Computational and Applied Mathematics
Vol. 20, N. 3, pp. 341–360, 2001

Algorithm for solution of convex MINLP problems

E. C. PEREIRA[1] and A. R. SECCHI[2]

[1]Departamento de Matemática, Fundação Universidade Federal do Rio Grande
96021-900 Rio Grande, RS, Brazil
E-mail: nani.elaine@bol.com.br

[2]Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul
90050-170 Porto Alegre, RS, Brazil
E-mail: arge@enq.ufrgs.br

Abstract. The current work shows the formulation and implementation of an algorithm for the solution of convex mixed-integer nonlinear programming (MINLP) problems. The proposed algorithm does not follow the traditional sequence of solutions of nonlinear programming (NLP) subproblems and master mixed-integer linear programming (MILP) problems. Instead, the master problem is defined dynamically during the tree search to reduce the number of nodes that need to be enumerated. A branch and bound search is performed to predict lower bounds by solving linear programming (LP) subproblems until feasible integer solutions are found. For these nodes, nonlinear programming subproblems are solved, providing upper bounds and new linear approximations, which are used to tighten the linear representation of the open nodes in the search tree. Numerical results for convex and nonconvex test problems are analyzed, comparing the efficiency of the proposed algorithm and the general algebraic modeling system (GAMS).

Mathematical subject classification: 90C30, 65K10, 49M37.

Key words: optimization, mixed-integer nonlinear programming, branch and bound search.

1 Introduction

Process system engineering is a rich area in optimization problems. Many problems of process design and process operation can be formulated as linear programming (LP), quadratic programming (QP), nonlinear programming

#513/01. Received: 26/I/01.

(NLP), mixed-integer linear programming (MILP), or mixed-integer nonlinear programming (MINLP). Among these formulations, the mixed-integer nonlinear programming consists the largest subset of the mathematical programming field, especially in examples of chemical engineering that involve heat exchanger network synthesis ([2], [5], [12], [20]), synthesis of process flowsheets ([18], [22]), and project of distillation columns ([9], [16]). The mathematical models of process engineering frequently involve discrete variables. Certain decisions are naturally discrete, for example, the number of trays in a distillation column. Optimization problems that involve discrete variables are formulated as mixed-integer linear and nonlinear programming [9].

Most of the numeric methods for the solution of MINLP problems are limited to the determination of a local minimum. Many of the process engineering applications lead to nonconvex MINLPs with multiples local minima. There exist numeric algorithms that, somehow, try to locate the global optimum of MINLP problems, as the outer approximation (OA) algorithm [6] and its extensions with the equality constraint relaxation (OA/ER) [14] and augmented penalty function (OA/ER/AP) [23]. Others important deterministic methods to solve MINLP problems include the spatial branch and bound [17], generalized Benders decomposition (GBD) ([1], [10]), generalized outer approximation (GOA) [8], and generalized cross decomposition (GCD) [13].

A common aspect of this class of MINLP problems is the nonconvexity, making difficult the determination of an optimal global solution with the use of most techniques of mathematical programming. On the other hand, it is important as a first step the development of enhanced methods for convex problems [19], that can be useful for the development of methods for the solution of nonconvex problems.

Due to the mentioned reasons, associated to economic and environmental factors, there has been a growing interest in developing and investigating new techniques for resolution of MINLP problems.

This work proposes a numeric algorithm to solve convex mixed-integer nonlinear problems. The numeric results of the convex and nonconvex examples obtained from experiments through the proposed algorithm and from the commercial software GAMS are compared.

2 Motivation for new algorithm

The methods of resolution of the MINLP problems can be classified in three main categories: *Branch and Bound, Generalized Benders Decomposition (GBD)* and *Outer Approximation (OA).*

The GBD and OA algorithms have the limitation that the size of the master problems (MILP) increases as the iterations proceed, being this a major drawback when the original MINLP has a large number of integer variables. The time used to solve the master problem increases as the iterations proceed, while the time for the NLP subproblems remains in the same order of magnitude.

An algorithm that avoids the above problems was developed by [19], improving the efficiency of the solution of convex MINLP problems and reducing the computational work demanded to solve the MILP master problems. The algorithm consists of a tree search over the space of the binary variables. The MILP master problem is defined dynamically during the tree search to reduce the number of nodes that need to be enumerated. A branch and bound search is carried out to determine lower bound in the solution of LP subproblems to find feasible integer solutions. For these nodes, NLP subproblems are resolved, determining upper bound and new linear approximations, which are used to extend the linear representation of the open nodes in the search tree. These linear approximations can be made in several ways.

The algorithm proposed in this work is based on the algorithm of [19] with modifications in the introduction of the linear approximations.

3 Detailed description of the algorithm

To describe the proposed algorithm, in detail, the following convex MINLP problem is considered.

$$z = \min_{x,y}\ c^T y + f(x)$$

subject to

$$\begin{aligned} & By + g(x) \leq 0 \\ & x \in X = \{x | x \in R^n, x^l < x < x^u\} \\ & y \in Y = \{y | y \in \{0,1\}^m, Ay \leq a\} \end{aligned} \tag{1}$$

For implementation of the algorithm, in the solution of (1), the following steps are followed:

Step 1 – Initial guesses for the binary variables y and continuous variables x are arbitrated. There is the option to let the algorithm chooses these values when they are not given.

