PARALLEL CONTRACTION OF VARIABLE SPACE FOR GLOBAL SOLUTION OF NLPs

P. K. Polisetty and E. P. Gatzke [1]

Department of Chemical Engineering
University of South Carolina
Columbia, SC, USA 29208

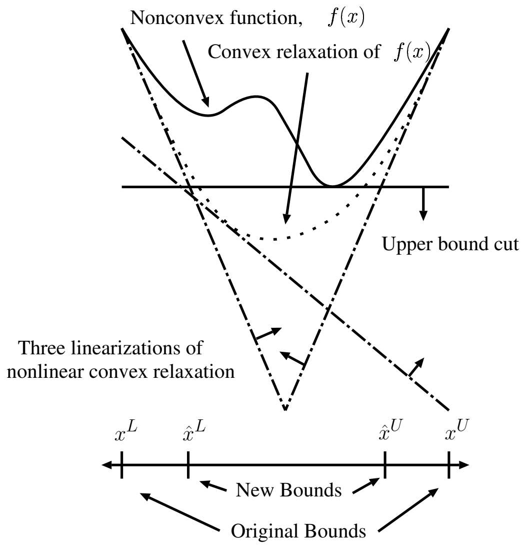
Abstract
This paper presents a parallel algorithm for obtaining global solutions to general mathematical programming problems with nonconvex constraints involving continuous variables. The proposed algorithm implements an optimization based bound tightening technique (Smith [1996], Ryoo and Sahinidis [1995], Adjiman et al. [2000]) in parallel on the root node of the branch-and-bound tree structure. Upon obtaining the convex relaxation of the original nonconvex nonlinear problem, it may be possible to tighten the bounds on any variable by solving two convex optimization problems. The proposed algorithm is implemented on a Heat Exchanger Network Synthesis problem (Yee and Grossmann [1990]). Computational results demonstrate that variable contraction at the root node can result in a substantial decrease in the number of partitions created when performing the branch-and-reduce global optimization algorithm. Additionally, the solution time may decrease significantly when this variable contraction is done in parallel. The proposed parallel algorithm is implemented in multiple ways for comparison.

Keywords:
Global optimization, spatial branch-and-bound, branch-and-reduce, optimization based bound tightening

INTRODUCTION

Systems engineering problems often require the solution to problems which involve hundreds or even thousands of variables. Typically these problems take the form of a large nonlinear program. These problems may take a long time to converge to the global solution when solved using the existing global optimization techniques. Deterministic global optimization techniques have been developed to solve difficult optimization problems. Some of these methods include spatial branch-and-bound (Falk and Soland [1969], Quesada and Grossmann [1993], Horst and Tuy [1993]), outer-approximation (Fletcher and Leyffer [1994]), and branch-and-reduce (Ryoo and Sahinidis [1995]). This paper presents a new parallel global optimization technique for solution of general nonconvex, nonlinear problems. The proposed algorithm makes use of both spatial branch-and-bound and branch-and-reduce algorithmic techniques in determining the global solution. In some examples, this algorithm proves to be computationally faster than previous methods. This paper is organized as follows: Section 2 describes the parallel contraction algorithm. Section 3 discusses some algorithmic design issues. Section 4 presents the computational results with parallel contraction algorithm implemented at root node of a branch-and-reduce global optimization algorithm. Finally conclusions are presented in Section 5.

[1] Author to whom correspondence should be addressed: gatzke@sc.edu



Nonconvex function, $f(x)$
Convex relaxation of $f(x)$
Upper bound cut
Three linearizations of nonlinear convex relaxation
x^L
$\hat{x}^L$
$\hat{x}^U$
x^U
New Bounds
Original Bounds

Figure 1. Optimization Based Bound Tightening showing variable contraction

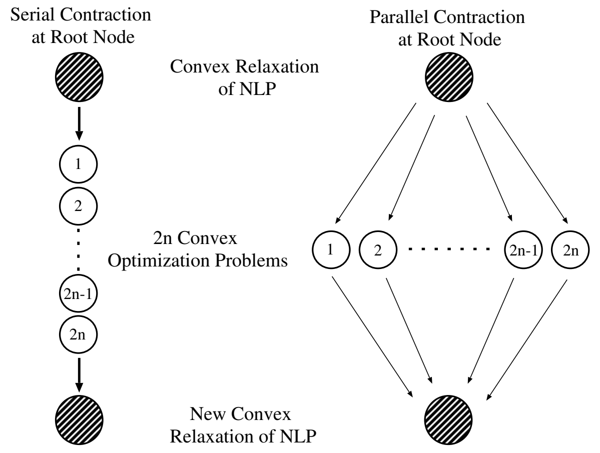
Branch-and-bound techniques typically rely on generating lower and upper bounds for a given partition in a search tree. Lagrangian and feasibility based range reduction tests (Ryoo and Sahinidis [1995]) applied to a partition helps derive tighter variable bounds by eliminating infeasible and suboptimal parts of the feasible region. In addition to those techniques, optimization based bound tightening (Smith [1996], Ryoo and Sahinidis [1995], Adjiman et al. [2000]) shown in Figure 1 can be applied on the root node of the branch-and-bound tree structure. Numerous methods (Adjiman et al. [1998], McCormick [1976], Tawarmalani and Sahinidis [2002], Gatzke et al. [2002]) have been proposed to construct convex relaxations to the original nonconvex problem. Once the convex relaxation for the original nonconvex nonlinear problem is obtained, it is then possible to tighten the bounds on any original variable x_i by solving two convex optimization problems formulated as :

