Piecewise Linear Relaxation Techniques for Solution of Nonconvex Nonlinear Programming Problems

Pradeep K. Polisetty and Edward P. Gatzke*

Department of Chemical Engineering

University of South Carolina

Columbia, SC 29208.

Abstract

A piecewise linear relaxation technique is presented for generation of mathematical programming relaxations of nonconvex nonlinear programming (NLP) problems. In this method, the original nonconvex nonlinear problem is converted to a Mixed-Integer Linear Programming (MILP) problem using logical constraints and outer approximations of convex function relaxations. A global solution of this MILP problem serves as a lower bound to the original problem. It may be possible to tighten the original problem variable bounds by use of a MILP-based bound contraction procedure. In some cases, multiple iterations of MILP-based bound contraction can aid in derivation of tighter variable bounds. The proposed techniques are implemented on several global optimization test problems. Computational results demonstrate that the gap between the lower bound and the upper bound can be significantly decreased by implementing the proposed technique. Additionally, the solution time using this method can be decreased significantly when variable contraction is performed in parallel.

Keywords

Piecewise linear relaxations, global optimization, variable space contraction, parallel optimization, branch-and-bound, branch-and-reduce.

*Author to whom correspondence should be addressed: gatzke@sc.edu

1 Introduction

Many industrial and engineering problems exhibit nonlinearity when posed as numerical optimization problems. Several algorithms have been developed to solve these optimization problems. While considerable progress has been made in solving global optimization problems, significant opportunities exist for further improvement. Branch-and-Bound [Falk and Soland, 1969] methods rely on generation of both lower and upper bounds on the problem objective function. The solution space is partitioned into regions until the lower bound on each partition is sufficiently close to or exceeds the overall problem upper bound. Any feasible local solution to the original nonconvex problem may serve as an initial upper bound for the global solution. Branch-and-Reduce [Ryoo and Sahinidis, 1995] extends the Branch-and-Bound algorithm, implementing bound tightening techniques to aid in rapid convergence of the algorithm. Reduction of the feasible space of any problem partition generally improves the lower bound on the partition, typically improving the problem convergence rate. Deterministic methods such as spatial Branch-and-Bound [Horst and Tuy, 1993, Quesada and Grossmann, 1993] and Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs [Gatzke and Voit, 2004] also depend on generation of relaxations of the original nonconvex nonlinear problems.

Numerous methods [Adjiman et al., 1998, McCormick, 1976, Tawarmalani and Sahinidis, 2002, Gatzke et al., 2002a] have been proposed to construct relaxations needed for the global solution of nonconvex optimization problems. The reformulation method of McCormick [McCormick, 1976, Smith, 1996, Byrne and Bogle, 1999] converts the original factorable nonconvex nonlinear algebraic functions into an equivalent form by the introduction of new variables and constraints. The reformulated problem contains only linear and simple nonlinear constraints. The convex relaxations for the simpler nonlinear constraints can be constructed using the convex and concave envelopes which are known for many simple algebraic functions. The αBB method [Adjiman et al., 1998, Adjiman et al., 1996] also generates convex relaxations for general twice-differentiable constrained NLP's. One advantage of this method as compared to the basic reformulation technique is that αBB does not require introduction of new variables. The αBB method requires the determination of bounds on the minimum eigenvalues of the hessian of the nonconvex functions. The Hybrid relaxation method [Gatzke et al., 2002b] combines both basic reformulation and αBB methods. This method may be advantageous in some cases where one of the above mentioned methods fails to generate a tight convex relaxation for the original NLP. Convex linear relaxations can also be generated by using the linearization strategy of Tawarmalani and Sahinidis [Tawarmalani and Sahinidis, 2000]. This method generates a convex nonlinear relaxation for the original factorable nonlinear problem. This nonlinear convex relaxation is further relaxed using multiple linearizations based on outer approximation at multiple points. The feasible space resulting from these outer approximations gives a convex linear relaxation of the original nonlinear problem. The bound on the relaxed problem is found by the solution of the resulting convex Linear Programming (LP) problem.

Global optimization algorithms, when used with the existing relaxation techniques may require a large amount of time to converge to the global solution. In this work, a MILP-based piecewise linear relaxation technique is used for generation of relaxations of nonconvex functions. This work considers nonconvex NLP's that only have continuous variables in the original problem. Using McCormick's [McCormick, 1976] reformulation method together with propositional logic constraints [Tyler and Morari, 1999], the original nonlinear nonconvex problem is relaxed to a MILP problem. The global solution to this MILP problem provides a lower bound on the original problem. This method can be advantageous in cases where the above mentioned relaxation methods fail to generate tight relaxations for the original problem, with the reservation that it requires the solution to a nonconvex MILP problem. The availability of robust Mixed Integer Programming (MIP) solvers like CPLEX 8.1 [ILOG, 2002] and IBM OSL [I. B. M. , 1997] may justify the use of this particular technique in many cases for solving nonconvex nonlinear problems.

Obviously, global solution of a lower bounding nonconvex MILP at every node in a Branch-and-Bound search tree is not desirable. Instead, any feasible integer solution to the MILP problem can still be a valid lower bound on the problem. The MILP relaxation can provide a tight lower bound for a single partition using robust MILP solution methods. The quality of this lower bound can be modified by changing the number of piecewise linear regions used in the lower bounding MILP problem.

The MILP relaxation technique can also be used for optimization-based bound tightening [Smith, 1996, Ryoo and Sahinidis, 1995, Adjiman et al., 2000]. After obtaining a local upper bounding solution for the original NLP problem, it may be possible to tighten the bounds on any variable by solving two optimization problems. These two problems are modified versions of the relaxed problem formulation which include an upper bound cut. This optimization-based bound tightening technique may help avoid branching of the original problem space during Branch-and-Reduce global optimization algorithm. This bound tightening technique can be applied on each of the n original variables in the problem, requiring the solution of $2n$ optimization problems.

The proposed MILP-based bound contraction technique when implemented on some problems may require a significant amount of time when solved on a single serial computer. In order to both decrease the computational burden and increase the efficiency of the algorithm, the proposed MILP-based bound contraction technique can be implemented in parallel [Polisetty and Gatzke, 2003]. This exploits the fact that the optimization-based bound tightening problems are decoupled from one another. Instead of contracting bounds of every variable, selecting those variables which may contribute the most to the gap between the original nonconvex problem and the relaxed problem can prove to be as effective and still reduce the computational burden.

2 MILP Based Relaxation/ Problem Formulation

Deterministic global optimization techniques typically rely on generation of convex relaxations of the original problem. Tighter relaxations usually result in faster convergence of the global algorithm. Tighter relaxations can aid in minimizing the partitioning of the solution space during the global optimization algorithm. A MILP-based piecewise linear relaxation technique is used in the current work to derive nonconvex relaxations to the original nonconvex nonlinear programming problem.