Step 2 – Fixing $y = y_0$, the problem (1) becomes a NLP subproblem, that is:

$$\begin{aligned} & z = \min_x \; c^T y_0 + f(x) \\ \text{subject to} & \\ & By_o + g(x) \leq 0 \\ & x \in X \end{aligned} \tag{2}$$

The problem (2) is solved, finding a solution (x_0, y_0), which corresponds to a certain value "z" of the objective function. The value $zu = z$ is considered as the upper bound for the optimal solution of the MINLP problem (1).

If the constraint violation goes larger than ϵ_c, where ϵ_c is the tolerance of constraint violation, then makes $zu = \infty$, which means that this first NLP is considered infeasible.

In the implementation, when y_0 is not given, there is an alternative to solve a NLP with the integrality conditions over the binary variables relaxed, finding a solution (x_0, y_0). If the obtained y_0 is not integer, then it is rounded, and the problem (2) is solved.

Step 3 – The nonlinear functions of the problem (1) are linearized, using the optimal solution x_0 of the NLP subproblem (2), resulting in the following MILP problem.

$$\begin{aligned} & z = \min \; \alpha \\ \text{subject to} & \\ & \alpha \geq c^T y + f(x_0) + \nabla f(x_0)^T (x - x_0) \\ & By + g(x_0) + \nabla g(x_0)^T (x - x_0) \leq 0 \\ & \alpha \in R, \quad x \in X, \quad y \in Y \end{aligned} \tag{3}$$

Step 4 – The problem (3) becomes a LP problem when the integrality conditions on the binary variables are relaxed. The solution of this LP is considered as a lower bound zl for the optimal solution of the problem (1). If $z_l + \epsilon_z \geq zu$, where ϵ_z is a tolerance for the objective function, then this problem is removed from the tree. If $zl < zu$, this first LP problem is stored as the first node of the search tree and if the solution of this LP for the binary variables y is integer, go to step 8.

Step 5 – If there is no more LPs to be solved, that is, any opened node in the tree, then the current upper bound ''zu'' is the optimal solution of the problem (1) and terminates the algorithm.

Step 6 – If the solution of the LP for the binary variables is not integer, then there will be a branching in the search tree with the creation of two child nodes, that is, two LP problems. The creation of these LPs will be made in the following way: in the solution of the last LP_j^k, where j is the number of the node and k the number of the parent node, the y_i whose fractional value is the most distant from the extremes 0 and 1 is selected. The constraints $y_i = 0$ and $y_i = 1$ are added to the subproblems LP_{j+1}^j and LP_{j+2}^j, respectively. The problem is substituted by the two child subproblems in the list of LPs.

Step 7 – If at the end of the list of LPs there are two child problems to be resolved that have the same parent problem, then makes the following:

- Solve LP_{j+2}^j. If the value of the objective function $z_{j+2} + \epsilon_z > zu$, this problem is removed from the tree. In case of z_{j+2} be an integer solution, go to step 8;

- Solve LP_{j+1}^j . If the value of the objective function $z_{j+1} + \epsilon_z > zu$, this problem is removed from the tree. In case z_{j+1} be an integer solution, go to step 8;

- If $z_{j+2} > z_{j+1}$ switch LP_{j+2} with LP_{j+1}, in the list of LPs.

Otherwise (that is, there is only a child problem), solve the LP_j^k and if $z_j + \epsilon_z \geq zu$, this problem is removed from the tree, else if z_j is an integer solution, go to step 8.

Go to step 5.

Step 8 – A NLP subproblem is solved fixing the binary variables y of the solution of the LP problem for the level of the node where the LP problem is located. If $z_{NLP} < zu$, makes $zu = z_{NLP}$. If $z_{NLP} \geq zu$, then the previous upper bound is kept.

Step 9 – The solution of the NLP is used to generate additional constraints, that is, after a NLP subproblem is solved, this solution is added as constraint for the solution of the next LP subproblem, and so forth. These additional constraints can be made in several ways as it is commented afterwards. All the nodes of LP_j which $z_j \geq zu$ are removed from the tree and go to step 5.

The proposed algorithm was implemented in MATLAB for the versions 4.2 or superior. For the solution of the NLP subproblems the sequential quadratic programming (SQP) was used with the BFGS formula to update the estimate of the Hessian matrix ([3], [7], [11], [21]). For the solution of the problems of linear programming the simplex algorithm was used [4].

4 Alternatives of approximate constraints

The additional constraints, commented in the step 9 of the algorithm, based on the solution of the NLP subproblems, can be added for the open nodes in the search tree in several ways, such as outer approximation (OA) [6] and generalized Benders decomposition (GBD) [10].

The outer approximation has the advantage of providing tighter representation of the feasible region. However, it has the limitation that the number of columns of the LPs problems resolved for the nodes may become very large. Also in many cases, a new linearization does not necessarily result into a new approximation of the nonlinear feasible region. To avoid this problem, the Benders cut planes may be used, but in general, they do not provide strong cuts.

The approximate constraints proposed by [19], and discussed in the next section, has as basic idea to join the linearizations of nonlinear functions, keeping the linear constraints in order to strengthen the cuts.

