^U x_i - x_i^U x \\ & a_{ij} \le x_i^L x_j + x_j^U x_i - x_i^L x_j^U,\ a_{ij} \le x_i^U x_j + x_j^L x_i - x_i^U x_j^L \end{aligned}$$

$$\begin{bmatrix} Y_1 \\ 2x_1 - x_2 + a_{13} \le 80 \\ c_1 = 54 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ 2x_1 - x_2 + a_{13} \le 0 \\ c_1 = 0 \end{bmatrix} \tag{R4}$$

$$\begin{bmatrix} Y_2 \\ 3.7x_2 - 7x_3 + 3.8x_4 - 5x_5 \\ +3.5a_{12} - 6a_{24} \le 75 \\ x_4 + 1.5x_5 \le 90 \\ c_2 = 58 \end{bmatrix} \vee \begin{bmatrix} \neg Y_2 \\ 3.7x_2 - 7x_3 + 3.5a_{12} \le 0 \\ x_4 = x_5 = a_{14} = a_{24} \\ = a_{34} = a_{25} = a_{35} = a_{45} = 0 \\ c_2 = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_3 \\ -7x_2 + 2x_3 - 2x_4 + 4x_5 \le 20 \\ x_1 + x_3 - x_5 \le 61.5 \\ c_3 = 30 \end{bmatrix} \vee \begin{bmatrix} \neg Y_3 \\ -7x_2 + 2x_3 - 2x_4 + 4x_5 \le 0 \\ x_1 + x_3 - x_5 \le 0 \\ c_3 = 0 \end{bmatrix}$$

$$0 \le x \le (53,85,40,64,68),\ 0 \le c,\ Y_k \in \{true,\ false\},\ k = 1,2,3$$

$$0 \le a_{ij},\ i, j = 1,2,...,5.$$

Using DICOPT++ to solve the MINLP problem (P-MIP) in step 0 yields the optimal solution of –116,575 (GUB) after one major iteration. In the bound contraction step, 45 LPs are solved and the reduction is 22.2 %. Figure 8 shows the branch and bound tree for step 2 and step 3. In step 2, we solve the convex hull relaxation of problem (R4) and the first lower bound (GLB) is –242,474 at the root node with $Y^L$ = (0,1,0). Now the algorithm switches to the spatial branch and bound with $Y^F$ = (0,1,0) (Node S1 in Figure 8) and it takes 33 nodes to verify that the optimal solution with $Y^F$ = (0,1,0) is –116,575 which then becomes the current upper bound GUB. When returning to step 2, we add the cut for $Y$ = (false, true, false) to make this choice infeasible. Note that we need to branch further since the gap between the global upper bound (GUB) and global lower bound (GLB) is not closed. While branching on $\lambda_j$, three more integer solutions are found. Switching to a spatial branch and bound search, we found that the nodes are fathomed by GUB. The optimal solutions of the spatial branch and bound steps S2, S3 and S4 (see Figure 8) are inferior to the GUB. The total number of search nodes in each spatial branch and bound step is also shown in Figure 8. Step 2 requires 11 nodes to prove optimality. In total, 137 LPs are solved in branch and bound step. For comparison, Zamora (1996) required the solution of 126 LPs and 2 NLPs in his branch and bound method.

**Example 5.**

This example is a nonconvex MINLP problem for the synthesis and design of a batch plant with multiple parallel units with ZW policy scheduling using single product campaigns (Birewar and Grossmann, 1990). Figure 9 shows one of the major decisions in this problem which is the assignments of tasks to units which impacts the schedule and equipment sizes. We introduce disjunctions for assignments of tasks to units, existence of units and the number of parallel equipments. Logic propositions are introduced for the flowshop network structure (see Appendix B). The nonconvex GDP model is as follows:

$$\min \mathrm{COST} = \sum_{j=1}^{M} N_j^{EQ} C_j$$

$$s.t. \qquad V_t^T \geq B_i S_{it} \quad i = 1,\dots,N_P\,; t = 1,\dots,T$$

$$pt_{ij} = \sum_{t \in T_j} pty_{itj} \quad i = 1,\dots,N_P\,;\, j = 1,\dots,M$$

$$n_i B_i \geq Q_i \quad i = 1,\dots,N_P$$

$$\sum_{i=1}^{N_p} n_i T_{Li} \leq H$$

$$\begin{bmatrix} Y_{11} \\ V_1 \geq V_1^T \\ pty_{i11} = pt_{i1}^T \\ pty_{i12} = pty_{i14} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{12} \\ V_2 \geq V_1^T \\ pty_{i12} = pt_{i1}^T \\ pty_{i11} = pty_{i14} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{14} \\ V_4 \geq V_1^T \\ pty_{i14} = pt_{i1}^T \\ pty_{i11} = pty_{i12} = 0 \end{bmatrix} \qquad \text{(E5)}$$

$$\begin{bmatrix} Y_{22} \\ V_2 \geq V_2^T \\ pty_{i22} = pt_{i2}^T, pty_{i24} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{24} \\ V_4 \geq V_2^T \\ pty_{i24} = pt_{i2}^T, pty_{i22} = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_{33} \\ V_3 \geq V_3^T \\ pty_{i33} = pt_{i3}^T, pty_{i34} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{34} \\ V_4 \geq V_3^T \\ pty_{i34} = pt_{i3}^T, pty_{i33} = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_{45} \\ V_5 \geq V_4^T \\ pty_{i45} = pt_{i4}^T \end{bmatrix}$$

$$\left[\begin{array}{c} YEX_j \\ C_j = g_j + a_j V_j^{0.6} \\ V_j^L \le V_j \le V_j^U \\ \begin{bmatrix} YC_{1j} \\ N_j^{EQ} = 1 \\ T_{Li} \ge pt_{ij} \end{bmatrix} \vee \begin{bmatrix} YC_{2j} \\ N_j^{EQ} = 2 \\ 2T_{Li} \ge pt_{ij} \end{bmatrix} \vee \begin{bmatrix} YC_{3j} \\ N_j^{EQ} = 3 \\ 3T_{Li} \ge pt_{ij} \end{bmatrix} \vee \begin{bmatrix} YC_{4j} \\ N_j^{EQ} = 4 \\ 4T_{Li} \ge pt_{ij} \end{bmatrix} \end{array}\right] \vee \begin{bmatrix} \neg YEX_j \\ C_j = 0 \\ V_j = 0 \\ N_j^{EQ} = 0 \\ pt_{ij} = 0 \\ T_{Li} \ge 0 \end{bmatrix} \quad j = 1,\dots,M$$

Logic Propositions

$$YEX_1 \Leftrightarrow Y_{11}, YEX_2 \Leftrightarrow Y_{12} \vee Y_{22}, YEX_3 \Leftrightarrow Y_{33}$$

$$YEX_4 \Leftrightarrow Y_{14} \vee Y_{24} \vee Y_{34}, YEX_5 \Leftrightarrow Y_{45}$$

$$W_{04} \underline{\vee} W_{14} \underline{\vee} W_{24} \underline{\vee} W_{34}$$

$$W_{04} \Leftrightarrow \neg Y_{14} \wedge \neg Y_{24} \wedge \neg Y_{34}$$

$$W_{14} \Leftrightarrow (Y_{14} \wedge \neg Y_{24} \wedge \neg Y_{34}) \underline{\vee} (\neg Y_{14} \wedge Y_{24} \wedge \neg Y_{34}) \underline{\vee} (\neg Y_{14} \wedge \neg Y_{24} \wedge Y_{34})$$

$$W_{24} \Leftrightarrow (Y_{14} \wedge Y_{24} \wedge \neg Y_{34}) \underline{\vee} (\neg Y_{14} \wedge Y_{24} \wedge Y_{34})$$

$$W_{34} \Leftrightarrow Y_{14} \wedge Y_{24} \wedge Y_{34}$$

$$0 \le C_j, V_j, V_t^T, n_i, B_i, T_{Li}, pt_{ij}, N_j^{EQ}, pty_{itj}; YEX_{tj}, Y_{tj}, YC_{cj}, W_{lj} \in \{true,\ false\}$$

$$i = 1,\dots,N_P; t = 1,\dots,T; j = 1,\dots,M; l = 0,1,2,3$$

$$V_j^L = 250, V_j^U = 5000, V_5^U = 15000$$

where $YEX_j$ = true means that we use equipment $j$. $Y_{tj}$ is the Boolean variable for task assignments: for instance, if $Y_{12}$ = true, then task 1 is assigned to equipment 2. This problem has 9 disjunctions, 38 terms, 33 Boolean variables, and 51 continuous variables. There are 5 equipment, 4 tasks, and 3 products. For a detailed explanation of the variables, constraints and structure of the optimal solution, see Birewar and Grossmann (1990). In the above model, the concave objective function is relaxed by a secant line and the bilinear terms are relaxed by linear underestimators.