Propositional logic is used to divide the solution space for a given problem partition into multiple regions by introduction of a single binary variable for each region. The constraints corresponding to the region in which the solution lies are enforced while relaxing the constraints corresponding to other regions. First, the established reformulations for NLP and LP based relaxations are presented. The MILP-based relaxation is derived from the LP-based relaxation by use of propositional logic constraints. The LP-based and NLP-based relaxations are then compared to the MILP-based piecewise linear relaxation. Note that the MILP-based relaxation technique may add numerous binary variables and linear constraints. McCormick's reformulation technique is applied to the original problem shown in Equation 1:

$$P = \begin{cases} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \mathbf{s.t.} & g(\mathbf{x}) \leq 0 \\ & \\ & \mathbf{x^l} \leq \mathbf{x} \leq \mathbf{x^u} \end{cases} \tag{1}$$

Here, $\mathbf{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, and $g : \mathbb{R}^n \to \mathbb{R}^m$ are inequality constraints. This problem is reformulated to a equivalent simpler form by introducing new variables for separable nonconvex nonlinear terms. Here, the objective function and the constraint functions may be nonlinear. The reformulated problem takes the form shown in Equation 2.

$$P_1 = \begin{cases} \min\limits_{\mathbf{x},\mathbf{w}} & C^T \left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T \\ & \\ \mathbf{s.t} & A_1 \left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T \leq B_1 \\ & A_2 \left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T = B_2 \\ & \mathbf{w} = \eta(\mathbf{x}, \mathbf{w}) \\ & \\ & \mathbf{x^l} \leq \mathbf{x} \leq \mathbf{x^u} \\ & \mathbf{w^l} \leq \mathbf{w} \leq \mathbf{w^u} \end{cases} \tag{2}$$

In Equation 2, $\mathbf{w} \in \mathbb{R}^{o+l}$ is a vector of the o new nonlinear variables and l new linear variables introduced, $C^T[\mathbf{x}^T \;\; \mathbf{w}^T]^T$ is the linear objective function, $A_1[\mathbf{x}^T \;\; \mathbf{w}^T]^T \leq B_1$ are the linear inequality constraints resulted from the inequality constraints of the original problem shown in Equation 1, $A_2[\mathbf{x}^T \;\; \mathbf{w}^T]^T = B_2$ are new linear equality constraints defining the new linear variables, and $\mathbf{w} = \eta(\mathbf{x}, \mathbf{w})$, $\eta : \mathbb{R}^n \times \mathbb{R}^o \to \mathbb{R}^o$ provides the relationship between the new nonlinear variables and original variables. The advantage of this reformulation is that the new functions η_i contain simple nonlinear terms relating only 2 or 3 variables. Bounds on $\mathbf{w}$ can be inferred from bounds on $\mathbf{x}$ using interval analysis [Moore, 1979]. The reformulated problem contains only linear and simple nonlinear constraints for which convex relaxations can be constructed using convex envelopes known for many algebraic functions. Convex relaxations can be generated in a various number of ways. The NLP-based convex relaxation is shown in Equation 3.

$$
P_2 = \begin{cases} \min\limits_{\mathbf{x},\mathbf{w}} & C^T \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T \\ \\ \mathbf{s.t.} & A_1 \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T \leq B_1 \\ & A_2 \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T = B_2 \\ & \check{\eta}(\mathbf{x},\mathbf{w},\mathbf{x^l},\mathbf{x^u},\mathbf{w^l},\mathbf{w^u}) \leq w \leq \hat{\eta}(\mathbf{x},\mathbf{w},\mathbf{x^l},\mathbf{x^u},\mathbf{w^l},\mathbf{w^u}) \\ \\ & \mathbf{x^l} \leq \mathbf{x} \leq \mathbf{x^u} \\ & \mathbf{w^l} \leq \mathbf{w} \leq \mathbf{w^u} \end{cases} \tag{3}
$$

Here, $\check{\eta}$ and $\hat{\eta}$ are the convex under and concave over estimates of the original nonconvex expressions. The solution to this relaxed problem will serve as a lower bound to the original problem. Problems described by Equation 3 can be further relaxed to a Linear Programming (LP) problem by use of multiple linearizations of the convex nonlinear functions. The resulting LP relaxation problem is of the form shown in Equation 4.

$$
P_3 = \begin{cases} \min\limits_{\mathbf{x},\mathbf{w}} & C^T \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T \\ \\ \mathbf{s.t.} & A_1 \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T \leq B_1 \\ & A_2 \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T = B_2 \\ & A_3 \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T \leq B_3 \\ \\ & \mathbf{x^l} \leq \mathbf{x} \leq \mathbf{x^u} \\ & \mathbf{w^l} \leq \mathbf{w} \leq \mathbf{w^u} \end{cases} \tag{4}
$$

Here, $A_3 \begin{bmatrix}\mathbf{x}^T & \mathbf{w}^T\end{bmatrix}^T \leq B_3$ expresses the new linear constraints obtained from multiple outer approximations of the convex and concave functions in Equation 3. Note that A_3 , B_3 and the bounds on $\mathbf{w}$ will change as the bounds on $\mathbf{x}$ are modified.

The proposed MILP-based piecewise linear relaxation technique can generate tighter relaxations as compared to those generated by LP-based and NLP-based relaxation methods. The problem space for the nonlinear terms is divided into multiple regions using propositional logic constraints. Outer approximations of nonlinear functions and secant under-estimates and over-estimates are then generated for each individual region, thereby converting the original nonlinear problem into a mixed integer linear programming problem. Most of the constraints in this MILP problem are relaxed while enforcing only those constraints corresponding to the single region containing the solution.

The MILP-based piecewise linear relaxation technique is illustrated on an example constraint which has a simple nonlinear term, $w = x^c$. Here, w is the new variable introduced during reformulation and c is a non-integer constant. The variable space for x is divided into S regions separated by $(S-1)$ boundaries. A binary variable is introduced for each region resulting in S new binary variables. For the S regions, $2(S-1)$ propositional logic inequality constraints are then added to represent these regions. In this technique, b_1 is forced to a take a value of 1 if x is in between x^l and s_1, where s_1 is the upper bound on the first region. The first region constraint is specified as follows:

$$-s_1 + x \leq M(1 - b_1)$$

The regions 2 through $(S-1)$ are specified with the following constraints:

$$\begin{aligned} s_1 - x + \delta &\leq M(1-b_2) \\ -s_2 + x &\leq M(1-b_2) \\ s_2 - x + \delta &\leq M(1-b_3) \\ -s_3 + x &\leq M(1-b_3) \\ &\vdots \end{aligned} \tag{5}$$

where δ is a small value used to ensure that the value of the variable x does not end up at the boundaries separating the regions. The final region constraint is specified as follows:

$$s_{(S-1)} - x + \delta \leq M(1 - b_S)$$

Since only a single region can contain the solution, an equality constraint is added to ensure that solution lies in only one region.

$$\sum_{i=1}^{S} b_i = 1 \tag{6}$$

Based on these propositional logic constraints and the constraint shown in Equation 6, if the value of the variable x is smaller than s_1, the binary variable b_1 is forced to take value of 1. On the other hand, if a binary variable takes a value of zero, the respective constraint is relaxed as the right hand side takes a large value of M. The nonlinear expression is replaced by outer approximation constraints written for each region. Depending on the number of linearizations, O, used to outer approximate the nonlinear expression in each piecewise region, the linear over-estimate constraints for the nonlinear expression in this example can be written as follows:

$$\begin{aligned} w \leq\ & f(x)|_{x=x^*_{i,j}} + \left.\frac{\partial f(x)}{\partial x}\right|_{x=x^*_{i,j}} (x - x^*_{i,j}) + M(1-b_i) \\ & \forall x^*_{i,j},\ where\ j = 1..O\ and\ \forall i = 1..S \end{aligned}$$

where $x^*_{i,j}$ are the linearization points, $f(x) = x^c$ is the nonlinear expression, and $\frac{\partial f(x)}{\partial x}$ is the gradient of the function, which in this example is cx^{c-1}. For each region, this function can be under estimated by a secant constraint written as follows:

$$secant(x^c, x^L, x^U) \leq w + M(1 - b_i) \quad \forall i = 1...S$$

where x^L, x^U are the lower and upper bounds for a particular region. If the binary variable related to a particular region takes a value of 1, the corresponding linearization and secant constraints are enforced while relaxing the constraints corresponding to other regions. The final MILP-based piecewise linear relaxation problem can be represented in a general form as follows:

$$P_4 = \begin{cases} \min\limits_{\mathbf{x},\mathbf{w},\mathbf{z}} \quad C\left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T \\ \\ \textbf{s.t.} \quad A_1\left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T \leq B_1 \\ \qquad A_2\left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T = B_2 \\ \qquad A_3\left[\mathbf{x}^T \;\; \mathbf{w}^T\right]^T \leq B_3 \\ \qquad A_4\left[\mathbf{x}^T \; \mathbf{w}^T \; \mathbf{z}^T\right]^T \leq B_4 \\ \\ \qquad \mathbf{z} \in \{0,1\}^Q \\ \qquad \mathbf{x^l} \leq \mathbf{x} \leq \mathbf{x^u} \\ \qquad \mathbf{w^l} \leq \mathbf{w} \leq \mathbf{w^u} \end{cases} \tag{7}$$