4.1 *Approximate constraints proposed by [19]*

From the MINLP original problem (1), consider the partition of the continuous variables into the subsets of linear variables, w, and nonlinear variables, v, so that the constraints are divided into linear and nonlinear constraints.

$$
\begin{aligned}
& z = \min \; c^T + a^T w + r(v) \\
\text{subject to} \quad & \\
& Cy + Dw + t(v) < 0 \qquad\qquad (4) \\
& Ey + Fw + Gv \leq b \\
& y \in Y, \quad w \in W, \quad v \in V
\end{aligned}
$$

where

$$
f(x) = a^T w + r(v) \,, \quad g(x) = [Dw + t(v) \;\; Fw + Gv]^T \,,
$$
$$
B = [C \;\; E]^T \quad \text{and} \quad X = W \times V \,.
$$

The problem (4) is reformulated by the addition of two continuous variables (α, β) to represent the linear and nonlinear parts of the objective function. After the realization of the outer approximation in (4) at the point (w^k, v^k) generated by the k-NLP subproblem, considering the Kuhn-Tucker [9] conditions of the k-NLP subproblem in (4) for the nonlinear variables v, and after mathematical simplifications, the following MILP is obtained:

$$
\begin{aligned}
& z = \min \; \alpha \\
\text{subject to} \quad & \\
& \beta \geq r(v^k) + (\lambda^k)^T (Cy + Dw + t(v^k)) \\
& \quad - (\mu^k)^T G(v - v^k) \quad k = 1, 2, \ldots K_{NLPs} \qquad\qquad (5) \\
& Ey + Fw + Gv \leq b \\
& c^T y + a^T w + \beta - \alpha = 0 \\
& x \in X, \quad y \in Y, \quad \alpha \in R
\end{aligned}
$$

Observe that using the linear approximations above only the first inequality is modified for the open nodes in the search tree when an integer solution of a LP is obtained.

4.2 Proposed approximate constraints

Considering the original MINLP problem modeled as

$$z = \min_{x,y}\ c^T y + f(x) + \mu u$$

subject to

$$\begin{aligned} & By + g(x) \leq u \\ & x \in X = \{x | x \in R^n, x^l \leq x \leq x^u\} \\ & y \in Y = \{y | y \in \{0,1\}^m, Ay \leq a\} \end{aligned} \tag{6}$$

where $u \geq 0$. If $u > 0$ then the corresponding constraint was violated, leaving to an unfeasible solution, and the term μu is a penalization introduced in the objective function.

Making an outer approximation in (6) for the point x_0, which was generated by the first NLP subproblem, the following MILP is obtained:

$$z = \min\ \alpha$$

subject to

$$\begin{aligned} & \alpha \geq c^T y + f(x^0) + \nabla f(x^0)^T (x - x^0) + \mu u \\ & By + g(x^0) + g(x^0)^T (x - x^0) \leq u \\ & x \in X, \quad y \in Y, \quad \alpha \in R \end{aligned} \tag{7}$$

From the relaxation of the binary variables in (7), j-LPs are solved until finding an integer solution. When this solution is found, a k-NLP subproblem is solved and, based on this solution, a new inequality constraint is added to all the open nodes in the search tree. If there is no constraint violation, then the outer approximation is applied, else the Benders cut plane is used.

Thus, the resulting MILP will have the following form:

$$z = \min\ \alpha$$

subject to

$$\begin{aligned} & \alpha \geq c^T y + f(x^0) + \nabla f(x^0)^T (x - x^0) + \mu u \\ & By + g(x^0) + g(x^0)^T (x - x^0) < u \\ & \alpha \geq c^T y + f(x^k) + \nabla f(x^k)^T (x - x^k) + \mu u \\ & \quad k = 1, 2, \ldots K_{feasibleNLPs} \\ & \alpha \geq c^T y + f(x^k) + (\lambda^k)^T (By + g(x^k) - u) + \mu u \\ & \quad k = 1, 2, \ldots K_{unfeasibleNLPs} \\ & x \in X, \quad y \in Y, \quad \alpha \in R, \quad u \in R \end{aligned} \tag{8}$$

where $k_{feasibleNLPs}$ are all feasible points obtained by NLP, and $k_{unfeasibleNLPs}$ are all unfeasible points obtained by NLP.

The advantage of using the OA for feasible NLPs (when there is no constraint violation) is that the feasible region is reduced, by adding the new linearization, since the cuts are stronger. On the other hand, due to the fact that GBD provides weaker cuts, it is used for problems where there is constraint violation.

5 Results

The obtained results, with the same accuracy, same initials guesses for the continuous and binary variables, and using relaxed initial NLP, of the proposed algorithm and GAMS (http://www.gams.com) were compared. The GAMS is a system of algebraic modeling of high level language for problems of mathematical programming. It is especially designed for modeling and solving mixed-integer linear and nonlinear problems. To solve the MINLP problems below using GAMS it was chosen the DICOPT solver [23], based on the outer approximation algorithm with equality constraint relaxation and augmented penalty function (OA/ER/AP), and the CONOPT solver [24], a generalized reduced gradient algorithm with feasible path approach to solve NLP problems.

The algorithm proposed by [19], (named here Modified GBD, for short), was implemented in this work with the Benders cut planes without the partition of the continuous variables into linear and nonlinear subsets. The partition was not implemented because it is not straightforward to automate for general MINLP problems, and it could also be done in the proposed algorithm. Moreover, as the proposed algorithm has the same scaling properties of the Modified GBD, the problem size is not a necessary parameter to compare these two algorithms. This kind of performance would be appropriate when comparing these algorithms with the software GAMS. However, as GAMS has a higher overhead to build the optimization problem, due to its nice user interface, the comparison would not be fair.

When comparing the Modified GBD with the proposed algorithm, the CPU time is directly related to the number of LPs and NLPs to be solved, because the number of operations to solve each LP and NLP is the same for both algorithms.

For the software GAMS the comments for the CPU time are similar to the problem size. Therefore, only the number of LPs and NLPs were taken into account in the comparisons.

