Constraint Logic Programming for Non Convex NLP and MINLP Problems

Kotecha P. R., Gudi R. D.[*]

Department of Chemical Engineering
Indian Institute of Technology Bombay, Powai, Mumbai – 400076, India.
* Author for correspondence: Email: ravigudi@che.iitb.ac.in

Abstract: This paper presents an algorithm to solve non-convex NLP and MINLP problems using CLP. In the proposed technique, the continuous variables are relaxed to take only integer values contained in the real domain of the variable. The merits of the CLP algorithm, *viz* powerful CP strategies are proposed to be exploited to get integer solutions to the relaxations. A lower bound to the objective function is obtained if the relaxed problem is feasible. This information is used in the successive stages wherein the continuous variables are corrected from their integer variable representation to obtain real solutions with desired accuracy. The proposed technique has been successfully demonstrated on two MILP, two non convex NLP and two non convex MINLP problems. The problems were also solved by traditional techniques and the superiority of the proposed method has been demonstrated.

Keywords: Constraint Logic Programming (CLP), Constraint Propagation (CP), Integer Programming (IP) Mixed Integer Linear Programming (MILP), Mixed Integer Non-Linear Programming (MINLP)

1. INTRODUCTION

Many of the problems arising in synthesis and design, and in planning and scheduling problems are MINLP models. This is due to the fact that MINLP provides much greater modeling flexibility for tackling a large variety of problems. While MILP methods have been largely developed outside process systems engineering, chemical engineers have played a prominent role in the development of MINLP methods. Some of the methods used to solve MINLP include Branch and Bound, Generalized Benders Decomposition (GBD), Outer Approximation (OA) and Extended Cutting Plane methods (ECP). In the above methods, the objective function and the constraints are assumed to be convex and differentiable. But, often times, some problems lead to formulations that do not satisfy these requirements of convexities. The trim loss problem in paper industry and the 10P3S problem (Munawar & Gudi, 2005) are some such problems. Ryoo and Sahinidis (1995) have reported a collection of twenty one non convex problems that arise in process synthesis.

There have been a number of attempts to handle non convex MINLPs. Tawarmalani and Sahinidis (2000) have developed a branch and bound method that branches on the continuous and discrete variables. This method, which relies on bounds reduction using underestimators, has been implemented in BARON. The SMIN-αBB and GMIN-αBB algorithms have been developed for twice-differentiable nonconvex MINLPs. The αBB method, which is a branch and bound procedure that branches on both the continuous and discrete variables according to specific options, was developed by using a valid convex underestimation of general functions as well as special functions. The branch-and-contract method 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 algorithm at each node of the tree for the spatial search. Lee and Grossmann (2001) developed a two level-branch and bound method for solving nonconvex disjunctive programming problems. Munawar & Gudi (2005) have proposed a hybrid technique to solve MINLP that makes use of Differential Evolution (DE) and Non Linear Programming (NLP).

Finding its roots in computer science and artificial intelligence communities, Constraint Programming (Puget, 1994; Van Hentenryck, 1989) is an alternative approach to discrete and continuous problem solving. For decades, it has proved successful in several applications, particularly in scheduling and logistics. Unlike mathematical programming, it does not use relaxations but it relies on methods of logical inference (primarily domain reduction and constraint-propagation) to reduce the domain of possible values for a discrete or continuous variable. The rich modeling language of CLP has contributed in a large way towards its success.

The methods described above for the solution of non convex MINLP essentially involve the Branch and Bound and Extended Cutting Plane methods. The successes of these methods are crucially dependent on the successful solution of the NLP sub-problems at each node. On the other hand, CLP methods are good at domain propagation but are restricted to the finite domain and do not handle continuous variables. In recent years, there has been substantial progress in the development of powerful constraint propagation engines, which could be exploited towards solution of problems represented in the finite domain. An alternative approach to solving MINLPs that relies on the use of CLP towards domain reduction could therefore be examined towards solution of such non-convex MINLP problems.