Here, $\mathbf{z}$ are the binary variables introduced and $A_4\left[\mathbf{x}^T \; \mathbf{w}^T \; \mathbf{z}^T\right]^T \leq B_4$ are the linear constraints representing logic constraints, outer approximation constraints, and secant constraints for all Q regions. Note that removal of the logic constraints $A_4\left[\mathbf{x}^T \; \mathbf{w}^T \; \mathbf{z}^T\right]^T \leq B_4$ and binary variables results in the equivalent LP-based convex relaxation.

The nonlinear expression $w = x^c$ with outer approximation constraints and secant under estimation constraint are illustrated in Figure 1. In this example, the variable space for x is divided into 2 regions and 2 linearizations are derived for the each region. Complex factorable nonlinear problems can be automatically reformulated using McCormick's method so that the final reformulated problem has expressions which involve only 2 or 3 variables. These simple expressions include bilinear terms xy, variable raised to constants x^c, where c can be non integer constant, even integer, odd integer, or negative integer. Other expressions include natural log of a variable $ln(x)$, exponential of a variable $exp(x)$, $sin(x)$, $cos(x)$. In the case of bilinear terms, binary variables and logic constraints can be used for both variables. MILP-based piecewise linear relaxations can be developed to nonconvex problems involving these simple nonlinear terms.

The MILP-based piecewise linear relaxation is now compared to that of LP-based linear relaxation for a standard global optimization test problem [Dixon et al., 1975] shown in Equation 8.

$$\begin{aligned} \min_{\mathbf{x}} \quad & 4x_1^2 - 2.1x_1^4 + \tfrac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \\ where \quad & \qquad -4 \le (x_1, x_2) \le 4 \end{aligned} \tag{8}$$

For this problem, the objective function surface, the MILP-based piecewise relaxation surface, and the LP-based relaxation surface are shown in Figure 2. The MILP-based piecewise relaxation is generated by considering 4 regions for each nonlinear expression involved in the problem. This figure demonstrates that the MILP-based lower bound is far improved as compared to that of LP-based lower bound for this problem. Since the gap between the objective function and that of the MILP based lower bound is small, this may aid in rapid convergence of global optimization algorithms, thereby requiring fewer partitions while searching for the global solution.

Deterministic global optimization techniques like Branch-and-Bound and Branch-and-Reduce methods depend on generation of relaxations of the original nonconvex problem. In the Branch-and-Bound method, a LP-based relaxation of the original nonlinear problem is first constructed. The solution to this relaxation problem provides a valid lower bound on the original problem. Local minimization techniques can be employed to obtain an initial upper bound for the problem. The feasible region is then divided into partitions and lower bounds are generated for the new partitions. The algorithm terminates when the lower bounds for all partitions either exceeds or are sufficiently close to the global upper bound. A more detailed description of Branch-and-Bounds methods can be found in [Falk and Soland, 1969, Ryoo and Sahinidis, 1995]. It may be possible to generate much tighter relaxations by using the proposed MILP-based piecewise linear relaxation technique as compared to LP-based relaxation technique for each partition during the Branch-and-Bound methods. This may aid in rapid convergence of the algorithm. The obvious disadvantage is that a MILP must be solved at each node in the Branch-and-Bound tree if this MILP-based relaxation is to be used. A graphical interpretation of Branch-and-Bound method using MILP-based relaxation technique for a single step is given in Figure 3.

3 MILP-based Bound Contraction

Typical global optimization problems may have tens or hundreds of variables. Interval analysis bound tightening [Moore, 1979], and optimization-based bound tightening [Ryoo and Sahinidis, 1995, Smith, 1996, Adjiman et al., 2000] may aid in derivation of tighter variable bounds by eliminating infeasible region or regions that can be proven to not include a solution any better than the current upper bound. The MILP-based bound contraction shown in Figure 4 is similar to optimization-based bound tightening, except in the former a MILP-based piecewise relaxation is used and in the later a LP relaxation is used. These bound tightening procedures can be applied on any partition during the Branch-and-Reduce algorithm. The proposed MILP-based bound contraction technique may help avoid branching of the original problem space during Branch-and-Reduce global optimization algorithm. It is shown in Theorem 1 by contradiction that the MILP-based bound contraction technique will not eliminate a point feasible in the original problem P_1 with an objective function value less than the current upper bound. This theorem is graphically demonstrated in Figure 5. After obtaining any local upper bounding solution, it may be possible to tighten the bounds on any variable x_i by solving two MILP-based piecewise linear relaxation problems with an upper bound cut formulated as :

$$P_5 = \begin{cases} \min\limits_{\mathbf{x},\mathbf{w},\mathbf{z}} \quad \pm x_i \\ \\ \mathbf{s.t} \quad C\left[\mathbf{x}^T \ \mathbf{w}^T\right]^T \leq ubd \\ \quad\quad A_1\left[\mathbf{x}^T \ \mathbf{w}^T\right]^T \leq B_1 \\ \quad\quad A_2\left[\mathbf{x}^T \ \mathbf{w}^T\right]^T = B_2 \\ \quad\quad A_3\left[\mathbf{x}^T \ \mathbf{w}^T\right]^T \leq B_3 \\ \quad\quad A_4\left[\mathbf{x}^T \ \mathbf{w}^T \ \mathbf{z}^T\right]^T \leq B_4 \\ \\ \quad\quad \mathbf{z} \in \{0,1\}^Q \\ \quad\quad \mathbf{x^l} \leq \mathbf{x} \leq \mathbf{x^u} \\ \quad\quad \mathbf{w^l} \leq \mathbf{w} \leq \mathbf{w^u} \end{cases} \tag{9}$$

where ubd is the current upper bound on the problem.

Theorem 1. *MILP-based bound contraction problem P_5 will not eliminate a point feasible in the original problem P_1 with objective function value $(\mathscr{O}) <$ current upper bound (UBD).*

Problem P_1 - Reformulated problem shown in Equation 2.

Problem P_4 - MILP-based piecewise linear relaxation problem shown in Equation 7.

Problem P_5 - MILP-based bound contraction problem shown in Equation 9.

Proof.

Assumption: Problem P_4 is a relaxation of the problem P_1 i.e. $\mathscr{O}(P_4) \leq \mathscr{O}(P_1)$

Know that problem P_5 includes a upper bound cut on problem P_4. Assume there exists a point p^* in the region eliminated by problem P_5 but is feasible in problem P_1, whose objective function value $\mathscr{O}(p^*) <$ UBD. Since problem P_4 is a relaxation of problem P_1, p^* must be a feasible point in P_4. However the upper bound cut added to problem P_4 to derive problem P_5 contradicts the existence of this point. Therefore p^* is infeasible in P_5. Hence problem P_5 cannot eliminate a point feasible in the original problem P_1 with objective function value $(\mathscr{O}) <$ current upper bound (UBD). □

The proposed MILP-based bound contraction technique is implemented on the test problem shown in Equation 8. The MILP-based bound contraction technique is first implemented sequentially for contraction of the bounds on the original variables at the root node of the Branch-and-Reduce tree. Computational results are presented for multiple iterations of MILP-based bound contraction with varying number of regions. After deriving tighter bounds using the MILP-based bound contraction technique, interval analysis is used to further tighten the variable bounds. Finally, the LP-based lower bound is determined for each iteration. The LP-based lower bound for multiple iterations of MILP-based bound contraction with varying regions and the upper bound for this problem are shown in Figure 6.