In step 2, we branch on $YEX_j$ first since they are the major decision variables. When all $YEX_j$ are fixed, then we branch on the task assignments variable $Y_{tj}$. The number of parallel equipment $N_j^{EQ}$, which is an integer variable, is treated as a continuous variable between 0 and 4 in the convex relaxation. Thus $N_j^{EQ}$ is not used as a branching variable in step 2, but we branch on $N_j^{EQ}$ in step 3. As shown in Table 2, an initial upper bound of 277,928 is found after four major iterations. With this upper bound, the bound contraction step yields 9.1 % reduction with 44 subproblems. The discrete branch and bound takes 32 nodes and 11 spatial branch and bound are performed. As shown in Table 3, the optimal solution has an objective value of 264,887 and the total CPU time required is less than one minute. For comparison, the heuristic search by DICOPT++ for solving the nonconvex MINLP yields the optimal solution in 5 major iterations with 3.7 CPU sec, but global optimality is not proved. The optimal design of the batch plant is shown in Figure 10.

It is interesting to note that when $YEX_j$ and $Y_{tj}$ are fixed, problem (E5) can be reformulated into a convex MINLP problem by substituting continuous variables with exponential terms through logarithmic transformations (Kocis and Grossmann, 1988). Then instead of using spatial branch and bound for nonconvex problem, OA algorithm can be used for solving this convex MINLP problem in step 3 where the only integer variables are $N_j^{EQ}$. Table 5 shows the numerical results and comparison of SBB and OA in step 3. A substantial CPU time is saved in step 3 with OA method, which shows that the convex reformulation can enhance the global search in this specific case.

**Example 6**

This example is again taken from Birewar and Grossmann (1990), but the number of units, tasks, and products is larger. There are 7 equipment, 6 tasks, and 6 products. Figure 11 shows the potential task assignments to equipment. The GDP model is as follows:

$$\min \mathrm{COST} = \sum_{j=1}^{M} N_j^{EQ} C_j$$

$$s.t. \qquad V_t^T \geq B_i S_{it} \quad i = 1,\ldots,N_P; t = 1,\ldots,T$$

$$pt_{ij} = \sum_{t \in T_j} pty_{itj} \quad i = 1,\ldots,N_P;\ j = 1,\ldots,M$$

$$n_i B_i \geq Q_i \quad i = 1,\ldots,N_P$$

$$\sum_{i=1}^{N_p} n_i T_{Li} \leq H$$

$$\begin{bmatrix} Y_{11} \\ V_1 \geq V_1^T \\ pty_{i11} = pt_{i1}^T \\ pty_{i1j} = 0, j \neq 1 \end{bmatrix} \vee \begin{bmatrix} Y_{12} \\ V_2 \geq V_1^T \\ pty_{i12} = pt_{i1}^T \\ pty_{i1j} = 0, j \neq 2 \end{bmatrix} \vee \begin{bmatrix} Y_{14} \\ V_4 \geq V_1^T \\ pty_{i14} = pt_{i1}^T \\ pty_{i1j} = 0, j \neq 4 \end{bmatrix} \vee \begin{bmatrix} Y_{15} \\ V_5 \geq V_1^T \\ pty_{i15} = pt_{i1}^T \\ pty_{i1j} = 0, j \neq 5 \end{bmatrix} \vee \begin{bmatrix} Y_{17} \\ V_7 \geq V_1^T \\ pty_{i17} = pt_{i1}^T \\ pty_{i1j} = 0, j \neq 7 \end{bmatrix}$$

$$\begin{bmatrix} Y_{22} \\ V_2 \geq V_2^T \\ pty_{i22} = pt_{i2}^T \\ pty_{i27} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{27} \\ V_7 \geq V_2^T \\ pty_{i27} = pt_{i2}^T \\ pty_{i22} = 0 \end{bmatrix} \qquad (E6)$$

$$\begin{bmatrix} Y_{33} \\ V_3 \geq V_3^T \\ pty_{i33} = pt_{i3}^T \end{bmatrix}$$

$$\begin{bmatrix} Y_{41} \\ V_1 \geq V_4^T \\ pty_{i41} = pt_{i4}^T \\ pty_{i4j} = 0, j \neq 1 \end{bmatrix} \vee \begin{bmatrix} Y_{42} \\ V_2 \geq V_4^T \\ pty_{i42} = pt_{i4}^T \\ pty_{i4j} = 0, j \neq 2 \end{bmatrix} \vee \begin{bmatrix} Y_{44} \\ V_4 \geq V_4^T \\ pty_{i44} = pt_{i4}^T \\ pty_{i4j} = 0, j \neq 4 \end{bmatrix}$$

$$\vee \begin{bmatrix} Y_{45} \\ V_5 \geq V_4^T \\ pty_{i45} = pt_{i4}^T \\ pty_{i4j} = 0, j \neq 5 \end{bmatrix} \vee \begin{bmatrix} Y_{47} \\ V_7 \geq V_4^T \\ pty_{i47} = pt_{i4}^T \\ pty_{i4j} = 0, j \neq 7 \end{bmatrix}$$

$$\begin{bmatrix} Y_{55} \\ V_5 \geq V_5^T \\ pty_{i55} = pt_{i5}^T \\ pty_{i57} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{57} \\ V_7 \geq V_5^T \\ pty_{i57} = pt_{i5}^T \\ pty_{i55} = 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_{62} \\ V_2 \geq V_6^T \\ pty_{i62} = pt_{i6}^T \\ pty_{i6j} = 0, j \neq 2 \end{bmatrix} \vee \begin{bmatrix} Y_{66} \\ V_6 \geq V_6^T \\ pty_{i66} = pt_{i6}^T \\ pty_{i6j} = 0, j \neq 6 \end{bmatrix} \vee \begin{bmatrix} Y_{67} \\ V_7 \geq V_6^T \\ pty_{i67} = pt_{i6}^T \\ pty_{i6j} = 0, j \neq 7 \end{bmatrix}$$

$$\left[\begin{array}{c} YEX_j \\ C_j = g_j + a_j V_j^{0.6} \\ V_j^L \le V_j \le V_j^U \\ \begin{bmatrix} YC_{1j} \\ N_j^{EQ} = 1 \\ T_{Li} \ge pt_{ij} \end{bmatrix} \vee \begin{bmatrix} YC_{2j} \\ N_j^{EQ} = 2 \\ 2T_{Li} \ge pt_{ij} \end{bmatrix} \vee \begin{bmatrix} YC_{3j} \\ N_j^{EQ} = 3 \\ 3T_{Li} \ge pt_{ij} \end{bmatrix} \vee \begin{bmatrix} YC_{4j} \\ N_j^{EQ} = 4 \\ 4T_{Li} \ge pt_{ij} \end{bmatrix} \end{array}\right] \vee \begin{bmatrix} \neg YEX_j \\ C_j = 0 \\ V_j = 0 \\ N_j^{EQ} = 0 \\ pt_{ij} = 0 \\ T_{Li} \ge 0 \end{bmatrix} \quad j = 1,\ldots,M$$

Logic Propositions

$$YEX_1 \Leftrightarrow Y_{11} \vee Y_{14} \;, YEX_2 \Leftrightarrow Y_{12} \vee Y_{22} \vee Y_{42} \vee Y_{62}$$

$$YEX_3 \Leftrightarrow Y_{33} \;, YEX_4 \Leftrightarrow Y_{14} \vee Y_{44}$$

$$YEX_5 \Leftrightarrow Y_{15} \vee Y_{45} \vee Y_{55} \;, YEX_6 \Leftrightarrow Y_{66}$$

$$YEX_7 \Leftrightarrow Y_{17} \vee Y_{27} \vee Y_{47} \vee Y_{57} \vee Y_{67}$$

$$W_{02} \underline{\vee} W_{12} \underline{\vee} W_{22}$$

$$W_{02} \Leftrightarrow \neg Y_{12} \wedge \neg Y_{22} \wedge \neg Y_{42} \wedge \neg Y_{62}$$

$$W_{12} \Leftrightarrow Y_{12} \underline{\vee} Y_{22} \underline{\vee} Y_{42} \underline{\vee} Y_{62}$$

$$W_{22} \Leftrightarrow Y_{12} \wedge Y_{22} \wedge \neg Y_{42} \wedge \neg Y_{62}$$

$$W_{05} \underline{\vee} W_{15} \underline{\vee} W_{25}$$

$$W_{05} \Leftrightarrow \neg Y_{15} \wedge \neg Y_{45} \wedge \neg Y_{55}$$

$$W_{15} \Leftrightarrow Y_{15} \underline{\vee} Y_{45} \underline{\vee} Y_{55}$$

$$W_{25} \Leftrightarrow \neg Y_{15} \wedge Y_{45} \wedge Y_{55}$$

$$W_{07} \underline{\vee} W_{17} \underline{\vee} W_{27} \underline{\vee} W_{37}$$