In this paper, a method has been proposed that uses CLP to solve non-convex MINLP problems. Contrary to the regular approach of relaxing on the integrality requirements as in the branch and bound algorithm, the proposed approach relaxes the continuous variables to discrete values. Since constraint propagation approaches are usually more suitable in the finite domain over other integer programming methods (Smith, *et al.,* 1997), we propose to use the powerful features of constraint propagation engine to reduce the finite domain space. The method involves solving a master problem obtained by relaxing the space of continuous variables to integer domain. If the relaxed problem is feasible, it ensures that a lower bound (for the maximization problem) is obtained for the problem. Also, the domain reduction inherently present in CP helps to specify tighter bounds on the continuous variables. These steps are followed by the solution of another sub problem in which the continuous variables are corrected from their integer variable representation to obtain real solutions with desired accuracy. In this sub problem, the bounds on the continuous variables are also tightened by inferring from the solution of the master problem. If the master problem is infeasible, the original problem itself is discretized and solved.

The remainder of the paper is organized as follows. The following section gives a review of CLP technique. Section 3 focuses on some theoretical aspects of CP relevant to the real domain. Section 4 discusses the results obtained by applying the proposed methodology on two MILP and two non convex NLP and MINLP problems. The second MILP problem is a planning and scheduling problem on a set of dissimilar parallel machines (Jain and Grossman, 2001). We solve this problem using CLP even when the start times, due dates, release dates and processing times are not integer parameters. This is particularly noteworthy considering the fact that ILOG Scheduler does not support continuous variables.

2. CONSTRAINT LOGIC PROGRAMMING

Constraint Programming (Hentenryck, 1989; Hooker, 2000) was originally developed to solve feasibility problems, but it has been extended to solve optimization problems as well. In finite domain CLP, each integer variable x_i has an associated domain D_i which is the set of possible values that this variable can take on in the optimal solution. The cartesian product of the domains D_1 x…..x D_n forms the solution space of the problem. This space is finite and can be searched exhaustively for a feasible or optimal solution, but to intelligently enumerate this search, CP is used to infer infeasible solutions and prune the corresponding domains. From this viewpoint, CP operates on, and narrows down, the set of possible solutions.

Constraint Programming is based on performing a tree enumeration. At each node the domains of the variables, which can be continuous, general integer, boolean and binary are reduced. If an empty domain is found the domain is pruned. Branching is performed whenever a domain of an integer, binary or boolean variable has more than one element, or when the bounds of the domain of a continuous variable do not lie within a tolerance. Whenever a solution is found, or a domain of a variable is reduced, new constraints are added to ensure that the node is not revisited. The search terminates when no further nodes need to be examined.

Traditional IP methods are very efficient for problems with good relaxations but suffer when the relaxation is weak or when its restricted modeling (linear inequalities) framework results in large models. CLP with its more expressive language results in smaller models that are closer to the problem description, and performs better for highly constrained problems; however, it lacks the global perspective of relaxations.

There have been a few attempts to integrate CLP and MILP so that their complementary strengths can be exploited. Some examples include the modified generalized assignment problem (Darby *et al.,* 1997), the template design problem, the progressive party problem (Smith *et al.,* 1997), and the change problem (Heipcke, 1999). These papers showed that MILP is very efficient when the relaxation is tight and the models have a structure that can be effectively exploited. CP works better for highly constrained discrete optimization problems where expressiveness of MILP is a major limitation. Hooker (2000) deals with the subject of MILP and CP integration in detail. Jain and Grossman (2001) have shown a decomposition method wherein a master MILP and a CLP subproblem work in cooperation and are able to address problems, that were otherwise intractable by both the methods.

3. DECOMPOSITION METHODOLOGY

The algorithms proposed in this section have been motivated by the high efficiency of CLP in reducing the domain of variables. A motivating example has also been presented at the end of the section. Although shown for an MINLP, this theory is also valid for solution of an MILP.

Consider a problem, which when modeled as an MINLP has the following structure,

$$\text{(M1)} \quad \max_{x,y} f(x,y) \tag{1}$$

$$\text{s.t} \quad G(\bar{x},y) \tag{2}$$

$$x \in \square \tag{3}$$

$$y \in \{0,1\} \tag{4}$$

where *G(x,y)* could represent both equality and inequality constraints. The above optimization problem has both continuous and binary variables. It is to be noted no restrictions are placed *on (1) and (2)* to be convex and linear in the discrete variables.