5.1 Convex example 1

Proposed by [9], where the MINLP is modeled as:

$$
\begin{aligned}
& z = \min_{x,y}\ y_1 + y_2 + y_3 + 5x^2 \\
\text{subject to}\quad & \\
& 3x - y_1 - y_2 < 0 \\
& -x + 0.1y_2 + 0.25y_3 \leq 0 \\
& y_1 + y_2 + y_3 \geq 2 \\
& y_1 + y_2 + 2(y_3 - 1) \geq 0 \\
& 0.2 \leq x \leq 1, \quad y \in \{0, 1\}^3
\end{aligned}
\tag{9}
$$

The optimal solution of (9) is found for

$$z = 2.2 \quad \text{with} \quad x = 0.2 \quad \text{and} \quad y = (1, 1, 0)\,.$$

Table 1 shows the statistics of the proposed algorithm and GAMS related to the number of NLPs and LPs.

x_0	y_0	Proposed Algorithm		GAMS	
		NLPs	LPs	NLPs	LPs
0.3	(1,0,0)	2	2	3	7
	(1,1,1)	2	2	3	7
	(1,0,1)	2	4	3	7
	(0,0,0)	2	2	3	7
	(0,1,0)	1	1	3	7
	(0,1,1)	1	1	3	7
	(0,0,1)	2	4	1	0

Table 1. Comparison of proposed algorithm and GAMS, convex example 1.

5.2 *Convex example 2*

MINLP problem given by [9] modeled as:

$$
\begin{aligned}
& z = \min_{x,y} \; -y + 4e^{-x} + x \\
\text{subject to} \quad & \\
& -2e^{-x} + x + y \le 0 \\
& 0.5 \le x \le 1.4, \quad y \in \{0, 1\}
\end{aligned}
\tag{10}
$$

The algorithm finds the optimal solution of the problem (10) for $x = 0.853$ and $y = 0$, being $z = 2.558$. Table 2 shows the comparison of proposed algorithm and GAMS.

x_0	y_0	Proposed Algorithm		GAMS	
		NLPs	LPs	NLPs	LPs
	0	2	4	3	4
0.6	1	2	4	3	4

Table 2. Comparision of proposed algorithm and GAMS, convex example 2.

5.3 *Convex example 3*

Considering the problem proposed by [14] defining the best configuration for the given processes according to Figure 1.

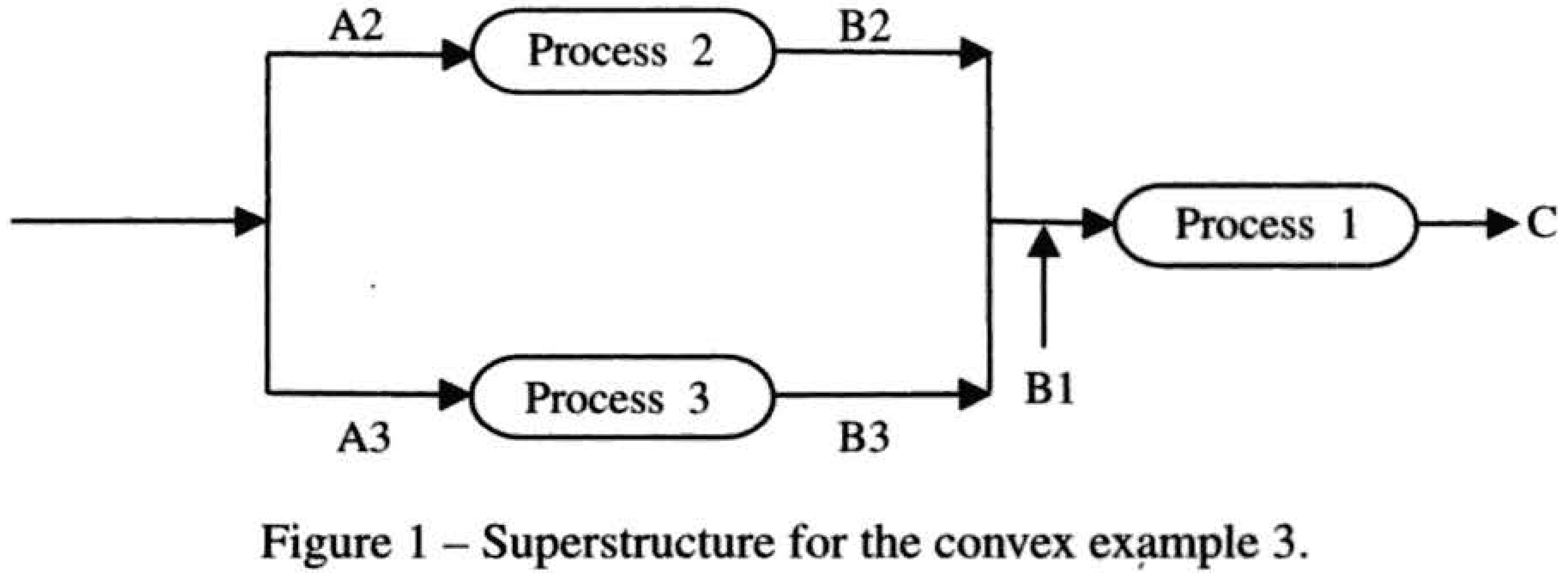

Figure 1 – Superstructure for the convex example 3.