$$W_{07} \Leftrightarrow \neg Y_{17} \wedge \neg Y_{27} \wedge \neg Y_{47} \wedge \neg Y_{57} \wedge \neg Y_{67}$$

$$W_{17} \Leftrightarrow Y_{17} \underline{\vee} Y_{27} \underline{\vee} Y_{47} \underline{\vee} Y_{57} \underline{\vee} Y_{67}$$

$$W_{27} \Leftrightarrow (Y_{17} \wedge Y_{27} \wedge \neg Y_{47} \wedge \neg Y_{57} \wedge \neg Y_{67}) \underline{\vee}$$

$$(\neg Y_{17} \wedge \neg Y_{27} \wedge Y_{47} \wedge Y_{57} \wedge \neg Y_{67}) \underline{\vee} (\neg Y_{17} \wedge \neg Y_{27} \wedge \neg Y_{47} \wedge Y_{57} \wedge Y_{67})$$

$$W_{37} \Leftrightarrow \neg Y_{17} \wedge \neg Y_{27} \wedge Y_{47} \wedge Y_{57} \wedge Y_{67}$$

$$0 \le C_j, V_j, V_t^T, n_i, B_i, T_{Li}, pt_{ij}, N_j^{EQ}, pty_{itj};\; YEX_{tj}, Y_{tj}, YC_{cj}, W_{lj} \in \{true,\ false\}$$

$$i = 1,\ldots,N_P;\, t = 1,\ldots,T;\; j = 1,\ldots,M;\, l = 0,1,2,3$$

The model has 13 disjunctions, 54 terms, 53 Boolean variables, and 105 continuous variables. Detailed results from each step are shown in Table 2. For exhaustive enumeration, there are more than a hundred feasible discrete choices for the Boolean variables, but the proposed algorithm performed only 46 spatial branch and bound step and proved the optimality. The global optimal solution is found in 163.7 CPU sec, while DICOPT++ failed to find the optimal solution after twenty major iterations (see Table 4). The optimal solution found by the proposed method, 726,205 is significantly lower than the solution obtained by DICOPT++ (763,450). As shown in Table 4, the first lower bound predicted by the proposed method is tighter than the lower bound from the big-M nonconvex MINLP problem. Figure 12 shows the optimal design. The convex MINLP reformulation, which is used in example 5, can be also used in this example. Table 5 shows the comparison of the SBB and OA algorithms in step 3. The difference in total CPU time comes from step 3 only. Again, the convex reformulation requires less CPU time compared with the spatial branch and bound method for solving the nonconvex MINLP problem.

Problem (E6) can be expanded by considering the storage tank after each unit *j*. We introduce the disjunction for the storage tank *j* which can be used for holding the intermediate products coming from unit *j*. The following disjunction and logic propositions are added to problem (E6):

$$\text{Disjunction for Storage Tank } j$$

$$\begin{bmatrix} YS_j \\ -f \le B_{ij} - B_{ij'} \le f \\ VST_j \ge S'_{ij}\, B_{ij} NEQ_j \\ VST_j \ge S'_{ij}\, B_{ij'} NEQ_{j'} \\ 100 \le VST_j \le 25000 \\ CS_j = 5000 + 80 VST_j^{0.5} \end{bmatrix} \vee \begin{bmatrix} \neg YS_j \\ B_{ij} = B_{ij'} \\ VST_j = 0 \\ CS_j = 0 \end{bmatrix} \qquad (D6)$$

$$f = 2000, \quad S'_{ij} = 1.2$$

$$\text{Logic Propositions}$$

$$\neg YEX_j \Rightarrow \neg YS_j$$

$$\text{Cost term in the objective}$$

$$COST = \sum_{j=1}^{M} CS_j$$

$$YS_j \in \{true,\ false\}, 0 \le CS_j, VST_j;\ j = 1,...,M$$

where $YS_j$ is the Boolean variable which is true if there is a storage tank after unit *j*. $CS_j$ is the cost variable for the storage tank *j*. When introducing the storage tanks, we disaggregate the variables $B_i$ and $n_i$ for batch size and number of batches into $B_{ij}$ and $n_{ij}$ for each unit *j* because now they can be different from unit to unit. If we use the storage tank *j*, then the batch sizes $B_{ij}$ before and after the storage tank can be different within a certain bound *f*. Otherwise, the batch sizes $B_{ij}$ should be same. The cost of storage tank *j* is given as a function of the volume, $VST_j$, which should be greater than the volume of the incoming products. With this augmentation, the problem is expected to have an optimal cost which is lower than the previous optimal solution 726,205 (without storage tank) because the storage tank decouples the process, and thus the equipment can be utilized more effectively.

First, the proposed two-level branch and bound algorithm was applied, and the search stopped due to the iteration limit in the spatial tree search. The best upper bound after termination is 671,284. Since the search is incomplete, this solution is not proved to be global optimum. Secondly, the convex MINLP reformulation for SBB/OA method was again applied to this problem. The global optimal solution 662,590 was found. This solution has 3 storage tanks in the process, and has a lower cost compared to 726,205 for no storage tanks. The number of subproblems and total CPU time were reduced with a factor of about 2 by the convex reformulation compared with the two-level branch and bound method.

**Example 7**

This example is a nonconvex MINLP problem for a heat exchanger network superstructure optimization (Yee and Grossmann, 1990; Zamora, 1997) as shown in Figure 13. For the GDP model, disjunctions are used for only the heaters and coolers, leading to a hybrid MINLP/GDP model (see Vecchietti and Grossmann, 1999). Big-M constraints are used for heat exchangers where the linear constraints give a relaxation comparable to the one when using the convex hull of the disjunctions. The arithmetic mean temperature is used in the objective function instead of the LMTD. For a detailed explanation of the problem data and superstructure, see example 2.4 in Zamora (1997). The hybrid model is as follows:

$$\min Z = \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} CF_{ij} z_{ijk} + \sum_{i\in I} CF_{i,cu} zcu_i + \sum_{j\in J} CF_{j,hu} zhu_j + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} C_{ij}\left[\frac{q_{ijk}}{U_{ij}\left[\frac{dt_{ijk}+dt_{ijk+1}}{2}\right]}\right]$$

$$+\sum_{i\in I} C_{icu}\left[\frac{q_{icu}}{U_{icu}\left[\frac{dtcu_i+(TOUT_i-TIN_{cu})}{2}\right]}\right] + \sum_{j\in J} C_{jhu}\left[\frac{q_{jhu}}{U_{jhu}\left[\frac{dthu_j+(TIN_{hu}-TOUT_j)}{2}\right]}\right]$$

$$+\sum_{i\in I} CCUqcu_i + \sum_{j\in J} CHUqhu_j$$

$$s.t.\quad \sum_{j\in J}\sum_{k\in K} q_{ijk} + qcu_i = FH_i(TH_{i,in} - TH_{i,out}) \quad i\in I$$

$$\sum_{i\in I}\sum_{k\in K} q_{ijk} + qhu_j = FC_j(TC_{j,out} - TC_{j,in}) \quad j\in J$$

$$\sum_{j\in J} q_{ijk} = FH_i(th_{i,k} - th_{i,k+1}) \quad i\in I, k\in K$$

$$\sum_{i\in I} q_{ijk} = FC_j(tc_{j,k} - tc_{j,k+1}) \quad j\in J, k\in K$$

$$th_{i,1} = TH_{i,in} \quad i\in I;\ tc_{j,NOK+1} = TC_{j,in} \quad j\in J$$

$$th_{i,k} \ge th_{i,k+1} \quad i\in I, k\in K;\ th_{i,NOK+1} \ge TH_{i,out} \quad i\in I$$

$$tc_{j,k} \ge tc_{j,k+1} \quad j\in J, k\in K;\ TC_{j,out} \ge tc_{j,1} \quad j\in J$$

$$\begin{bmatrix} Y_{cu1} \\ qcu_1 = FH_1(th_{1,NOK+1} - TH_{1,out}) \\ dtcu_1 \le th_{1,NOK+1} - T_{cu,out} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{cu1} \\ qcu_1 = 0, dtcu_1 \ge \Delta T_{mapp} \\ th_{1,NOK+1} = TH_{1,out} \end{bmatrix}$$

$$\begin{bmatrix} Y_{cu2} \\ qcu_2 = FH_2(th_{2,NOK+1} - TH_{2,out}) \\ dtcu_2 \le th_{2,NOK+1} - T_{cu,out} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{cu2} \\ qcu_2 = 0, dtcu_2 \ge \Delta T_{mapp} \\ th_{2,NOK+1} = TH_{2,out} \end{bmatrix} \tag{E7}$$

$$\begin{bmatrix} Y_{hu1} \\ qhu_1 = FC_1(TC_{1,out} - tc_{1,1}) \\ dthu_1 \le T_{hu,out} - tc_{1,1} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{hu1} \\ qhu_1 = 0, dthu_1 \ge \Delta T_{mapp} \\ tc_{1,1} = TC_{1,out} \end{bmatrix}$$