The master problem of (M1) is given by

$$\text{(M2)} \quad \max_{x,y} f(x,y) \tag{5}$$

$$\text{s.t} \quad G(\bar{x},y) \tag{6}$$

$$x \in \square \tag{7}$$

$$y \in \{0,1\} \tag{8}$$

Any solution obtained for M2 will provide a lower bound for the problem M1. In other words, M1 will never have a value that is lower than M2. Let f', the solution to M2, define this lower bound.

The sub problem (M3) is defined as

$$\text{(M3)} \quad \max_{\bar{x},y} f(\bar{x},y) \tag{9}$$

$$\text{s.t} \quad G(\bar{x},y) \tag{10}$$

$$f \geq f' \tag{11}$$

$$\bar{x} = \sum_{i=0}^{n} 10^{-i}(a_i\_x) \tag{12}$$

$$y \in \{0,1\} \tag{13}$$

$$a_0\_x \in \bar{\square} \tag{14}$$

$$a_i\_x \in \{0,1,2,3....9\} \qquad i \neq 0 \tag{15}$$

The domain $\bar{\square}$ can be inferred from the solution of M2. For a maximization (minimization) problem, the lower (upper) bound of x will start from (end at) the solution of x in M2. The upper (lower) bound will essentially remain the same as in M2.

The sub-problem (M4) is defined as

$$\text{(M4)} \quad \max_{\bar{x},y} f(\bar{x},y) \tag{16}$$

$$\text{s.t} \quad G(\bar{x},y) \tag{17}$$

$$\bar{x} = \sum_{i=0}^{n} 10^{-i}(a_i\_x) \tag{18}$$

$$y \in \{0,1\} \tag{19}$$

$$a_0\_x \in \tilde{\square} \tag{20}$$

$$a_i\_x \in \{0,1,2,3....9\} \qquad i \neq 0 \tag{21}$$

This problem is solved if and only if M1 gives an infeasible solution. The domain in equation (20) is larger than (14) because no inference can be made from the solution of M1 and M4 is a larger problem than M3. It should be noted that M4 can be solved for an optimal solution even without solving M2.

Table 1: Proposed Algorithm to solve MILP/MINLP problems

```
Algorithm

Step 1: Formulate M2 by relaxing the continuous
variable space of M1 to Integer space
Step 2: Solve M2 using CLP.
        If M2 is feasible,
                go to Step 5
Step 3: Formulate M4 by discretizing M1 using (17)
subject to (19) and (20)
Step 4: Solve M4.
        If feasible,
                Solution is Optimal for M1
        else
                M1 is Infeasible
Step 5: Formulate M3 by discretizing M2 using (11)
subject to (13) and (14)
Step 6: Solve M3 to obtain optimal of M1
```

4. CASE STUDIES

This section discusses the application of the proposed methodology on some MILP, NLP and non-convex MINLP problems.

4.1 Case Study 1

The example discussed is an MILP with 3 continuous and 3 binary variables and will be hereon referred as Case Study 1.

(M1)

$$\begin{aligned} &Min_{x,y} && f = x_1 + x_2 - x_3 + y_1 + y_2 - y_3 \\ &s.t && x_1 + x_2 - x_3 \leq 151.2 \\ &&& x_3 - x_1 \geq 2 \\ &&& x_2 - x_1 \geq 70 \\ &&& y_1 + y_2 + y_3 \geq 2 \\ &&& x_1 \geq 5.9 \end{aligned}$$

$$(5,0,0) \leq X \leq (100,100,100); \quad Y \in \{0,1\}^3$$

The formulation of model (M2) is done by relaxing the continuous spaces of x to integer space.

$$\begin{aligned} &&Min_{x,y} && f = x_1 + x_2 - x_3 + y_1 + y_2 - y_3 \\ &&s.t && x_1 + x_2 - x_3 \leq 151.2 \\ &&&& x_3 - x_1 \geq 2 \\ &\text{(M2)} &&& x_2 - x_1 \geq 70 \\ &&&& y_1 + y_2 + y_3 \geq 2 \\ &&&& x_1 \geq 5.9 \qquad x_2, x_3 \in \{0,1,2...,100\}; \\ &&&& x_1 \in \{5,6...,100\}; \quad Y \in \{0,1\}^3 \end{aligned}$$

The optimal solution for (M2) is $(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*, y_3^*; f^*) = (6,76,69,0,1,1;13)$.