The formulation of the MINLP model is given by:

$$
\begin{aligned}
z = \min \quad & -[11C - 7B1 - B2 - 1.2B3 \\
& \quad + 1.8(A2 + A3) - 3.5y_1 - y_2 - 1.5y_3]
\end{aligned}
$$

subject to

$$
\begin{aligned}
& C = 0.9B \\
& B2 = \log(1 + A2) \\
& B3 = 1.2\log(1 + A3) \\
& B = B1 + B2 + B3 \\
& C \leq y_1 \\
& B2 \leq 10y_2 \\
& B3 \leq 10y_3 \\
& y_2 + y_3 < 1 \\
& y_1, y_2, y_3 \in \{0, 1\} \\
& C, B1, B2, B3, A2, A3 > 0
\end{aligned}
\tag{11}
$$

The variables y_1, y_2, and y_3 define the existence or not of the processes 1, 2, and 3, respectively.

For implementation, the equality constraints of the problem (11) were eliminated, resulting in the following continuous variables: $A2 = x1$, $A3 = x2$, and $B1 = x3$. Thus, the problem (11) can be rewritten in the following way:

$$
\begin{aligned}
z = \min_{x,y} \quad & -2.9x_3 - 8.9\log(1 + x_1) \\
& - 10.44\log(1 + x_2) + 1.8x_1 + 1.8x_2 \\
& + 3.5y_1 + y_2 + 1.5y_3
\end{aligned}
$$

subject to

$$
\begin{aligned}
& -y_1 + 0.9\log(1 + x_1) + 1.08\log(1 + x_2) + 0.9x_3 \leq 0 \\
& -10y_2 + \log(1 + x_1) \leq 0 \\
& -10y_3 + 1.2\log(1 + x_2) < 0 \\
& y_2 + y_3 - 1 \leq 0
\end{aligned}
\tag{12}
$$

This example, using the initial values for the binary variables $y_0 = (0, 1, 0)$ and for continuous variables $x_0 = (0, 0, 1)$, a search is conducted by evaluating nodes 1 and 2, according to Figure 2. The initial NLP results in an upper bound, $zu = 1.0$, and the first node results in a lower bound, $zl = -4.333$, for the optimal solution of the MINLP.

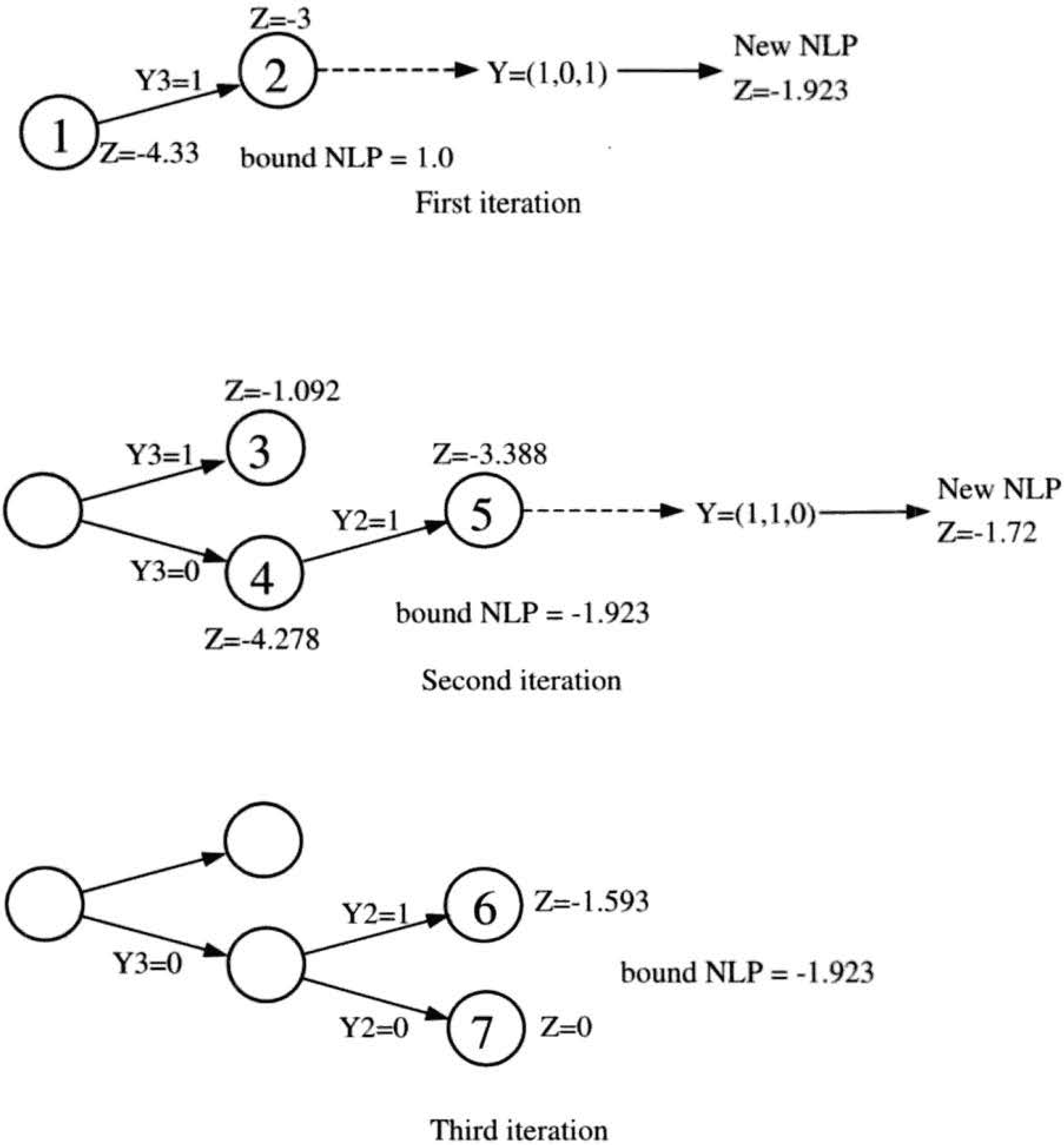

Figure 2 – Branch and bound search of the master problem MILP, convex example 3.