$$\begin{bmatrix} Y_{hu2} \\ qhu_2 = FC_2(TC_{2,out} - tc_{2,1}) \\ dthu_2 \le T_{hu,out} - tc_{2,1} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{hu2} \\ qhu_2 = 0, dthu_2 \ge \Delta T_{mapp} \\ tc_{2,1} = TC_{2,out} \end{bmatrix}$$

$$dt_{ijk} \geq \Delta T_{mapp} \quad i \in I, j \in J, k \in K$$
$$dtcu_i \geq \Delta T_{mapp} \quad i \in I$$
$$dthu_j \geq \Delta T_{mapp} \quad j \in J$$
$$q_{ijk} \leq \Omega z_{ijk} \quad i \in I, j \in J, k \in K$$
$$dt_{ijk} \leq th_{i,k} - tc_{j,k} + \Gamma(1 - z_{ijk}) \quad i \in I, j \in J, k \in K$$
$$dt_{ijk+1} \leq th_{i,k+1} - tc_{j,k+1} + \Gamma(1 - z_{ijk}) \quad i \in I, j \in J, k \in K$$
$$\sum_{j \in J} z_{ijk} \leq 1, \quad i \in I, k \in K$$
$$\sum_{i \in I} z_{ijk} \leq 1, \quad j \in J, k \in K$$
$$z_{ijk} \in \{0,1\}; Y_{cui}, Y_{huj} \in \{true,\ false\}$$
$$TH_{i,out} \leq th_{i,k} \leq TH_{i,in}, \quad i \in I; TC_{j,in} \leq tc_{j,k} \leq TC_{j,out}, \quad j \in J$$
$$0 \leq q_{ijk}, qcu_i, qhu_j, \quad i \in I, j \in J, k \in K$$

The optimal solution is Z* = 74,710 $/yr which is shown in Figure 14. All the constraints are linear and only the objective function is nonconvex. Zamora (1997) used thermodynamic based linear fractional underestimators for the area of heat exchangers. Here we transform the nonconvex terms in the objective function into linear fractional terms and use nonlinear convex underestimators by Quesada and Grossmann (1995) as follows:

$$
\begin{aligned}
&A_{i,j,k} \geq \frac{q_{ijk}}{fq^{U}(dt_{ijk}, dt_{ijk+1})} + \frac{q_{ijk}^{L}}{fq(dt_{ijk}, dt_{ijk+1})} - \frac{q_{ijk}^{L}}{fq^{U}(dt_{ijk}, dt_{ijk+1})} \\
&A_{i,j,k} \geq \frac{q_{ijk}}{fq^{L}(dt_{ijk}, dt_{ijk+1})} + \frac{q_{ijk}^{U}}{fq(dt_{ijk}, dt_{ijk+1})} - \frac{q_{ijk}^{U}}{fq^{L}(dt_{ijk}, dt_{ijk+1})} \\
&ACU_{i} \geq \frac{qcu_{i}}{fc^{U}(dtcu_{i})} + \frac{qcu_{i}^{L}}{fc(dtcu_{i})} - \frac{qcu_{i}^{L}}{fc^{U}(dtcu_{i})} \\
&ACU_{i} \geq \frac{qcu_{i}}{fc^{L}(dtcu_{i})} + \frac{qcu_{i}^{U}}{fc(dtcu_{i})} - \frac{qcu_{i}^{U}}{fc^{L}(dtcu_{i})} \\
&AHU_{j} \geq \frac{qhu_{j}}{fh^{U}(dthu_{j})} + \frac{qhu_{j}^{L}}{fh(dthu_{j})} - \frac{qhu_{j}^{L}}{fh^{U}(dthu_{j})} \qquad \text{(R7)} \\
&AHU_{j} \geq \frac{qhu_{j}}{fh^{L}(dthu_{j})} + \frac{qhu_{j}^{U}}{fh(dthu_{j})} - \frac{qhu_{j}^{U}}{fh^{L}(dthu_{j})} \\
&\textit{where} \\
&fq(dt_{ijk}, dt_{ijk+1}) = U_{ij}\left[\frac{dt_{ijk} + dt_{ijk+1}}{2}\right] \\
&fc(dtcu_{i}) = UCU_{i}\left[\frac{dtcu_{i} + (TOUT_{i} - TIN_{cu})}{2}\right] \\
&fh(dthu_{j}) = UHU_{j}\left[\frac{dthu_{j} + (TIN_{hu} - TOUT_{j})}{2}\right]
\end{aligned}
$$

where $A_{ijk}$ are the areas of heat exchangers, $ACU_i$ are the areas of coolers, and $AHU_j$ are the areas of heaters. Since the functions *fq*, *fc,* and *fh* in (R7) are linear functions, their upper and lower bounds can be calculated from the upper and lower bounds of the continuous variables $dt_{ijk}$, $dtcu_i$, and $dthu_j$. The convex lower bounding problem (CRP) of problem (E7) yields the solution 28,260 \$/yr which is an initial GLB.

As shown in Table 2, DICOPT++ finds a feasible solution of 75,696 \$/yr after 4 major iterations. The bound contraction step yields 49 % reduction of the feasible region of continuous variables. In step 2, the branch and bound on the discrete variables searched 517 nodes. A total of 60 feasible discrete choices to problem (R) are found and they are solved by the spatial branch and bound method in step 3. In this example, the main difficulty during the search comes from the nonconvex objective terms, not from the linear constraints. Much of effort is spent to update the upper bound in the spatial branch and bound step. Among all steps, 71 % of the total CPU time is spent in step 3, which means that we might accelerate the search if we can reduce the number of subproblems to be solved in step 3.

## Conclusion

A global optimization algorithm for nonconvex GDP problems that involve bilinear, linear fractional and concave separable functions has been proposed in this paper. The nonconvex terms are substituted by convex underestimators for constructing a convex GDP problem. The convex hull relaxation of the GDP is introduced resulting in a convex NLP that yields lower bounds. This NLP problem is solved at the nodes of the proposed two-level branch and bound method in which the disjunctive branch and bound method by Lee and Grossmann (2000) is used combined with a spatial branch and bound method for the continuous variables. Tight lower bounds from the convex hull relaxation and the bound contraction, along with an upper bound have been shown to be effective in solving nonconvex discrete/continuous optimization problems. Numerical results of 7 example problems have been shown, as well as comparisons with the big-M MINLP model.

## Acknowledgements

The authors would like to acknowledge financial support from the NSF Grant CTS-9710303.

## References

Adjiman C.S., I.P. Androulakis and C.A. Floudas, Global Optimization of MINLP Problems in Process Synthesis and Design. *Computers and Chem. Engng.*, **21**, Suppl., S445-S450, 1997.

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

Adjiman C.S. and C.A. Floudas, Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, **9**(1), 23-40. 1996.

Al-Khayyal F.A. and J.E. Falk, Jointly constrained biconvex programming. *Mathematics of Operations Research*, **8**, 273-286, 1983.

Bazaraa M., H. Sherali and C. Shetty, *Nonlinear Programming, Theory and Applications*. John Wiley & Sons, New York, second ed., 1993.

Birewar D.B. and I.E. Grossmann, Simultaneous Synthesis, Sizing, and Scheduling of Multiproduct Batch Plants. *Ind. Eng. Chem. Res.*, **29**(11), 2242-2251, 1990.

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

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

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

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.

Hiriart-Urruty J. and C. Lemaréchal, *Convex Analysis and minimization algorithms*, vol. 1. Springer-Verlag, 1993.

Horst R. and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed., Springer-Verlag, Berlin, 1996.

Kesavan P. and P.I. Barton, Decomposition algorithms for nonconvex mixed-integer nonlinear programs. *American Institute of Chemical Engineering Symposium Series*, (in press) 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, Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis, *Ind. Eng. Chem. Res*., **27**, 1407-1421, 1988.

Kocis G.R. and I.E. Grossmann, A Modelling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets. *Computers and Chem. Engng*., **13**(7), 797-819, 1989.

Lee S. and I.E. Grossmann, New Algorithms for Nonlinear Generalized Disjunctive Programming. *Computers and Chem. Engng*., **24**(9-10), 2125-2141, 2000.

Lee S. and I.E. Grossmann, Generalized Disjunctive Programming: Nonlinear Convex Hull Relaxation and Algorithms. To be submitted for publication, 2001.

McCormick G.P., Computability of Global Solutions to Factorable Nonconvex Programs: Part I – Convex Underestimating Problems. *Mathematical Programming*, **10**, 147-175, 1976.

Quesada I. and I.E. Grossmann, A Global Optimization Algorithm for Linear Fractional and Bilinear Programs. *Journal of Global Optimization*, **6**(1), 39-76, 1995.

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

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

Tawarmalani M. and N.V. Sahinidis, Semidefinite Relaxations of Fractional Programs via Novel Techniques for Constructing Convex Envelopes of Nonlinear Functions, Submitted to *Journal of Global Optimization*, May 2000b.