This solution gives us a bound on the objective function i.e., the objective function of (M1) can never be greater than 13.

The model (M3) is formulated by inferring tighter bounds from the solution of (M2)

(M3)

$$
\begin{aligned}
Min_{x,y} \quad & f=\bar{x}_1+\bar{x}_2-\bar{x}_3+y_1+y_2-y_3 \\
s.t \quad & \bar{x}_1+\bar{x}_2-\bar{x}_3 \le 151.2 \\
& \bar{x}_3-\bar{x}_1 \ge 2 \\
& \bar{x}_2-\bar{x}_1 \ge 70 \\
& y_1+y_2+y_3 \ge 2 \\
& \bar{x}_1 \ge 5.9; \quad Y \in \{0,1\}^3 \\
& a_0\_x_1 \in \{5,6\}; \quad a_0\_x_2 \in \{0,1,...76\}; \\
& a_0\_x_3 \in \{0,1,..69\} \\
& \left.\begin{array}{l} a_1\_x_1, a_2\_x_1, a_1\_x_2, \\ a_2\_x_2, a_1\_x_3, a_2\_x_3 \end{array}\right\} \in \{0,1,2...9\} \\
where \quad & \bar{x}_1 = a_1\_x_1 + 0.1\, a_2\_x_1 + 0.01\, a_3\_x_1 \\
& \bar{x}_2 = a_1\_x_2 + 0.1\, a_2\_x_2 + 0.01\, a_3\_x_2 \\
& \bar{x}_3 = a_1\_x_3 + 0.1\, a_2\_x_3 + 0.01\, a_3\_x_3
\end{aligned}
$$

The optimal solution for (M3) is $(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*, y_3^*; f^*)$ is $(5.95, 75.95, 69.3, 0, 1, 1; 12.6)$

The formulation for model M4 is given by

$$
\begin{aligned}
& Min_{x,y} \quad f=\bar{x}_1+\bar{x}_2-\bar{x}_3+y_1+y_2-y_3 \\
& s.t \quad \bar{x}_1+\bar{x}_2-\bar{x}_3 \le 151.2 \\
& \qquad \bar{x}_3-\bar{x}_1 \ge 2 \\
& \qquad \bar{x}_2-\bar{x}_1 \ge 70 \\
(M4) \quad & \qquad y_1+y_2+y_3 \ge 2 \\
& \qquad \bar{x}_1 \ge 5.9; \quad Y \in \{0,1\}^3 \\
& \qquad a_0\_x_1 \in \{5,6,..,100\} \\
& \qquad a_0\_x_2 \in \{0,1,...100\} \\
& \qquad a_0\_x_3 \in \{0,1,..100\}
\end{aligned}
$$

Though the master problem was feasible, nevertheless, M4 can be formulated and solved for this case study. But as said earlier, it is more computationally expensive. This fact is substantiated by the Table 2.

Table 2: CLP Parameters for Case Study 1

Model		No. of Choice Points	No. of Failures
Method 1	M2	3	64
	M3	13	53
Method 2	M4	27	128

4.2 Case Study 2

The following planning and scheduling MILP model has been taken from Jain and Grossman (2001). The scheduling problem involves finding a least-cost schedule to process a set of orders I using a set of dissimilar parallel machines M. Processing of an order $i \in I$ can only begin after the release date r_i and must be completed at the latest by the due date d_i. Order i can be processed on any of the machines. The processing cost and the processing time of order $i \in I$ on machine $m \in M$ are C_{im} and p_{im}, respectively.