Node 2 yields the integer solution $y = (1, 0, 1)$ with a lower bound $zl = -3$ (see Figure 2). At this configuration, a second NLP subproblem is solved and it yields an upper bound $zu = -1.923$.

The upper bound, to prune nodes in the tree, is provided by the solutions of the NLPs subproblems. Then feasible nodes that are below to this bound are kept opened. With the resolution of the second NLP subproblem, new linearizations are added to these open nodes, tightening the linear representation of the feasible region. As the node 2 has not been branched, it is updated by adding the new approximate constraints (see node 3 in Figure 2). This node can be pruned because the solution is $zl = -1.092$, being this lower bound larger than the current upper bound. Going back to the node 1, which was already branched, then it is not modified, and a new node, 4, is created. A new integer solution is found in the sequence, $y = (1, 1, 0)$ and the corresponding NLP subproblem is solved giving a higher upper bound, $zu = -1.72$. Consequently, $zu = -1.923$ is a better upper bound.

Again, new outer approximation are added to the open nodes 6 and 7, but the solutions of the respective LPs exceed the current upper bound (see Figure 2). Consequently the search at this point can be finished, confirming that $z = -1.923$ is the optimal solution of the MINLP, being $x = (0, 1.524, 0)$ and $y = (1, 0, 1)$. Note that with the modified GBD and the proposed method, the search finishes after examining 7 nodes, where 3 NLP subproblems are resolved.

Table 3 shows the results of the number of LPs for each NLP solved in the implementation of the standard GBD, the algorithm proposed by [19] (named Modified GBD), and the proposed algorithm in this work.

NLP	Number of LPs		
	Standard GBD	Modified GBD	Proposed Algorithm
1	5	2	2
2	3	3	3
3	5	2	2

Table 3. Number of LPs and NLPs for the convex example 3.

Using the same example, a test was taken for initial guesses of binary and continuous variables, $y_0 = (1, 0, 1)$ and $x_0 = (0, 0, 1)$, respectively. The proposed algorithm finds the optimal solution after examining 5 nodes in the branch and bound search, being solved 2 NLP.

Table 4 shows the comparison of this method with the standard GBD and the modified GBD.

Table 5 shows the results for LPs and NLPs of the proposed algorithm and GAMS for several initial guess for the binary variables.

In spite of the algorithm be developed for convex examples, it was verified its behavior for the following nonconvex examples.

NLP	Number of LPs		
	Standard GBD	Modified GBD	Proposed Algorithm
1	4	4	4
2	1	1	1
3	1	1	–
4	1	2	–
5	2	–	–

Table 4. Number of LPs and NLPs for the convex example 3 with another initial guess.

x_0	y_0	Proposed Algorithm		GAMS	
		NLPs	LPs	NLPs	LPs
	(0,1,0)	3	6	3	28
	(1,0,1)	3	6	3	27
	(1,1,1)	3	6	3	28
(0,0,0)	(0,0,0)	3	6	3	27
	(1,0,0)	3	6	3	28
	(0,1,1)	3	6	3	27
	(1,1,0)	3	6	3	27
	(0,0,1)	3	6	3	27

Table 5. Comparision of proposed algorithm and GAMS, convex example 3.

5.4 Nonconvex example 1

Considering the problem proposed by [15], modeled as:

$$
\begin{aligned}
& z = \min_{x,y}\ 2x + y \\
\text{subject to} \quad & \\
& 1.25 - x^2 - y < 0 \\
& x + y \leq 1.6 \\
& 0 \leq x \leq 1.6, \quad y \in \{0, 1\}
\end{aligned}
\tag{13}
$$

where the nonlinear inequality constraint contains a nonconvex term for the continuous variable x.

The global optimum of the problem is located at $x = 0.5$ and $y = 1$, where the value of the objective function is $z = 2$. Table 6 shows the comparison of proposed algorithm and GAMS.

x_0	y_0	Proposed Algorithm		GAMS	
		NLPs	LPs	NLPs	LPs
	0	1	1	1	0
0.1	1	1	1	1	0

Table 6. Comparision of proposed algorithm and GAMS, nonconvex example 1.

5.5 *Nonconvex example 2*

The example of the MINLP problem proposed by [23] is modeled as:

$$z = \min_{x,y} \quad -0.7y + 5(x-0.5)^2 + 0.8$$

subject to

$$\begin{aligned} &\exp(x-0.2) + 1.1y + 1 \leq 0 \\ &x - 1.2y - 0.2 \leq 0 \\ &0 \leq x \leq 1, \quad y \in \{0, 1\} \end{aligned} \tag{14}$$

The optimal solution of the problem (14) is given by $z = 1.076$, where $x = 0.942$ and y = 1. Table 7 shows the number of LPs and NLPs, where it can be observed for the initial guess of the binary variable $y_0 = 0$, the GAMS did not find a solution.

x_0	y_0	Proposed Algorithm		GAMS	
		NLPs	LPs	NLPs	LPs
	0	2	3	–	–
0.1	1	2	3	3	3

Table 7. Comparision of proposed algorithm and GAMS, nonconvex example 2.

5.6 *Nonconvex example 3*

Figure 3 represents a superstructure for a selection problem among two candidate reactors to minimize the production cost of a desired product, proposed by [15].