Vecchietti A. and I.E. Grossmann, Modeling issues and implementation of language for disjunctive programming. *Computers and Chem. Engng*., **24**(9-10), 2143-2155, 2000.

Viswanathan J. and I.E. Grossmann, A combined penalty function and outer-approximation method for MINLP optimization. *Computers and Chem. Engng*., **14**, 769-782, 1990.

Yee T.F. and I.E. Grossmann, Simultaneous Optimization Models for Heat Integration-II. Heat Exchanger Network Synthesis. *Computers and Chem. Engng*., **14**(10), 1165-1184, 1990.

Zamora J.M., Global Optimization of Nonconvex NLP and MINLP Models, Ph. D. Thesis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA15213, 1997.

Zamora J.M. and I.E. Grossmann, A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. *Computers and Chem. Engng.*, **22**(13), 367-384, 1998a.

Zamora J.M. and I.E. Grossmann, Continuous Global Optimization of Structured Process Systems Models. *Computers and Chem. Engng.*, **22**(12), 1749-1770, 1998b.

Zamora J.M. and I.E. Grossmann, A Branch and Bound Algorithm for Problems with Concave Univariate, Bilinear and Linear Fractional Terms. *Journal of Global Optimization*, **14**(3), 217-249, 1999.

## Appendix A. Convex Underestimators for Bilinear/Linear Fractional/Concave Separable functions.

Consider a bilinear function $x_i x_j$. Valid convex/concave envelopes are as follows (McCormick, 1976;Al-Khayyal and Falk, 1983):

$$
\begin{aligned}
x_i x_j &\geq x_i^U x_j + x_i x_j^U - x_i^U x_j^U \\
x_i x_j &\geq x_i^L x_j + x_i x_j^L - x_i^L x_j^L \\
x_i x_j &\leq x_i^U x_j + x_i x_j^L - x_i^U x_j^L \\
x_i x_j &\leq x_i^L x_j + x_i x_j^U - x_i^L x_j^U
\end{aligned}
\qquad (A1)
$$

For a linear fractional function $x_i/x_i$, Quesada and Grossmann (1995) developed nonlinear convex underestimators.

$$
\begin{aligned}
\frac{x_i}{x_j} &\geq \frac{x_i}{x_j^U} + \frac{x_i^L}{x_j} - \frac{x_i^L}{x_j^U} \\
\frac{x_i}{x_j} &\geq \frac{x_i}{x_j^L} + \frac{x_i^U}{x_j} - \frac{x_i^U}{x_j^L}
\end{aligned}
\qquad (A2)
$$

Alternate convex underestimator/envelope are the following:

$$
\frac{x_i}{x_j} \geq \frac{1}{x_j}\left(\frac{x_i + \sqrt{x_i^L x_i^U}}{\sqrt{x_i^L} + \sqrt{x_i^U}}\right)^2 \qquad (A3)
$$

$$
\begin{aligned}
& z = \frac{x_i}{x_j} \\
& (z - z_p)(x_j - x_j^p)(x_i^U - x_i^L)^2 \geq x_i^U (x_i - x_i^L)^2 \qquad (A4) \\
& z_p \geq \frac{x_i^L \left( x_i^L x_j^p - x_i \left( x_j^L + x_j^U \right) + x_i^U \left( x_j^L - x_j^p + x_j^U \right) \right)}{(x_i^U - x_i^L) x_j^L x_j^U} \\
& x_j^L \left( x_i^U - x_i \right) \leq x_j^p \left( x_i^U - x_i^L \right) \leq x_j^U \left( x_i^U - x_i \right) \\
& x_j^L \left( x_i - x_i^L \right) \leq \left( x_j - x_j^p \right) \left( x_i^U - x_i^L \right) \leq x_j^U \left( x_i - x_i^L \right) \\
& z - z_p, z_p, x_j^p \geq 0
\end{aligned}
$$

(A3) and (A4) were developed by Zamora and Grossmann (1998a) and by Tawarmalani and Sahinidis (2000b), respectively. The convex envelope (A4) is based on the convex hull of $x_i^L/x_j$ and $x_i^U/x_j$.

Univariate concave separable function $l_i(x)$ can be underestimated by a secant line which matches concave function at the upper and lower bound (Falk and Soland, 1969):

$$
l_i(x) \geq l_i(x^L) + (x - x^L) \frac{l_i(x^U) - l_i(x^L)}{x^U - x^L} \qquad (A5)
$$

## Appendix B. Logic Propositions for Example 5.

### *B1. Equipment Units*

The logic relation between the existence of unit and the assignment of task can be stated as follows:

1) Unit *j*: If unit *j* exists, then at least one of the tasks *t*, which can be assigned to the unit *j*, should be assigned to unit *j*.
2) Task *t*: If task *t* is assigned to unit *j*, then unit *j* should exist.

In example 5, only task 1 (mixing) can be assigned to unit 1. The logic propositions for the Boolean variables $YEX_1$ are $Y_{11}$ then,

$$
\begin{aligned}
& YEX_1 \Rightarrow Y_{11} \\
& Y_{11} \Rightarrow YEX_1 \qquad (B1)
\end{aligned}
$$

which can be combined into one proportional logic as $YEX_1 \Leftrightarrow Y_{11}$. By applying the same procedure, we can produce the logic propositions for each unit (for the task assignment, see Figure 9).

$$\begin{aligned}
&YEX_1 \Leftrightarrow Y_{11} \\
&YEX_2 \Leftrightarrow Y_{12} \vee Y_{22} \\
&YEX_3 \Leftrightarrow Y_{33} \\
&YEX_4 \Leftrightarrow Y_{14} \vee Y_{24} \vee Y_{34} \\
&YEX_5 \Leftrightarrow Y_{45}
\end{aligned} \qquad (B2)$$

*B2. Assignments of Tasks*

The feasible choices in task assignment for unit 4 are considered. Unit 4 can process 3 types of tasks: Task 1, 2, and 3. A number of combinations of 3 tasks are feasible for the assignment. In terms of the number of tasks to be assigned, there are four choices: No task, 1 task, 2 tasks, and 3 tasks. For each case, we introduce Boolean variables $W_{04}$, $W_{14}$, $W_{24}$, and $W_{34}$. $W_{04}$ is true when no task is assigned to unit 4. Otherwise, it is false. Task assignment Boolean variable $Y_{14}$ is true when task 1 is assigned to unit 4. And it is same for $Y_{24}$ and $Y_{34}$. The logic propositions for task assignments are as follows:

$$\begin{aligned}
&W_{04} \underline{\vee} W_{14} \underline{\vee} W_{24} \underline{\vee} W_{34} \\
&W_{04} \Leftrightarrow \neg Y_{14} \wedge \neg Y_{24} \wedge \neg Y_{34} \\
&W_{14} \Leftrightarrow (Y_{14} \wedge \neg Y_{24} \wedge \neg Y_{34}) \underline{\vee} (\neg Y_{14} \wedge Y_{24} \wedge \neg Y_{34}) \underline{\vee} (\neg Y_{14} \wedge \neg Y_{24} \wedge Y_{34}) \\
&W_{24} \Leftrightarrow (Y_{14} \wedge Y_{24} \wedge \neg Y_{34}) \underline{\vee} (\neg Y_{14} \wedge Y_{24} \wedge Y_{34}) \\
&W_{34} \Leftrightarrow Y_{14} \wedge Y_{24} \wedge Y_{34}
\end{aligned} \qquad (B3)$$

The first logic statement means that the number of tasks which can be assigned to unit 4 is 0,1,2, or 3. The second logic proposition states that if we do not assign any task to unit 4, then all $Y_{t4}$ are false. The third logic proposition enforces that if we assign one task, then only one of $Y_{t4}$ is true ($t$ = 1,2,4) and others are false. The fourth logic enforces that if we assign two tasks to unit 4, only tasks (1, 2) or tasks (2, 3) can be assigned to unit 4. Tasks (1, 3) cannot be assigned to unit 4 without task 2 assigned. The last logic statement means that if we assign three tasks, then all $Y_{t4}$ are true. The logic propositions (B3) can be transformed into the integer linear constraints $A\lambda \leq a$ when solving problem (CRP).

We follow the same procedure in Example 6 for constructing the logic propositions for the task assignments to units 2, 5 and 7 as shown in the model (E6).