$$
\begin{aligned}
\min \quad & \sum_{i\in I}\sum_{m\in M} C_{im}x_{im} \\
\text{s.t} \quad & ts_i \ge r_i \quad \forall\, i \in I; \qquad \sum_{m\in M} x_{im} = 1 \quad \forall\, i \in I \\
& ts_i \le d_i - \sum_{m\in M} p_{im}x_{im} \quad \forall\, i \in I \\
& \sum_{i\in I} p_{im}x_{im} \le \max_i\{d_i\} - \min_i\{r_i\} \\
& y_{ii'} + y_{i'i} \ge x_{im} + x_{i'm} - 1 \quad \forall\, i,i' \in I, i' > i, m \in M \\
& ts_{i'} \ge ts_i + \sum_{m\in M} p_{im}x_{im} - U\left(1-y_{ii'}\right) \quad \forall\, i,i' \in I, i' \ne i \\
& y_{ii'} + y_{i'i} \le 1 \quad \forall\, i,i' \in I, i' > i \\
& y_{ii'} + y_{i'i} + x_{im} + x_{i'm'} \le 2 \quad \forall\, i,i' \in I, i' > i \\
& \qquad\qquad m, m' \in M, m \ne m' \\
& ts_i \ge 0; \quad x_{im} \in \{0,1\} \quad \forall\, i \in I, m \in M \\
& y_{ii'} \in \{0,1\} \quad \forall\, i \;\;, i' \in I, i' \ne i; \quad U = \sum_{i\in I} \max_{m\in M}\{p_{im}\}
\end{aligned}
$$

The main decisions involved in this scheduling problem are assignment of orders on machines, sequence of orders on each machine, and start time for all the orders. The binary variable x_{im} is an assignment variable, and it is equal to one when order i is assigned to machine m. Binary variable $y_{ii'}$ is the sequencing variable, and it is equal to one when both i and i' are assigned to the same machine and order i' is processed after order i. The continuous variable ts_i denotes the start time of order i.

Table 3: Data for Case Study – 2

Order	Cost		Processing time		Release date	Due date
	Mch 1	Mch 2	Mch 1	Mch 2		
1	10.6	6.25	10.2	14.5	2.1	16.9
2	8.26	5.45	6.25	8.10	3.5	14.2
3	12.47	7.06	11.98	16.14	4.8	25.5

Table 4: Results for Case Study – 2

Order	Machine	Start Time	Processing Time	Finish
1	2	2.1	14.5	16.6
2	1	3.5	6.25	9.75
3	1	9.75	11.98	21.73

Jain and Grossman (2001) have discussed 10 instances of this problem. They have formulated the same problem as a CLP model to be compatible with ILOG Scheduler. The processing times are assumed to be integers in their formulation. In our work, we allow these parameters to be real and test the ILOG Solver's capability to accommodate this change using our formulation. Table 3 shows the scheduling data of 3 orders on 2 machines.

For this problem, the master problem M2 was infeasible and hence M4 was solved using CLP. Table 4 shows the results of this case study. The results were equivalent to those obtained when M1 was solved using a MILP solver such as CPLEX.

4.3 Case Study 3

The following is a pooling NLP problem that has been studied extensively in the literature (Ryoo and Sahinidis, 1995).

$$\begin{aligned}
\min_{x,y} \quad & -9x_5 - 15x_9 + 6x_1 + 16x_2 + 10x_6 \\
s.t \quad & x_1 + x_2 = x_3 + x_4;\quad x_3 + x_7 = x_5 \\
& x_4 + x_8 = x_9;\quad x_7 + x_8 = x_6 \\
& x_{10}x_3 + 2x_7 \le 2.5x_5;\quad x_{10}x_4 + 2x_8 \le 1.5x_9 \\
& 3x_1 + x_2 = x_{10}\left(x_3 + x_4\right); \\
& \left(0,0,0,0,0,0,0,0,0,1\right) \le X \le (300,300,100, \\
& 200,100,300,100,200,200,3)
\end{aligned}$$

The proposed CLP based strategy reaches the global optimum without getting stuck at any of the infinite local solutions (Ryoo and Sahinidis, 1995). The global optimum is given by $\left(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*\right) =$ (0,100,0,100, $0,100,0,100,200,1)$.