In Figure 6, the results for 1 region in the MILP-based relaxation are equivalent to those using just the LP-based relaxation for bounds contraction. Examining this figure, using the LP-based relaxation for optimization-based bounds contraction results in no improvement of the lower bound for the root node partition. This can be inferred from Figure 2. A upper bound objective function cut at -1.0 for the LP-based relaxation would not aid in reducing the bounds for x_1 and x_2. This can be seen by examining the four planes $x_1 = \pm 1$ and $x_2 = \pm 1$. There are multiple feasible points in these planes with the objective function cut, implying no improvement in the bounds using bounds contraction. On the other hand, the MILP-based bound contraction results in some improvement. However, the two nearly equivalent local minima limit the extent of possible bounds contraction. In this problem, partitioning of the root node would be necessary for global convergence.

The MILP-based piecewise linear relaxation technique with optimization bound tightening was implemented on Heat exchanger network design [Liebman et al., 1986] and Reactor network design [Manousiouthakis and Sourlas, 1992] problems shown in Equations 10 and 11.

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & x_1 + x_2 + x_3 \\
36000x_1 - 100000x_4 - 120x_1x_4 &= -10000000 \\
32000x_2 + 100000x_4 - 100000x_5 - 80x_2x_5 &= 0 \\
4000x_3 + 50000000x_5 &= = 50000000 \\
(0,0,0,100,100) &\le \mathbf{x} \le (15834, 36250, 10000, 300, 400)
\end{aligned} \tag{10}
$$

The global optimal solution for this problem is $\mathbf{x} = (579.307, 1359.97, 5109.97, 182.018, 295.601)$ with objective function value of 7049.249.

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & -x_4 \\
x_1 + k_1x_1x_5 &= 1 \\
-x_1 + x_2 + k_2x_2x_6 &= 0 \\
x_1 + x_3 + k_3x_3x_5 &= 1 \\
-x_1 + x_2 - x_3 + x_4 + k_4x_4x_6 &= 0 \\
x_5^{0.5} + x_6^{0.5} &\le 4 \\
(0,0,0,0,0,0) &\le \mathbf{x} \le (1,1,1,1,16,16)
\end{aligned} \tag{11}
$$

where $k_1 = 0.0976$, $k_2 = 0.99k_1$, $k_3 = 0.0392$, and $k_4 = 0.9k_3$. The global optimum for this problem is $\mathbf{x} = (0.7715, 0.517, 0.204, 0.388812, 3.037, 5.096)$ with objective function value of -0.388812.

The LP-based lower bounds for multiple passes of MILP-based bound contraction with varying regions and the upper bound for the Heat exchanger network design problem are shown in Figure 7. Similar implementation is performed on Reactor network design problem and the LP-based lower bounds and the upper bound for the problem are shown in Figure 8. It should be noted that for both the cases, the gap between the global upper bound and LP-based lower bound decreased significantly prior to global search for the solution as the number of passes of MILP-based bound contraction and number of regions are increased.

Increasing the number of regions and performing multiple passes may take a significant amount of time. To alleviate this problem, the MILP-based bound contraction can be implemented in parallel, as the optimization based bound tightening problems are decoupled. This bound tightening technique can be applied on each of the n original variables in the problem, which requires the solution of $2n$ optimization problems. In order to decrease the computational burden and increase the efficiency and speedup of the algorithm, the proposed optimization technique is implemented in parallel. This exploits the fact that the optimization based bound tightening problems given in Equation 9 are all decoupled from one another.

4 Parallel Bound Contraction

The proposed MILP-based bound contraction can significantly improve the lower bound on the problem before any partitioning of the variable space is performed. However, this technique may consume a considerable amount of time for certain problems when implemented on a single processor. The main objective in parallelizing the sequential MILP-based bound contraction technique is to distribute the work among several processors so as to minimize the overall time required to solve $2n$ optimization problems. For this, the proposed technique is implemented on multiple processors using 2 Manager / Worker paradigm described in Figure 9.

In this parallel implementation, an application running as Manager 1 formulates the MILP-based bound contraction problem and transfers the problem through a FIFO pipe [Brian, 1997] to another manager. The advantage of using a FIFO pipe is that unrelated processes can communicate and receive problem data through this pipe. Manager 1 then waits for the results file containing new variables bounds from Manager 2. Manager 2 sends the original variable bounds to worker processors. The worker processors then solve the MILP-based optimization problem and returns the new variable bounds to the Manager 2. Manager 2 then passes the results to Manager 1. The standard *Message Passing Interface* (MPI) [Forum, 1997] is used for communication between the processors. The presented parallel technique can implement multiple passes of optimization bound contraction technique, thereby generating extremely tight bounds. This algorithm is implemented on a Beowulf style computer using CPLEX 8.0 [ILOG, 2002] for solution of MILP problems. The machine has 32 nodes, each with a single 933 MHz Pentium-3, 1Gbyte of memory, and 15 Gbytes of disk space.

The proposed parallel technique has been tested on Heat exchanger network design and Reactor network design problems presented earlier. The total time required to perform multiple iterations of MILP based optimization bound tightening with varying regions, both for serial and parallel implementations are shown in Table 1 for the Heat exchanger network design problem. For illustration, $2n$ processors are used as workers, where n is the number of original variables in the problem. Computational results demonstrate that total time decreased significantly in the parallel implementation.

Results for the Reactor network design problem are shown in Table 2. It is observed that for problems of these size, the total time required to perform multiple iterations of MILP-based bound tightening for 1 and 2 regions are almost similar for both serial and parallel implementation. This is due to synchronization and communication overhead between the manger and worker processors. In many cases, overhead due to communication and synchronization between manger and worker processors may consume significant amount of time. However, significant decrease in time can be obtained for many engineering optimization problems as they involve hundreds or even thousands of variables.

5 Conclusions

In this paper, a MILP-based piecewise linear relaxation has been used for generation of relaxations of nonconvex expressions. Using McCormick's [McCormick, 1976] reformulation method together with the propositional logic constraints method, the original nonlinear nonconvex problem is relaxed to a MILP problem. The solution to this MILP problem serves as the lower bound to the original problem. Upon obtaining any local upper bounding solution, it may be possible to tighten the bounds on any variable by solving two MILP-based relaxation problems with an upper bound cut. In some cases, multiple passes of optimization based bound tightening can help in deriving tighter bounds. This technique helps in avoiding branching of the original problem space during Branch-and-Reduce global optimization algorithm. The proposed technique is implemented on several problems and the computational results justify the use of MILP-based piecewise linear relaxation technique for solving global optimization problems. Substantial decrease in time can be obtained by utilizing the proposed parallel tightening technique.

Acknowledgments

The authors would like to acknowledge financial support from the National Science Foundation under Grant No. 0238663. The parallel computations were performed on the 32-node Beowulf cluster, daniel, in the Department of Computer Science and Engineering at USC. We thank Dr. Duncan Buell for access to this machine. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[Adjiman et al., 2000] Adjiman, C. S., Androulakis, I. P., and Floudas, C. A. (2000). Global Optimization of Mixed-Integer Nonlinear Problems. *AIChE J.*, 46(9):1769–1797.

[Adjiman et al., 1996] Adjiman, C. S., Androulakis, I. P., Maranas, C. D., and Floudas, C. A. (1996). A Global Optimization Method, α BB, for Process Design. *Comput. Chem. Eng.*, 20:S419–S424.