## List of Tables

## List of Figures

### Table 1. Numerical results of illustrative example.

| Step | Step 0 | Step 1 | Step 2 | Step 3 |
|---|---|---|---|---|
| Method | OA | Bound Contraction | Discrete BB | Spatial BB |
| Result | First UB = 99.2 | 63.5 % reduction | First LB = 97.5 | 1 SBB / UB = 99.2 |
| Iter. / Nodes | 3 Iter. | 4 Iter. (NLPs) | 2 Nodes (NLPs) | 5 Nodes (NLPs) |

### Table 2. Numerical results of nonconvex GDP problems.

| Example | Optimal | No. of var. | Step 0 | Step 1 | Step 2 | Step 3 | Total |
|---|---|---|---|---|---|---|---|
| No. | Solution | (cont./disc.) | First UB | prob./reduc. | First LB | Nodes | CPU |
|  |  | No. of const. | Major iter./CPU* | CPU | Nodes/CPU | /CPU | sec |
| 1 | -11.0 | 2/3 | -11.0 | 9/69.5 % | -11.25 | 1 SBB | 3.2 |
|  |  | 6 | 4/1.2 | 1.3 | 3/0.36 | 3/0.35 |  |
| 2 | -14.0 | 2/3 | -14.0 | 8/52.6 % | -37.36 | 1 SBB | 9.3 |
|  |  | 6 | 4/3.8 | 1.4 | 6/1.8 | 27/2.3 |  |
| 3 | -510.08 | 26/2 | 0.000 | 14/10.8 % | -684.56 | 1 SBB | 4.9 |
|  |  | 49 | 1/0.1 | 1.7 | 6/1.3 | 27/1.6 |  |
| 4 | -116,575 | 5/3 | -116,575 | 45/22.2 % | -242,474 | 4 SBB | 21.7 |
|  |  | 24 | 1/1.1 | 5.7 | 11/0.88 | 126/14.1 |  |
| 5 | 264,887 | 51/33 | 277,928 | 44/9.1 % | 82,354 | 11 SBB | 47.1 |
|  |  | 102 | 4/2.5 | 11.8 | 32/7.3 | 159/25.5 |  |
| 6 | 726,205 | 105/53 | 763,450 | 36/4.3 % | 341,284 | 46 SBB | 163.7 |
|  |  | 271 | 3/9.2 | 16.7 | 169/76.5 | 210/61.3 |  |
| 7 | 74,710 | 52/16 | 75,696 | 34/49 % | 28,260 | 60 SBB | 420.6 |
|  |  | 124 | 4/3.0 | 7.0 | 517/110.6 | 1494/300 |  |

*On Pentium II 300MHz with 128Mbyte RAM Memory.

### Table 3. Comparison of solution algorithms for example 5.

| Model | Solution Method | CPU sec | First LB | Solution | Note |
|---|---|---|---|---|---|
| Big-M nonconvex MINLP | DICOPT++ (local search) | 3.7 (5 Major Iter.) | 161,339 (not global) | 264,887 | Optimality not proved |
| Nonconvex GDP | Proposed Global Algorithm | 47.1 | 82,354 (globally valid) | 264,887 | Global Optimum |

### Table 4. Comparison of solution algorithms for example 6.

| Model | Solution Method | CPU sec | First LB | Solution | Note |
|---|---|---|---|---|---|
| Big-M nonconvex MINLP | DICOPT++ (local search) | 42.7 (20 Major Iter.) | 325,453 (not global) | 763,450 | Sub-optimal Solution |
| Nonconvex GDP | Proposed Global Algorithm | 163.7 | 341,284 (globally valid) | 726,205 | Global Optimum |

**Table 5. Convex MINLP reformulation for example 5 and 6.**

| Problem | Example 5 | | Example 6 | |
|---|---|---|---|---|
| Solution Method | Discrete BB/SBB<br>Iter./Nodes/sec | Discrete BB/OA<br>Iter./Nodes/sec | Discrete BB/SBB<br>Iter./Nodes/sec | Discrete BB/OA<br>Iter./Nodes/sec |
| Step 0 | 4 iter./2.5 | 4 iter./2.2 | 3 iter./9.2 | 3 iter./9.2 |
| Step 1 | 44 iter./11.8 | 44 iter./8.0 | 36 iter./16.7 | 36 iter./14.2 |
| Step 2 | 32 nodes/7.3 | 32 nodes/7.3 | 169 nodes/76.5 | 163 nodes/80.9 |
| Step 3 | 11 SBB/25.5 | 11 OA/8.8 | 46 SBB/61.3 | 46 OA/29.3 |
| Total CPU sec | 47.1 | 26.3 | 163.7 | 133.6 |