The above problem when solved using the NLP solver CONOPT was dependent on initial guesses and has the possibility of converging at local optima.

4.4 Case Study 4

The following non convex NLP has been taken again from Ryoo and Sahinidis (1995)

$$\begin{aligned}
\min_{x,y} \quad & x_1 + x_2 \\
s.t \quad & x_1^2 + x_2^2 \le 4;\quad x_1^2 + x_2^2 \ge 1; \\
& x_1 - x_2 \le 1;\quad x_2 - x_1 \le 1 \\
& -2 \le X \le 2
\end{aligned}$$

The solution of this problem with GAMS modeler and CONOPT as solver did not give satisfactory results and it was found that the global optima reported was dependent on the initial guess provided. The global optima for this problem using the proposed CLP approach agreed with the reported value $X = \left(-1.414214, -1.414214\right)$ and $f = -2.828427$.

4.5 Case Study 5

The following problem is a MINLP with one binary and one continuous variable. This problem was proposed by Kocis and Grossmann (1988), and was also solved by Floudas et al. (1989), Ryoo and Sahinidis (1995) and Cardoso et al. (1997).

$$\begin{aligned}
\min_{x,y} \quad & f = 2x + y \\
s.t \quad & 1.25 - x^2 - y \le 0 \\
& x + y \le 1.6 \\
& 0 \le x \le 1.6 \\
& y \in \{0,1\}
\end{aligned}$$

The first nonlinear inequality constraint contains a non-convex term for the continuous variable x. The global optimum is (x, y; f) = (0.5, 1; 2).The master (M2) problem of this case study is infeasible and hence this problem was successfully solved to global optimality using the M4 transformation. Munawar and Gudi (2005) have solved this problem using GAMS Solvers viz. CONOPT2 and SNOPT. It has been shown that the optimum is strongly dependent on initial guesses. Such problems are not encountered when the above non-convex MINLP is solved using the proposed approach M4.

4.6 Case Study 6

The following problem is a non convex MINLP problem with three continuous and two discrete variables.

$$\begin{aligned}
\max_{x,y} \quad & x_1 + y_1 + x_1^2 + y_1^2 + x_1x_2 + y_1y_2 + y_2^2 + x_1^2y_1^2 + x_3 \\
s.t \quad & x_1x_2y_1 \ge 10;\quad x_1 + y_1 \ge 1 \\
& x_2 + y_2 \ge 1;\quad y_1 + y_2 \le 1 \\
& y_1y_2 \le 1;\quad x_3\left(1 - x_3\right) = 0 \\
& x_1 \le 3.5;\quad x_2 \le 15.5 \\
& 0 \le x_1 \le 4;\quad 0 \le x_2 \le 16;\quad 0 \le x_3 \le 1; \\
& Y \in \{0,1\}^3
\end{aligned}$$

Unlike in case study 3, this problem has a feasible solution for the master problem and is found to be $\left(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*; f^*\right) = \left(3,15,1,1,0;69\right)$.

This implies that the objective function cannot be less than 69. The sub-problem (M3) is solved and the global optima is determined to be $\left(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*; f^*\right) = \left(3.5,15.5,1,1,0; 85.25,\right)$.

This problem was also solved using M4. Though the global optimum was obtained, it was at a higher computational effort, as we report in Table 5.

Table 5: CLP Parameters for Case Study 6

Model		No. of Choice Points	No. of Failures
Method 1	M2	0	26
	M3	31	537
Method 2	M4	271	3735

This problem was also solved using the GAMS modeler with the DICOPT Solver. As can be seen in Table 6, solution using DICOPT is dependent on the initial guess to reach the global optimum. But in the proposed algorithm, there is no need for any initial guesses and yet the solutions are guaranteed to be globally optimal.