[Adjiman et al., 1998] Adjiman, C. S., Dalliwig, S., Floudas, C. A., and Neumaier, A. (1998). A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLPs - I Theoretical Advances. *Comput. Chem. Eng.*, 22(9):1137–1158.

[Brian, 1997] Brian, E. H. (1997). Beej's guide to Unix interprocess communication.

[Byrne and Bogle, 1999] Byrne, R. P. and Bogle, I. D. L. (1999). Global optimisation of constrained non-convex programs using reformulation and interval analysis. *Computers and Chemical Engineering*, 23:1341–1350.

[Dixon et al., 1975] Dixon, L. C. W., Gomulka, J., and Szega, G. P. (1975). *Towards Global Optimisation*. Number 29-54. North-Holland, Amsterdam. camel back function.

[Falk and Soland, 1969] Falk, J. E. and Soland, R. M. (1969). An Algorithm for Separable Nonconvex Programming Problems. *Management Science*, 15(9):550–569.

[Forum, 1997] Forum, M. P. I. (1997). MPI-2: Extensions to the Message-Passing Interface. Technical report, University of Tennessee, Knoxville, TN.

[Gatzke et al., 2002a] Gatzke, E. P., Tolsma, J. E., and Barton, P. I. (2002a). Construction of Convex Function Relaxations Using Automated Code Generation Techniques. *Optimization and Engineering*, 3(3):305–326.

[Gatzke et al., 2002b] Gatzke, E. P., Tolsma, J. E., and Barton, P. I. (2002b). Construction of Convex Function Relaxations Using Automated Code Generation Techniques. *Optimization and Engineering*, 3(3):305–326.

[Gatzke and Voit, 2004] Gatzke, E. P. and Voit, E. O. (2004). Modeling of Metabolic Systems Using Global Optimization Methods. In *IFAC ADCHEM (Advanced Control of Chemical Processes) Conference*, Hong Kong.

[Horst and Tuy, 1993] Horst, R. and Tuy, H. (1993). *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin, second edition.

[I. B. M. , 1997] I. B. M. (1997). IBM Optimization Solutions and Library Linear Programming Solutions. Technical report, I. B. M.

[ILOG, 2002] ILOG (2002). *ILOG CPLEX 8.1: User's Manual*. Mountain View, CA.

[Liebman et al., 1986] Liebman, J., Lasdon, L., Schrage, L., and Waren, A. (1986). Modeling and Optimization with GINO. Palo Alto, CA.

[Manousiouthakis and Sourlas, 1992] Manousiouthakis, M. and Sourlas, D. (1992). A global optimization approach to rationally constrained rational programming. *Chem. Engng. Commun.*, 115:127–147.

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

[Moore, 1979] Moore, R. E. (1979). *Methods and Applications of Interval Analysis*. SIAM, Philadelphia.

[Polisetty and Gatzke, 2003] Polisetty, P. K. and Gatzke, E. P. (2003). Parallel Contraction of Variable Space for Global Solution of NLPs. In *FOCAPD, Fundamentals of Computer Aided Process Design*, Princeton, NJ.

[Quesada and Grossmann, 1993] Quesada, I. and Grossmann, I. E. (1993). Global optimization algorithm for heat exchanger networks. *Industrial and Engineering Chemistry Research*, 32:487–499.

[Ryoo and Sahinidis, 1995] Ryoo, H. S. and Sahinidis, N. V. (1995). Global Optimization of Nonconvex NLPS and MINLPs with Application to Process Design. *Comput. Chem. Eng.*, 19(5):551–566.

[Smith, 1996] Smith, E. M. B. (1996). *On the Optimal Design of Continuous Processes*. PhD thesis, Imperial College, London.

[Tawarmalani and Sahinidis, 2000] Tawarmalani, M. and Sahinidis, N. V. (2000). Global Optimization of Mixed Integer Nonlinear Programs: A Theoretical and Computational Study. Technical report, University of Illinois.

[Tawarmalani and Sahinidis, 2002] Tawarmalani, M. and Sahinidis, N. V. (2002). Global Optimization of Mixed-Integer Nonlinear Programs: A Theoretical and Computational Study. *Mathematical Programming*.

[Tyler and Morari, 1999] Tyler, M. L. and Morari, M. (1999). Propositional Logic in Control and Monitoring Problems. *Automatica*, 35:565–582.

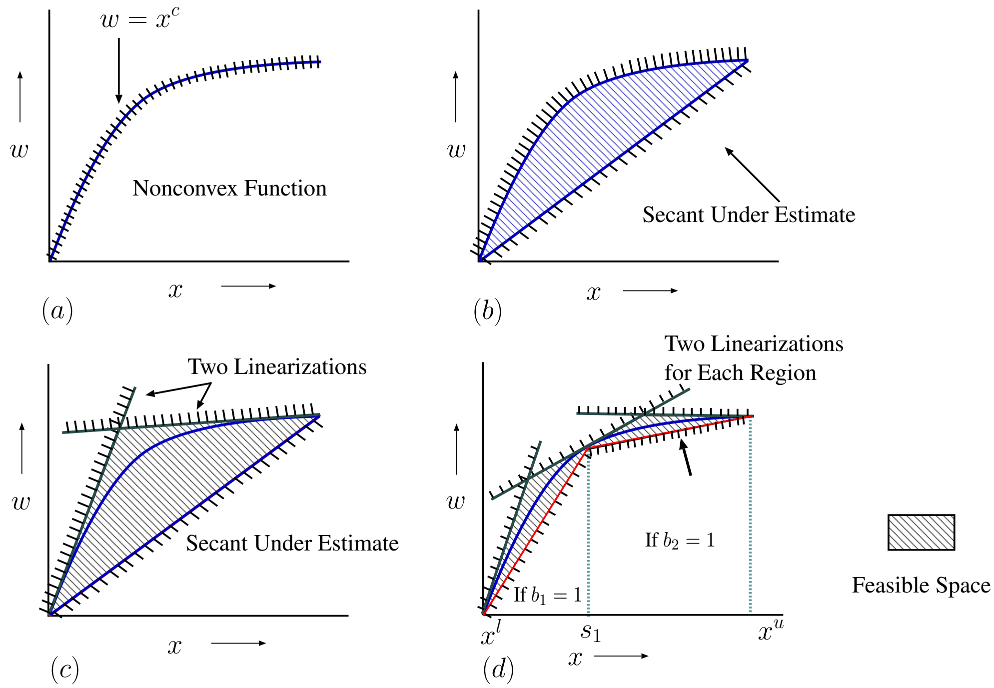

Figure 1: (a) Original nonlinear nonconvex constraint $w = x^c$. (b) Relaxation of $w = x^c$ using the secant under estimate. (c) Two outer approximation constraints over estimating the nonlinear expression resulting in linear constraints. (d) Two outer approximation constraints and a secant under estimate constraint for each region in the MILP-based piecewise relaxation problem.