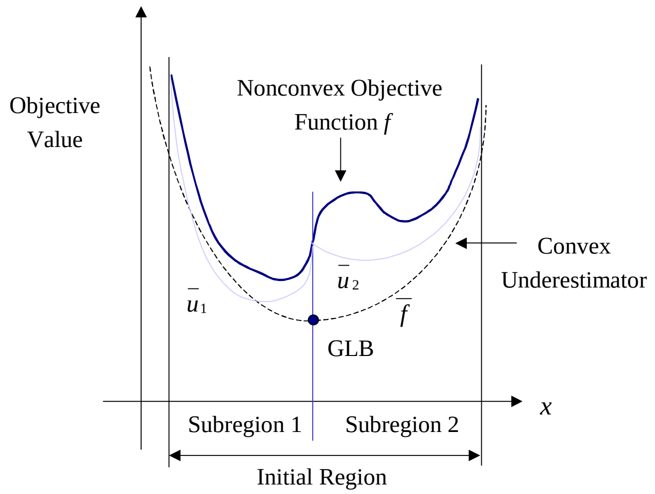

**Figure 1. Convex underesimator function.**

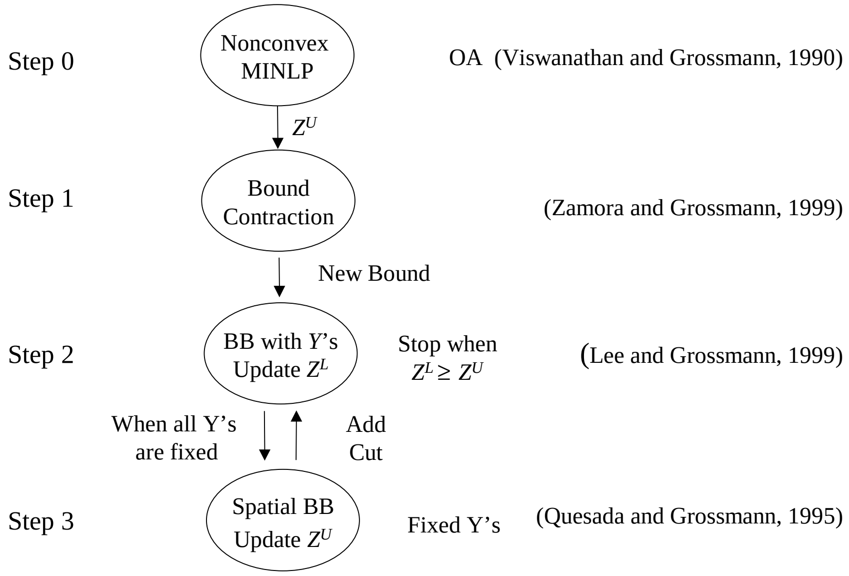

**Figure 2. Proposed algorithm for nonconvex GDP.**

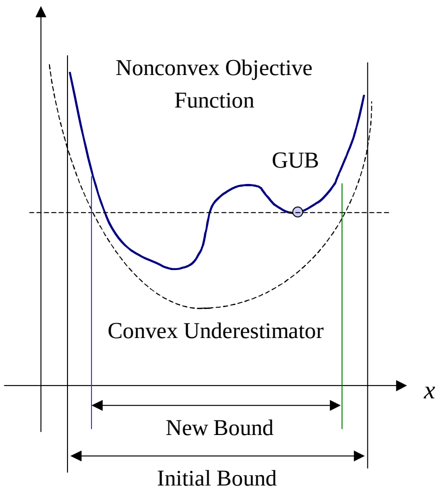

**Figure 3. Bound contraction subproblem.**

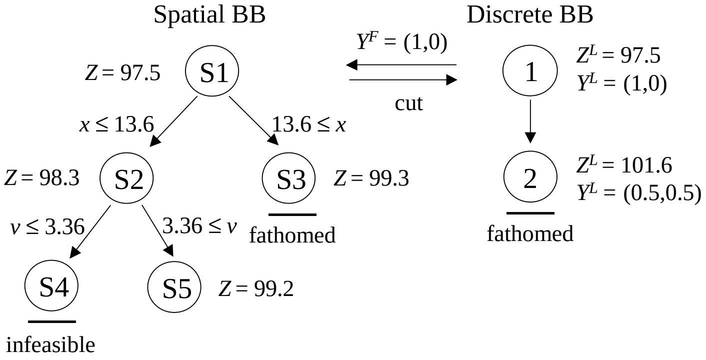

**Figure 4. Branch and bound tree: Illustrative example.**

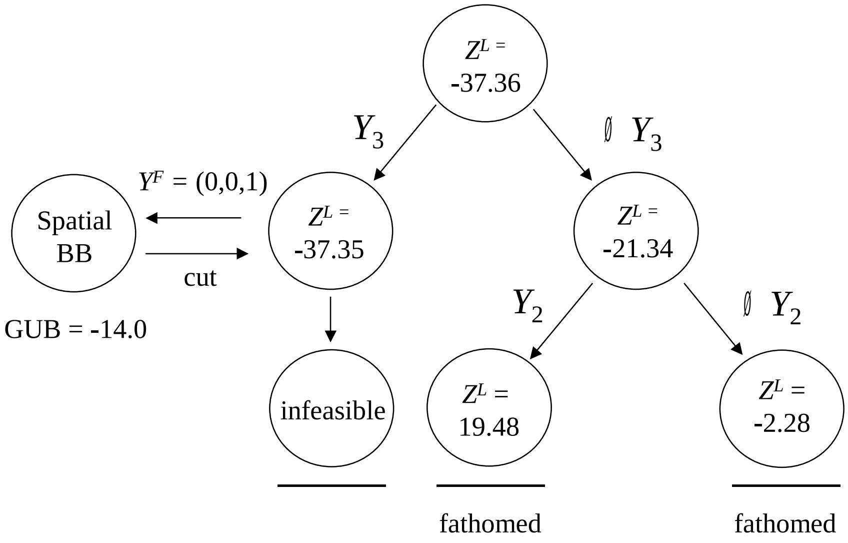

**Figure 5. Discrete branch and bound tree: Example 2.**

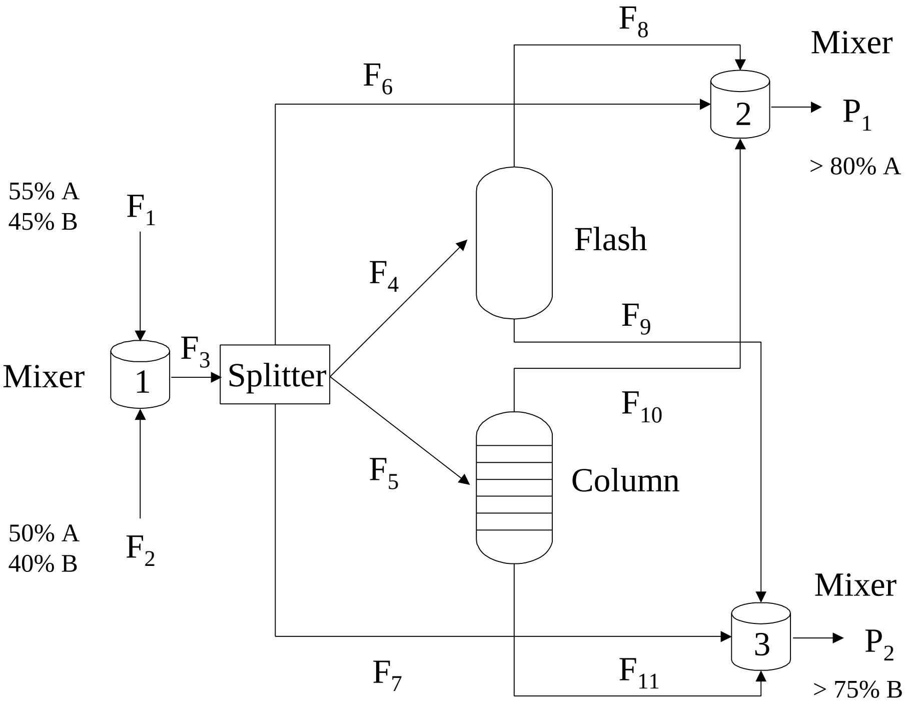

**Figure 6. Superstructure of example 3.**

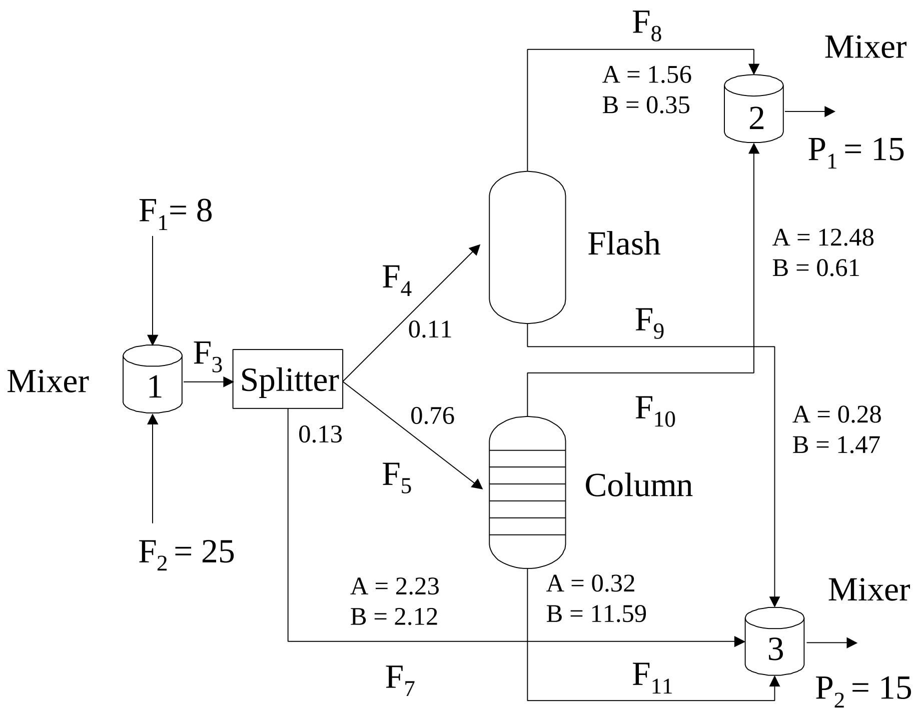

**Figure 7. Optimal solution of example 3.**

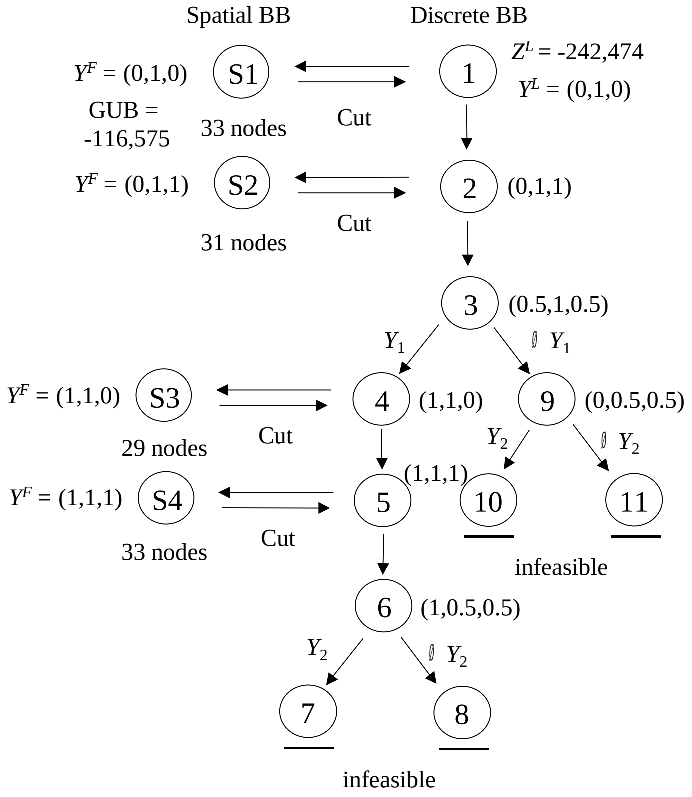