Table 6: Results for Case Study 6 using (DICOPT)

Initial Guesses $\left(x_1^0, x_2^0, x_3^0, y_1^0, y_2^0; f^0\right)$	**Optima** $\left(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*; f^*\right)$
No initial Guess	$(3.5, 15.5, 0, 1, 0; 84.25)$
$(3.5, 15.5, 0, 0, 1; 80,)$	$(3.5, 15.5, 0, 1, 0; 84.25)$
$(0, 10, 0.5, 1, 1; 80)$	$(0, 10, 0.5, 1, 1; 80)$
$(3.5, 10, 1, 1, 1; 80)$	$(3.5, 15.5, 1, 1, 0; 85.25,)$

5. CONCLUSION

In this paper, CLP has been used to solve non convex MINLP problems by transforming them into master problem that are of pure IP by nature. The enumeration strength of CP can be suitably exploited to generate solution to IP problem which can subsequently corrected in sub-problems. The solution of the master problem is used to tighten bounds and add additional constraints so as to reduce the computational burden. This method has been successfully tested on two MILP and two non convex NLP and two non convex MINLP problems. The superiority of the proposed method lies in the fact that it does not require any initial guess as required in the traditional techniques. It has also been demonstrated that the traditional techniques are not very robust and yield different optima, depending on the initial values.

ACKNOWLEDGEMENTS

We gratefully acknowledge the critical comments on an earlier version of this paper by Dr. K.P Madhavan, Prof. Emeritus, IIT Bombay. We also would like to thank Prof. Mani Bhushan, IIT Bombay for his comments on the paper.

REFERENCES

Cardoso, M. F., Salcedo, R. L., Feyo de Azevedo, S. and Barbosa, D., 1997, A simulated annealing approach to the solution of MINLP problems. *Comp. Chem. Eng.*, **21**(12): 1349.

Darby-Dowman, K., Little, J., Mitra, G., & Zaffalon, M. (1997). Constraint logic programming and integer programming approaches and their collaboration in solving an assignment scheduling problem. *Constraints—An International Journal,* **1**, 245–264.

Floudas, C. A., Aggarwal, A., Ciric, A. R., 1989, Global sdhbsd optimum search for nonconvex NLP and MINLP problems. *Comp. Chem. Eng.*, 13(10): 1117.

Heipcke, S. (1999b). Comparing constraint programming and mathematical programming approaches to discrete optimisation—the change problem. *Journal of Operational Research Society,* **50**, 581–595.

Hooker, J. N. (2000). *Logic-based methods for optimization: Combining optimization and constraint satisfaction*. New York: Wiley.

Hooker, J. N., & Osorio, M. A. (1999). Mixed logical, linear programming. *Discrete Applied Mathematics,* **96–97**, 395–442.

Hooker, J. N., Ottosson, G., Thorsteinsson, E. S., & Kim, H. J. (1999). On integrating constraint propagation and linear programming for combinatorial optimization. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence* **(AAM-99)** 136–141.

Jain, V., & Grossman, I.E. (2001). Algorithms for hybrid MILP/ CP models for class of optimization problems. *INFORMS Journal of Computing*, **13**, 258-276

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

Lee, S., & Grossmann, I. E. (2001). A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. *Computers and Chemical Engineering,* **25**, 1675– 1697.

Munawar, S.A., Bhushan, M., Gudi, R.D. and Belliappa, A.M., 2003, Cyclic scheduling of continuous multi-product plants in a hybrid flowshop facility. *Ind. Eng. Chem. Res.*, **42,** 5861-5882.

Munawar, S.A. and Gudi, R.D., "A nonlinear transformation based hybrid evolutionary method for MINLP solution", [To appear in *Chemical Engineering Research and Design* (2005)]

Puget, J. -F. (1994). A C++ implementation of CLP. In *Proceedings of SPICIS'94*. Singapore.

Ryoo, H. S. and Sahinidis, B. P., 1995, Global optimization of nonconvex NLPs and MINLPs with application in process design. *Comp. Chem. Eng.*, **19,** 551.

Smith, B. M., Brailsford, S. C., Hubbard, P.-M., & Williams, H.-P. (1997). The progressive party problem: integer linear programming and constraint programming compared. *Constraints—An International Journal,* **1**, 119–138.

Tawarmalani, M., & Sahinidis, N. V. (2000). Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming.* Ser. A, 99(3), 563-591, 2004

Van Hentenryck, P. (1989). *Constraint satisfaction in logic programming*. Cambridge, MA: MIT Press.