$$\begin{aligned} \min_{\mathbf{x}} \quad & \pm x_i \\ s.t. \quad \hat{\mathbf{f}}(\mathbf{x}) & \leq ubd \\ \hat{\mathbf{g}}(\mathbf{x}) & \leq 0 \\ \mathbf{x}^l \leq \mathbf{x} & \leq \mathbf{x}^u \end{aligned} \tag{1}$$

where $\hat{f}$, $\hat{g}$ are relaxed objective function and constraints and ubd is the current upper bound on the problem.

If this variable contraction is applied at the root node, the number of partitions created when performing the branch-and-reduce algorithm may significantly decrease, aiding in quick convergence of the algorithm. This bound tightening technique can be applied on each of the n original variables in the problem, which requires the solution of $2n$ convex problems as shown in Figure 2 . 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 Eq (1) are all decoupled from one another.

The parallel algorithm is implemented using a *master/slave* paradigm, where the master keeps track of all the calculations and ensures that the slaves are doing useful work. The standard *Message Passing Interface (MPI)* (Forum [1997]) is used for communication between the processors. The master processor sends the original variable bounds of the particular variable to be contracted to the slave processor. The slave then solves the two convex optimization problems given by Eq (1) and returns the new variable bounds to the master. Multiple passes of these variables for contraction may result in generating extremely tight bounds. This algorithm is implemented on a Beowulf style computer using CPLEX8.0 (ILOG [2002]) for solution of Linear Programming (LP) problems. The machine has 32 nodes, each with a single 933 MHz Pentium-3, 1Gbyte of memory, and 15 Gbytes of disk space.



Serial Contraction at Root Node
Parallel Contraction at Root Node
Convex Relaxation of NLP
1
2
2n-1
2n
2n Convex Optimization Problems
New Convex Relaxation of NLP

Figure 2. Optimization Based Bound Tightening Algorithm (Serial and Parallel)

PARALLEL ALGORITHM FOR CONTRACTION OF VARIABLES

The proposed bound tightening technique when applied to the convex relaxation of the original nonconvex NLP can significantly improve the lower bound on the problem before any global optimization algorithm is used for obtaining the solution. This tightening procedure can take a significant amount of time when solved using a single processor. The key contribution is the implementation of this procedure using multiple processors to minimize the overall time required to solve the requisite $2n$ convex problems. Parallel optimization based bounds tightening can be clearly explained by the following pseudo code where n is total number of variables in the problem, p is the number of optimization based passes, i is the loop iteration counter.

```
Search for upper bound to nonconvex NLP
Create convex relaxation of the nonconvex NLP
Tighten the bounds and generate new lower bound
While {($i < p$) and (lbd for obj function $<$ ubd for obj function)}
    While {(processors are running) and
                    (variable to be contracted $< n$)}
        While {(processors are available) and
                    (variable to be contracted $< n$)}
            send the current lbd and ubd of the variable
            to a free processor
        End
        If {processors are running}
            receive the new lbd and ubd of the variable
            from the slave processors
        End
    End
    Tighten the bounds and generate lbd for
    the problem using new variable bounds
End
```

Upon obtaining tighter bounds for the variables using the newly proposed parallel algorithm, a serial or parallel branch-and-reduce algorithm can then be used to search for the global solution. 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. The ratios of the difference in the current bounds and the difference in the original bounds for all of the variables are computed. Those particular variables with the worst (largest) ratios are then selected for variable contraction.

ALGORITHMIC DESIGN ISSUES

There are many issues to consider with parallel programming implementation. Even a perfect parallel program may not always result in 100 percent efficiency because of various algorithmic design issues. A common problem at the beginning of the parallel algorithm is the availability of more processors than work units. The work load should be balanced among all processors making sure some processors are not idle. The time required to coordinate the parallel tasks is another important issue to be considered. Some common problems encountered are: initializing all slave processors, loading the problem to be solved onto the slave processors, the time required to actually carry out the task, time required to terminate the started task, and synchronization time. In many cases overhead due to synchronization may consume wall clock time. Significant amount of time is required for the communication between the master and the slave processors. All these issues can cause the parallel programs not to result in 100 percent efficiency.