**Figure 8. Branch and bound tree of example 4.**

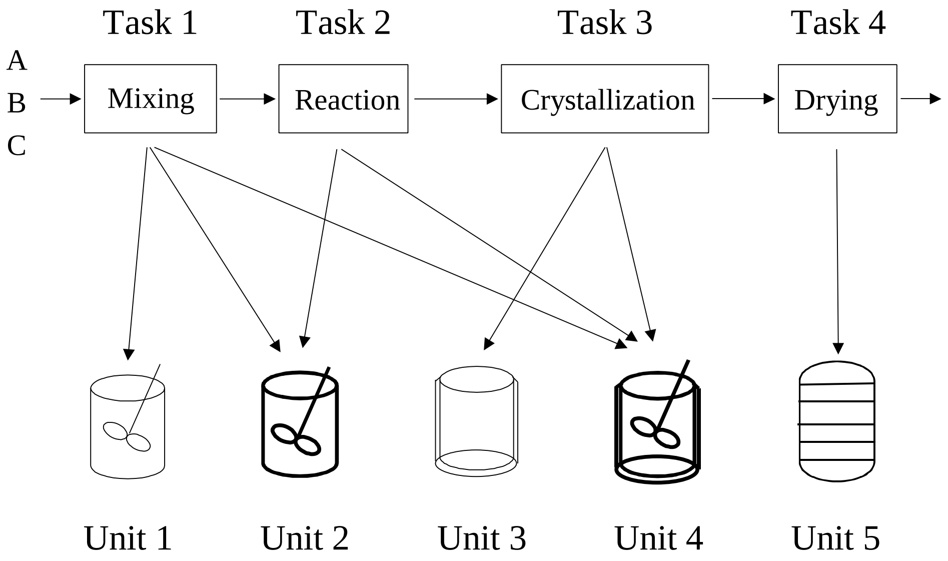

**Figure 9. Feasible task assignments of example 5.**

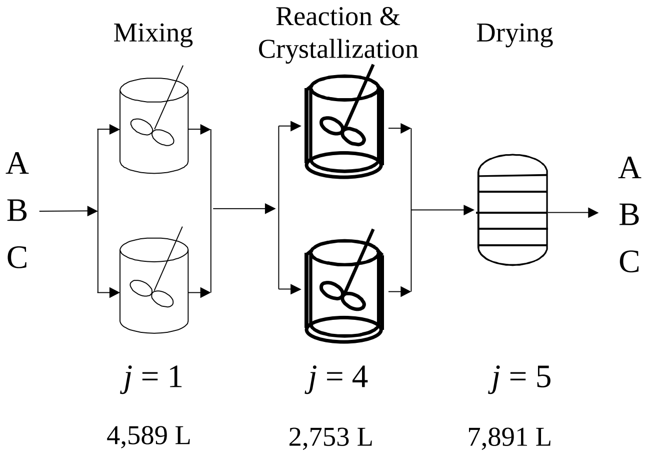

**Figure 10. Optimal batch plant design of example 5.**

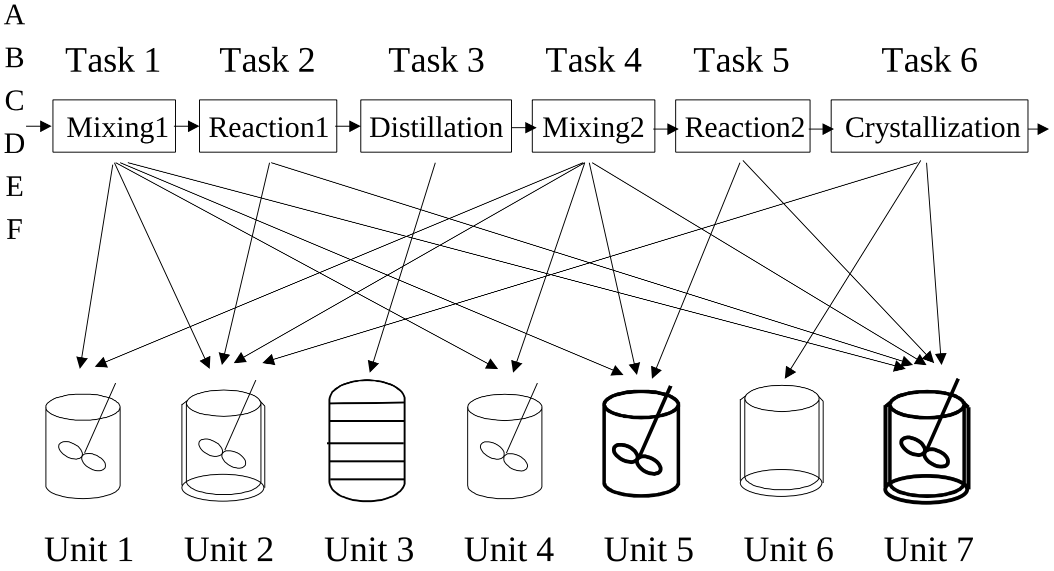

**Figure 11. Feasible task assignments of example 6.**

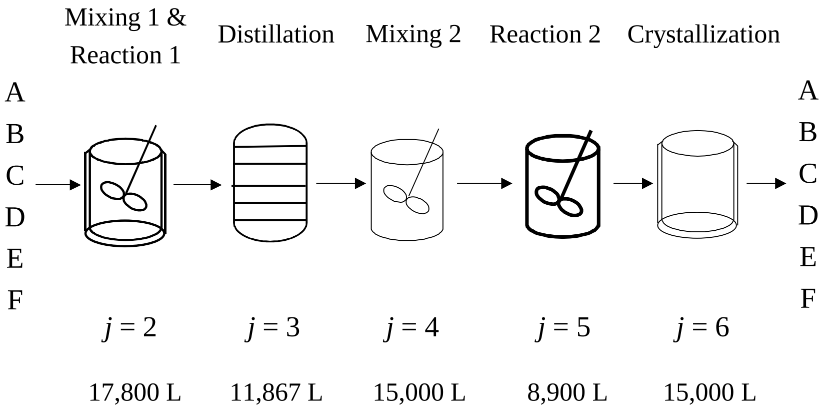

**Figure 12. Optimal batch plant design of example 6.**

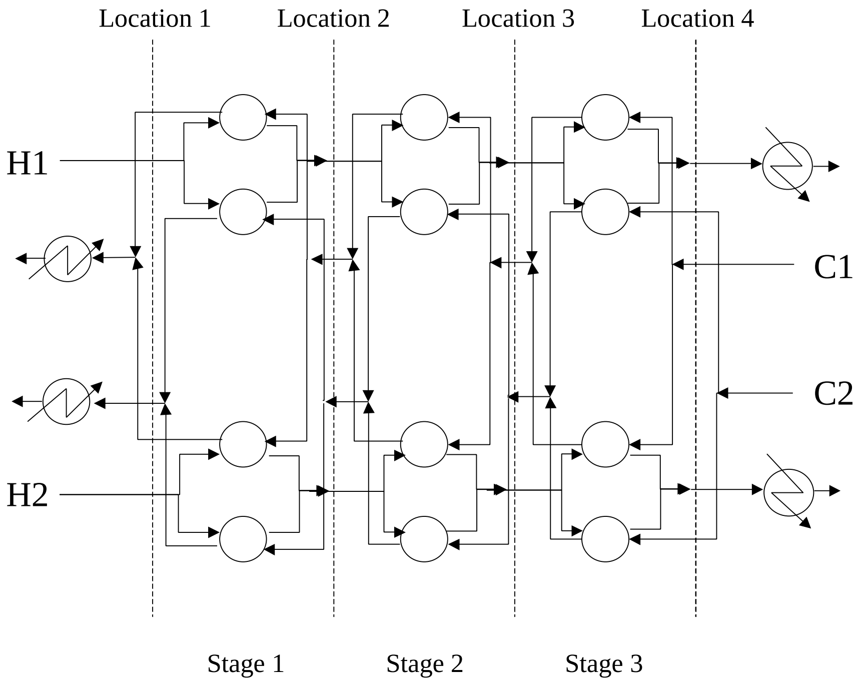

**Figure 13. Superstructure of heat exchanger network.**

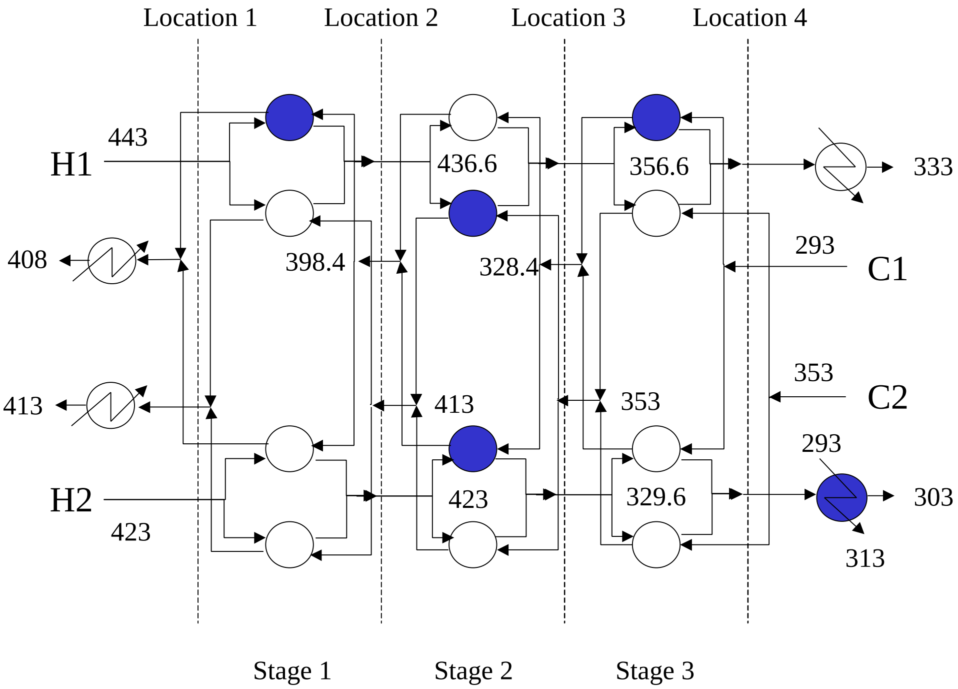

**Figure 14. Global optimal solution of example 7.**