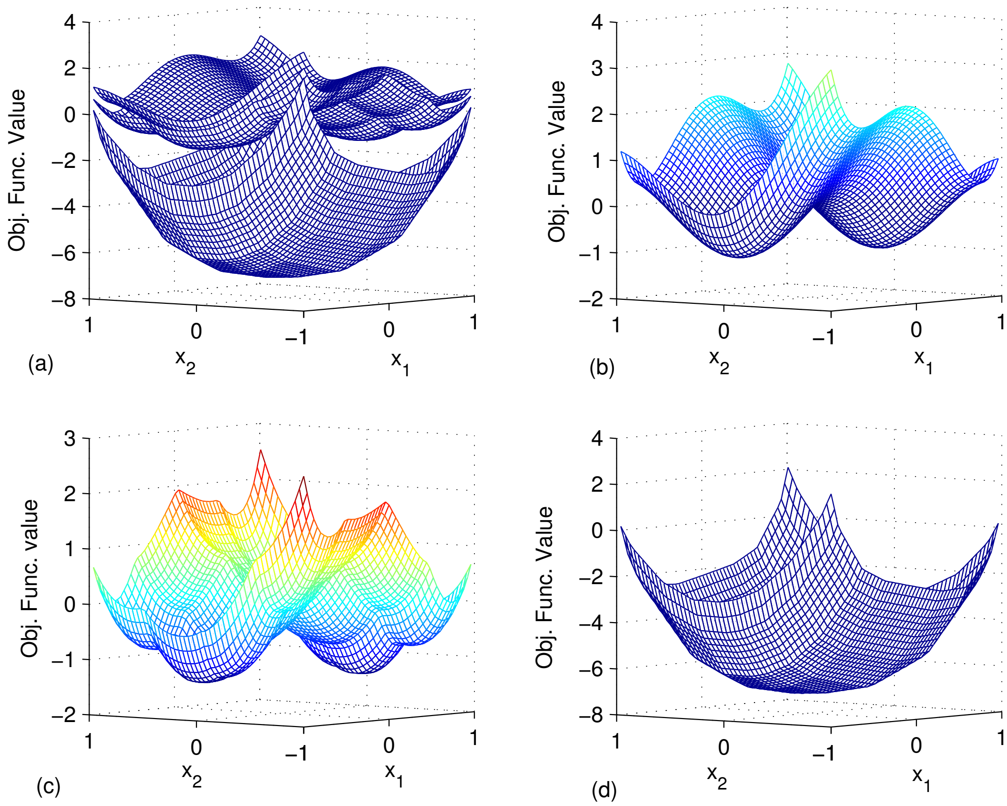

Figure 2: (a) The objective function surface, MILP-based relaxation surface, and the LP-based relaxation surface for 6 Hump camel back problem. (b) The objective function surface. (c) MILP-based relaxation surface generated with 4 regions. (d) LP-based relaxation surface.

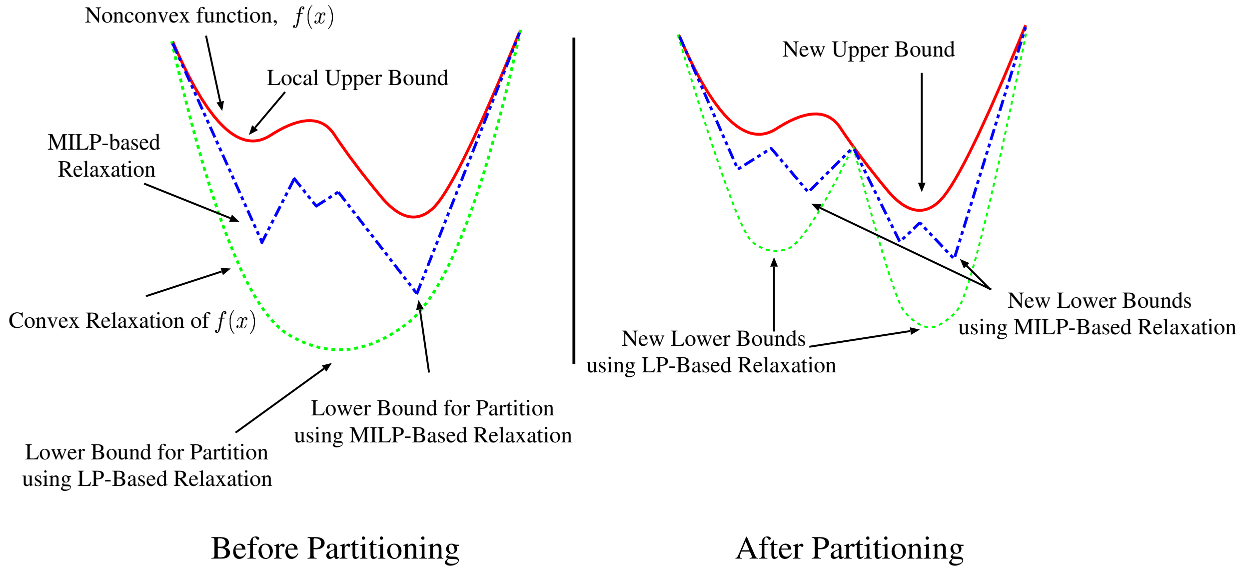

Figure 3: A single branch-and-bound step for a nonconvex function of a continuous variable using LP-based relaxation and MILP-based relaxation.

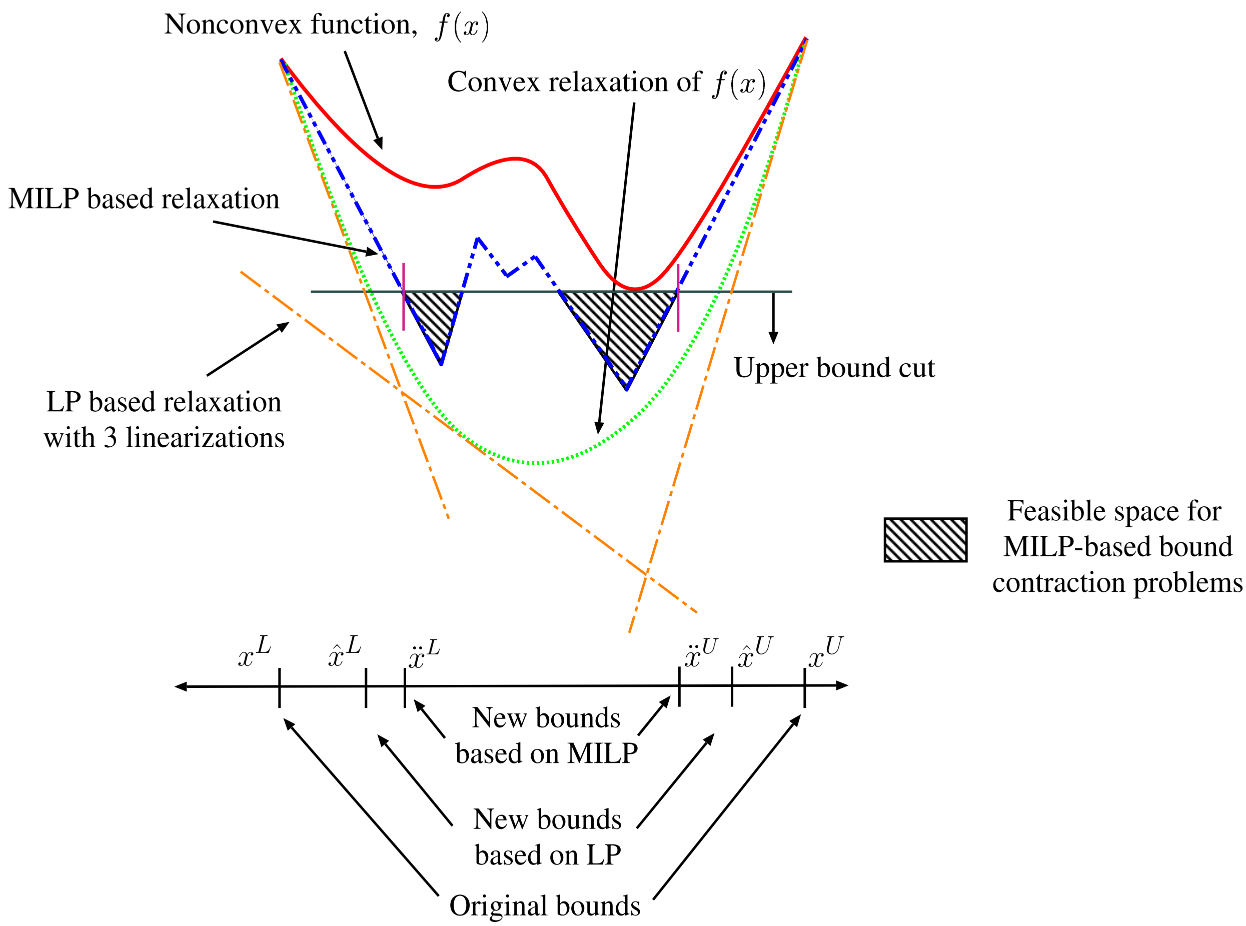

Figure 4: MILP-based bound contraction showing the new bounds resulted from the use of the convex relaxation and nonconvex MILP-based relaxation.