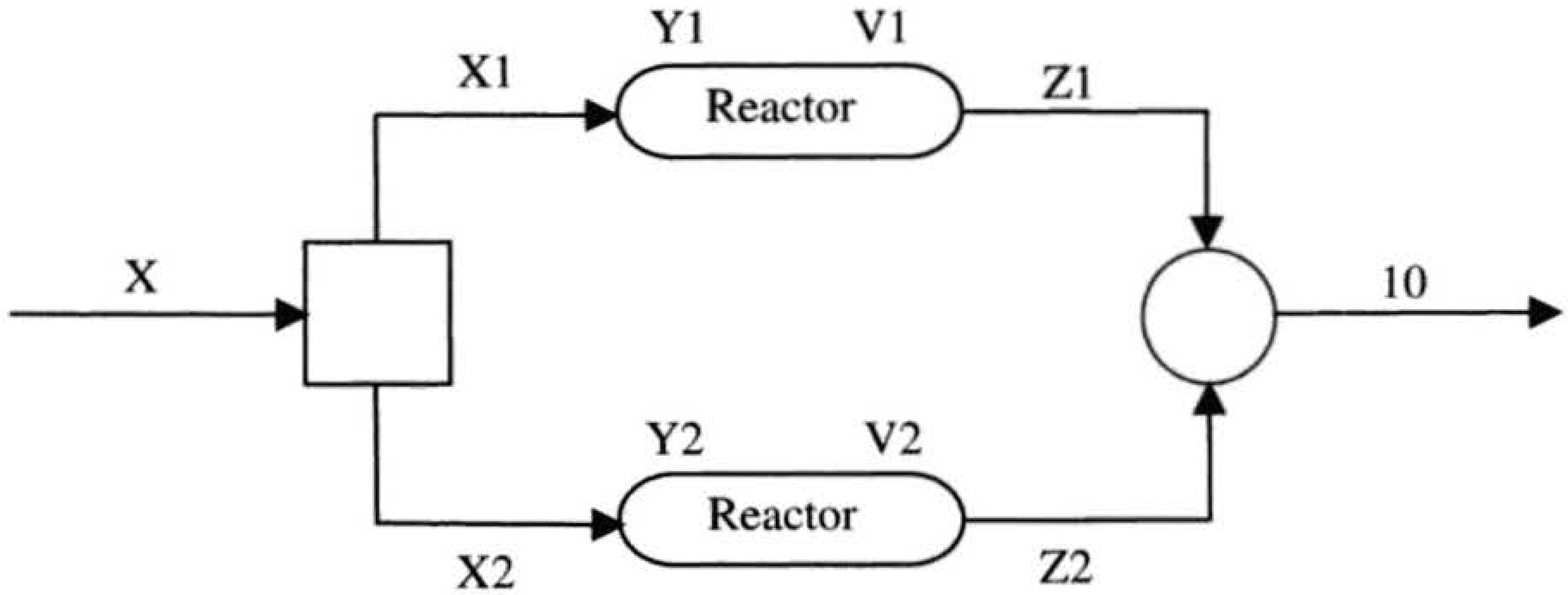

Figure 3 – Superstructure for the nonconvex example 3.

The formulation of the MINLP of this problem is given as:

$$cost = \min\ 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

subject to

$$\begin{aligned}
&z_1 = 0.9(1 - \exp(-0.5v_1))x_1\\
&z_2 = 0.8(1 - \exp(-0.4v_2))x_2\\
&x_1 + x_2 - x = 0\\
&z_1 + z_2 = 10\\
&v_1 \le 10y_1\\
&v_2 \le 10y_2\\
&x_1 \le 20y_1\\
&x_2 \le 20y_2\\
&y_1 + y_2 \ge 1\\
&x_1, x_2, z_1, z_2, v_1, v_2 > 0, \quad y_1, y_2 \in \{0, 1\}^2
\end{aligned} \tag{15}$$

The binary variables y_1 and y_2 denote the existence or nonexistence of the reactors 1 and 2, respectively. In the objective function, the values 7.5 and 5.5 represent the capacity of the reactors 1 and 2, respectively; v_1 represent the volume of the reactor 1, v_2, the volume of the reactor 2, and x is the amount of the raw material. The two nonlinear equations are input-output relations for the reactors which define the output flows of exit z_1 and z_2 in terms of the input flows x_1 and x_2 and the reactors volumes. The raw material x is split into the reactor input flows x_1 and x_2, and the total demand of 10 units should be met by the output flows. The next four inequalities are logical constraints which insure that if a given reactor does not exist (for example $y_1 = 0$), then the corresponding volume and feed stream are zero. The last constraint requires that either reactor 1 or 2 be selected.

Due to implementation reasons, the equality constraints were transformed in inequality constraints, then the MINLP is rewritten as:

$$cost = \min_{x,v,y}\ 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5(x_1 + x_2)$$

subject to

$$\begin{aligned}
&z_1 = 0.9(1 - \exp(-0.5v_1))x_1 \\
&z_2 = 0.8(1 - \exp(-0.4v_2))x_2 \\
&z_1 + z_2 \geq 10 \\
&z_1 + z_2 \leq 10 \\
&v_1 \leq 10y_1 \\
&v_2 \leq 10y_2 \\
&x_1 \leq 20y_1 \\
&x_2 \leq 20y_2 \\
&y_1 + y_2 \geq 1 \\
&x_1, x_2, z_1, z_2, v_1, v_2 > 0, \quad y_1, y_2 \in \{0, 1\}^2
\end{aligned} \tag{16}$$

Table 8 shows the number of LPs and NLPs of the problem for the proposed algorithm and GAMS, where $x_0 = (x_1, x_2, v_1, v_2)$. The proposed algorithm always finds the optimal solution of $cost = 99.24$ for $x = (13.428, 0)$ and $v = (3.514, 0)$. In case of GAMS, for initial guess of $y_0 = (0, 1)$, it solves a NLP subproblem and no LP problem but finds a sub-optimal solution of

$$cost = 107.376\,, \quad \text{where} \quad x = (0, 15) \quad \text{and} \quad v = (0, 4.479)\,.$$

x_0	y_0	Proposed Algorithm		GAMS	
		NLPs	LPs	NLPs	LPs
	(1,0)	1	3	3	10
(1,1,0,0)	(0,1)	1	3	1	0

Table 8. Comparision of proposed algorithm and GAMS, nonconvex example 3.