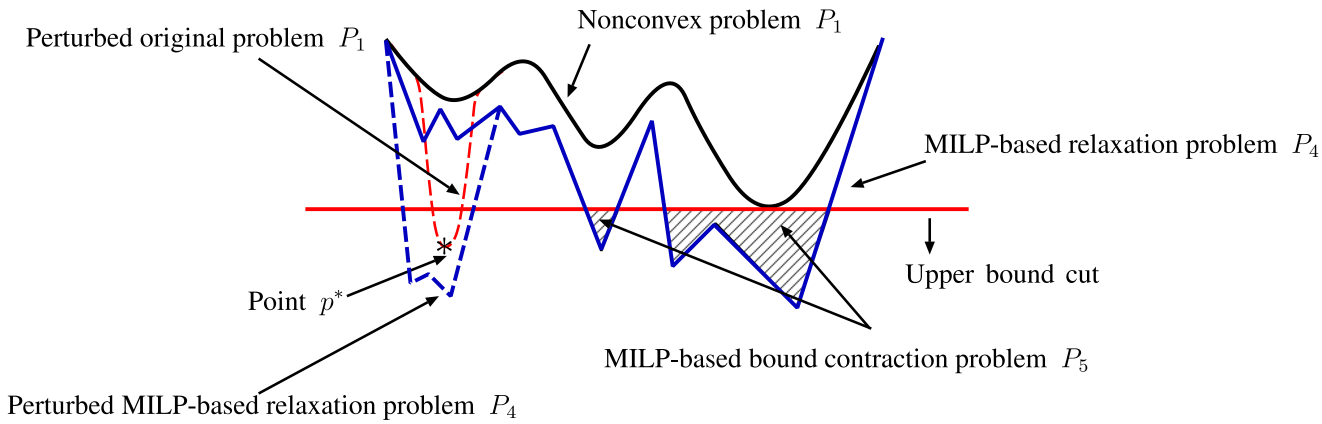

Figure 5: Graphical Interpretation of Theorem 1.

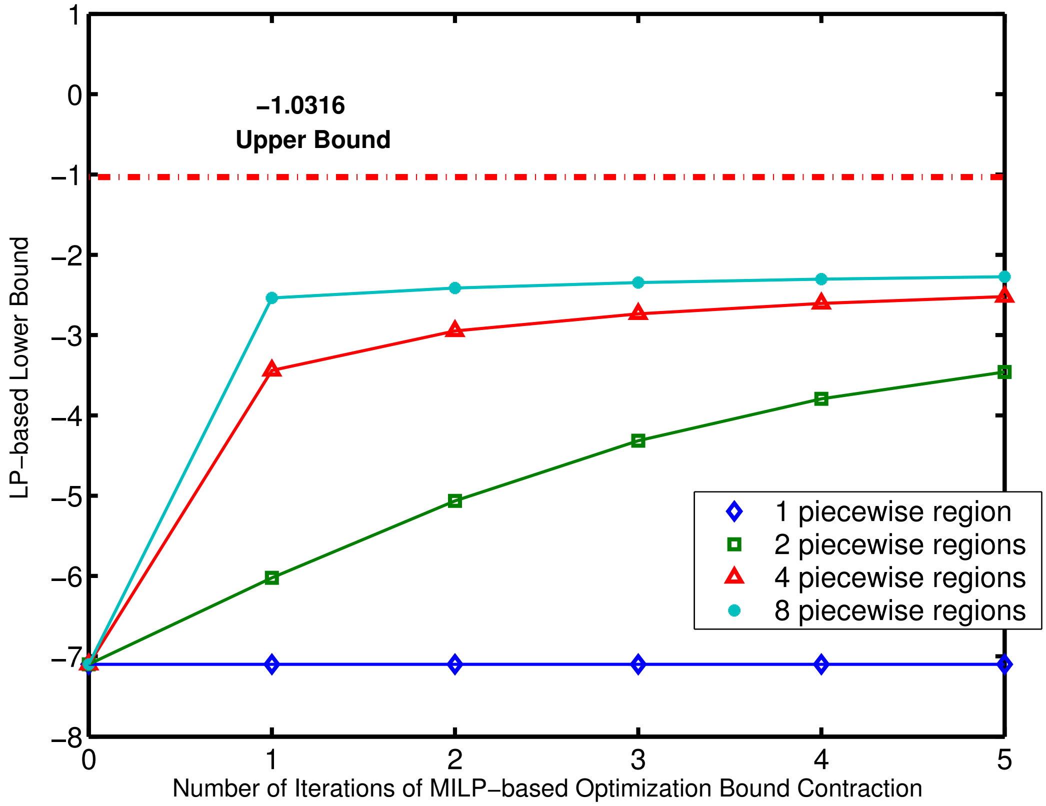

Figure 6: LP-based lower bounds for multiple iterations of MILP-based Bound Contraction with varying partitions for 6 Hump camel back problem shown in Equation 8.

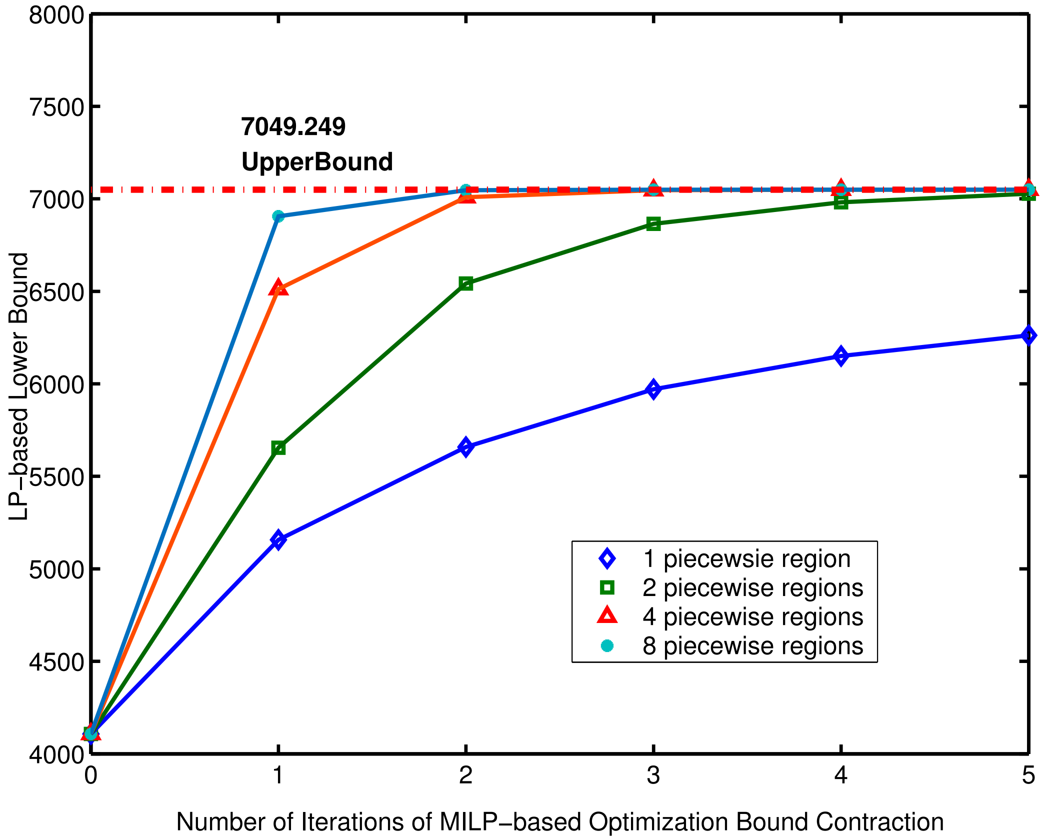

Figure 7: LP-based lower bounds after multiple iterations of MILP-based Bound Contraction method with varying number of partitions for Heat Exchanger Network Design Problem shown in Equation 10.

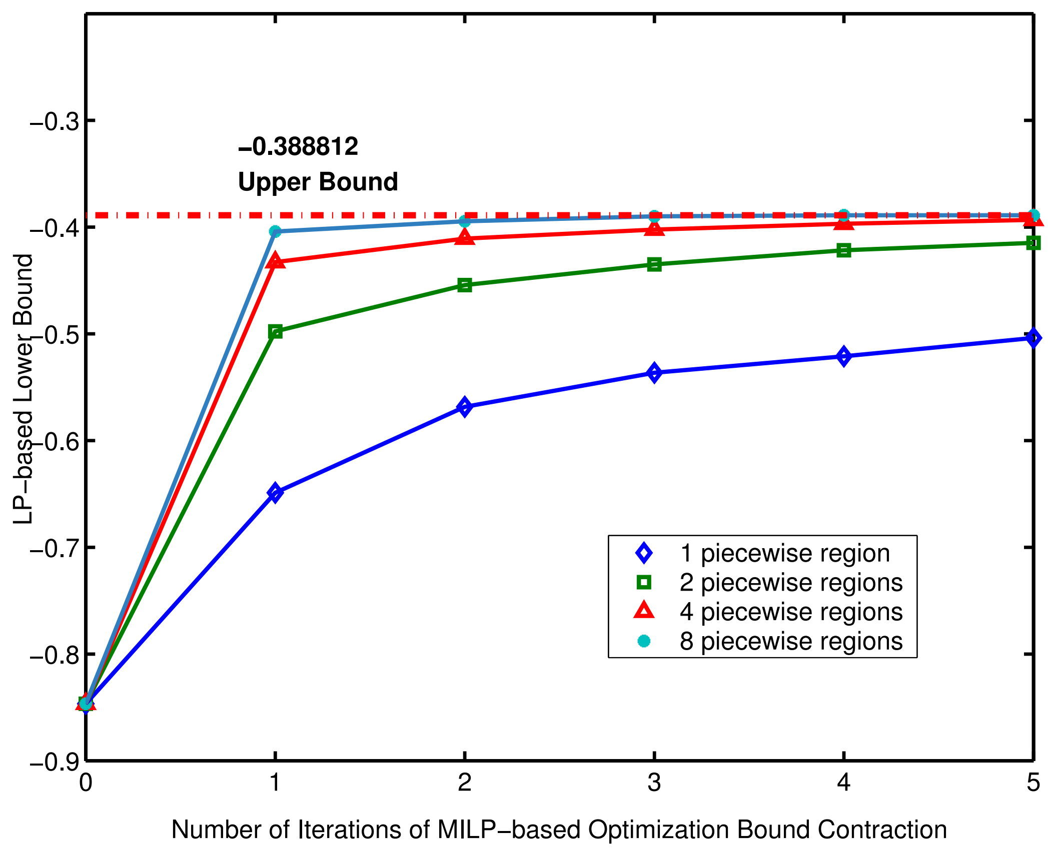

Figure 8: LP-based lower bounds after multiple iterations of MILP-based Bound Contraction method with varying number of partitions for Reactor Network Design Problem shown in Equation 11.

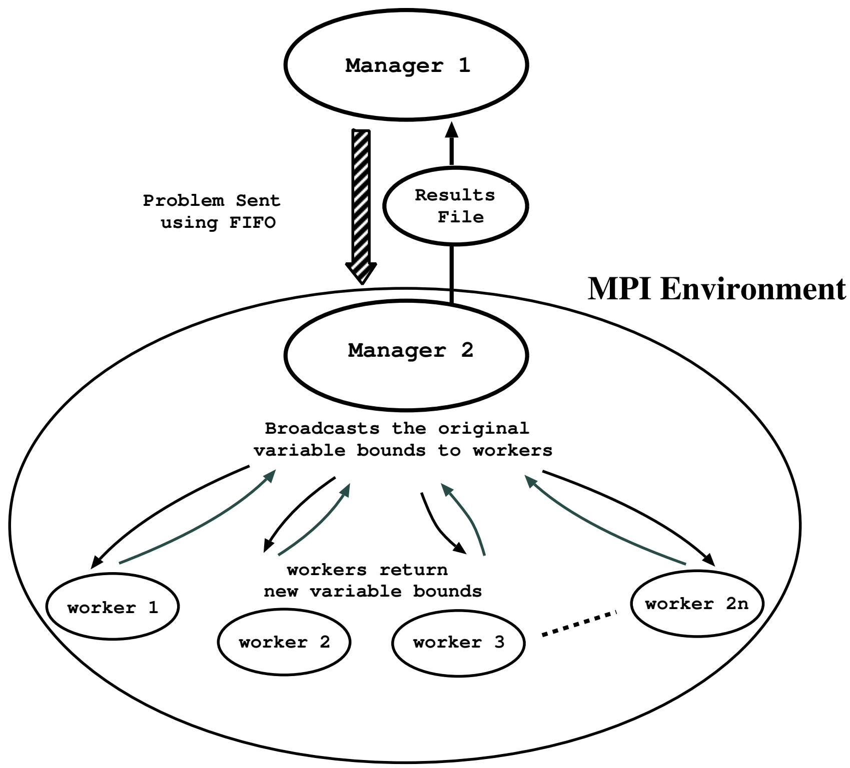

Figure 9: Parallel MILP-based Bound Contraction Technique using 2 Manager / Worker paradigm.

Table 1: Total time required in performing multiple iterations of MILP-based Bound Contraction with varying partitions for both serial and parallel implementation on Heat Exchanger Network Design Problem.

No of Regions	Number of Iterations of MILP-based Bound Contractions									
	1		2		3		4		5	
	Serial	Parallel	Serial	Parallel	Serial	Parallel	Serial	Parallel	Serial	Parallel
1	0.108	0.068	0.251	0.096	0.323	0.143	0.429	0.150	0.538	0.178
2	0.349	0.090	0.579	0.115	0.826	0.199	1.070	0.228	1.312	0.277
4	0.800	0.198	1.979	0.351	3.157	0.579	4.448	0.785	5.431	0.944

Table 2: Total time required in performing multiple iterations of MILP-based Bound Contraction with varying partitions for both serial and parallel implementation on Reactor Network Design Problem.

No of Regions	Number of Iterations of MILP-based Bound Contractions									
	1		2		3		4		5	
	Serial	Parallel	Serial	Parallel	Serial	Parallel	Serial	Parallel	Serial	Parallel
1	0.408	0.038	0.523	0.069	0.729	0.11	0.882	0.146	1.037	0.173
2	0.757	0.158	1.313	0.303	1.903	0.474	2.508	0.594	3.549	0.718
4	5.011	2.393	11.821	4.620	19.495	7.167	27.344	10.477	34.263	13.380

List of Figures

List of Tables