6 Conclusions

This work presented a branch and bound method for convex MINLP problems that is based on the solution of LPs problems and NLPs subproblems. The method avoids the solution of sequential NLPs subproblems and MILP master problems that is demanded in the implementation of standard GBD and OA algorithms. The obtained results of the convex problems and nonconvex tests showed that the algorithm is efficient, reducing the number of nodes that need to be examined when compared with the standard GBD and, at least, obtaining the same results when compared with the modified GBD. When compared with GAMS, satisfactory results were also obtained in favor of the proposed algorithm, for convex and nonconvex examples, in spite of the algorithm be designed for convex problems.

REFERENCES

[1] Benders, J.F., it Partitioning procedures for solving mixed-variables programming problems, Numer. Math., **4** (1962), pp. 238-252.

[2] Boland, D. and Hindmarsh, E., *Heat exchanger network improvements*, Chem. Engng. Prog., **80** (1984), pp. 47-54.

[3] Broyden, C.G., *The convergence of a class of double-rank minimization algorithms 2: the new algorithm*, Journal Institute of Math and its Appl., **6** (1970), pp. 222-231.

[4] Dantzig, G.B., *Maximization of a linear function of variables subject to linear inequalities. Activity analysis of production an allocation*, eds: Tj. C. Koopmans, pp. 339-347, Wiley, New York, (1951).

[5] Doldan, O.B., Bagajewicz, M.J. and Cerda, J., *Designing heat exchanger networks for existing chemical plants*, Comput. Chem. Engng., **9** (1985), pp. 483-498.

[6] Duran, M.A. and Grossmann, I.E., *An outer approximation algorithm for a class of mixed-integer nonlinear programs*, Math. Prog., **36** (1986), pp. 307-339.

[7] Fletcher, R., *A new approach to variable metric algorithm*, The Computer Journal, **13**, 3 (1970), pp. 317-322.

[8] Fletcher, R. and Leyffer, S., *Solving mixed integer nonlinear programs by outer approximation*, Math. Progr., **66**, 3 (1994), pp. 327-349.

[9] Floudas, C.A., *Nonlinear and Mixed-Integer Optimization*, Oxford University, New York, (1995).

[10] Geoffrion, A.M., *Generalized Benders Decomposition*, Journal Optimization Theory Aplic., **10**, 4 (1972), pp. 237-260.

[11] Goldfarb, D., *A family of variable metric methods derived by variational means*, Mathematics of Computation, **24** (1970), pp. 23-26.

[12] Gundersen, T. and Naess, L., *The synthesis of cost optimal heat exchanger network synthesis*, Comp. Chem. Engng., **12**, 4 (1988), pp. 503-530.

[13] Holmberg, K., *On the convergence of the cross decomposition*, Math. Progr., **47**, 2 (1990), pp. 269-296.

[14] Kocis, G.R. and Grossmann, I.E., *Relaxation strategy for the structural optimization of process flowsheets*, Ind. Engng. Chem. Res., **26**, 9 (1987), pp. 1869-1880.

[15] Kocis, G.R. and Grossmann, I.E., *Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis*, Ind. Engng. Chem. Res., **27** (1988), pp. 1407-1421.

[16] Kumar, A. and Lucia, A., *Distillation optimization*, Comp. Chem. Eng., **12** (1988), pp. 1263-1266.

[17] Land, A.H. and Doig, A.G., *An automatic method for solving discrete programming problems*, Econometrica, **28** (1960), p. 497.

[18] Mahalec, V. and Motard, R.L., *Evolutionary search for na optimal limiting process flowsheet*, Comput. Chem. Engng., **1** (1977), pp. 149-160.

[19] Quesada, I. and Grossmann, I.E., *An LP/NLP based branch and bound algorithm for convex MINLP optimization problems*, Computers Chem. Engng., **16**, 10/11 (1992), pp. 937-947.

[20] Saboo, A.K., Morari, M. and Golberg, R.D., *RESHEX: An interactive software package for the synthesis and analysis of resilient heat-exchanger networks*, Comput Chem. Eng., **10**, 6 (1986), pp. 577-600.

[21] Shanno, D.F., *Conditioning of quasi-Newton methods for function minimization*, Mathematics of Computation, **24** (1970), pp. 641-656.

[22] Siirola, J.J., *Strategies in process synthesis: Advances in the hierarchical approach*, Comp. Chem. Engng., **20** (1996), suppl. S1637-S1643.

[23] Viswanathan, J. and Grossmann, I.E., *A combined penalty function and outer approximation method for MINLP optimization*, Computers Chem. Engng., **14**, 7 (1990), pp. 769-782.

[24] Drud, A.S., *CONOPT – A large-scale GRG code*, ORSA Journal on Computing, **6** (1992), pp. 207-216.