COMPUTATIONAL RESULTS

The proposed algorithm has been tested on several optimization problems. For this paper the computational results are only shown for a Heat Exchanger Network Synthesis problem. For this problem, the objective is to minimize the annual cost of the complete heat exchanger network by finding which hot stream/cold stream, hot stream/cold utility, and cold stream/hot utility are potentially matched along with what heat loads are to be assigned to each heat exchanger. Binary variables are introduced to represent the existence of each heat exchanger. The heat exchanger calculation introduces highly nonlinear terms in the objective function. The introduction of binary variables and the nonlinearities make the given problem a Mixed Integer Nonlinear program (MINLP). For the purpose of this paper, the binary variables are fixed corresponding to the global solution of the MINLP. By fixing the binary variables, the problem is converted to a general nonconvex NLP.

The proposed algorithm is implemented several ways. In case A, the optimization based bound tightening technique is implemented sequentially for contracting the variable bounds at the root node. After obtaining tighter bounds, a serial global branch-and-reduce approach is used for finding the global solution. The solution times for multiple optimization based variable bound contractions and the lower and upper bounds of the objective function are shown in Figure 3. It should be noted that gap between the upper and lower bounds on the objective function decreased significantly prior to performing the global search for the solution. Note that the total solution time increases with the number of contractions. This computational burden will be decreased by the use of the newly proposed parallel contraction algorithm.

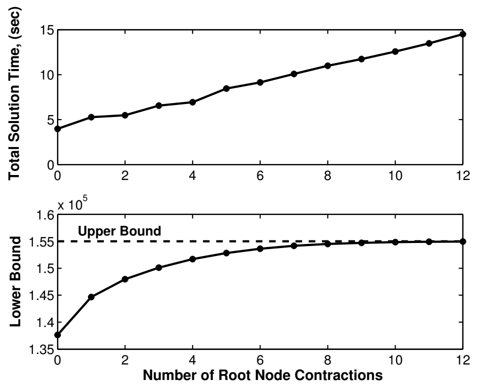


Figure 3. Solution times and the lower bounds on the objective function shown for multiple passes of serial optimization based bound tightening used with a serial branch-and-reduce algorithm.

In case B, the parallel technique is used for bounds tightening and again the serial global branch-and-reduce method is used. In both cases, as expected, equally tight variables bounds are obtained. The solution time for multiple passes of contraction levels using different number of processors are shown in Figure 4. As expected, the solution times decreased significantly as the number of processors increase. The average solution time in this case remained around $4-6$ seconds when more than 2 processors are used, compared to

$4 - 14$ seconds when used serial contraction . It is observed that for a problem of this size, the solution time remained almost constant after using more than 12 processors. This is due to the synchronization and communication overhead between the large number of processors as discussed earlier.

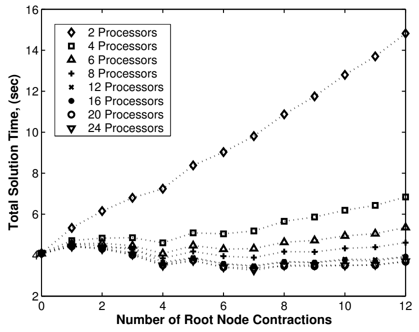


Figure 4. Solution times for multiple passes of parallel optimization based bound tightening used with a serial branch-and-reduce algorithm for a problem involving 65 variables.

A parallel global branch-and-reduce algorithm was developed in order to decrease the solution time even more significantly. The solution times using a different number of processors without using any optimization based bound tightening methods were found as: 6 seconds for one processor, 4.2 seconds for two processors, 2.4 seconds for four processors, and 2.1 seconds for six or more processors.The solution time decreased for a increase in the number of processors but remained nearly constant after a particular point because of communication overhead.

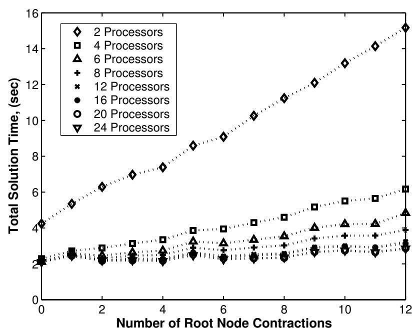


Figure 5. Solution times for multiple, parallel passes of optimization based bound tightening used with parallel branch and reduce algorithm

In case C, the proposed parallel contraction technique is used along with the parallel global branch-and-reduce algorithm. The solution times using multiple processors, both for performing multiple passes of variable contraction and for parallel branch-and-reduce algorithm, are shown in Figure 5. A significant decrease in solution times is observed as compared to the case of using serial branch-and-reduce algorithm. The average solve time remained around $2 - 3$ seconds for this case. Again, to further decrease solution times, only those variables which contribute the most to the gap between the original nonconvex problem and the relaxed problem can be selected for variable bound contraction.

CONCLUSIONS

A new parallel optimization based bound tightening algorithm for solving a general class of nonconvex nonlinear problems has been proposed. The proposed parallel algorithm is implemented in several ways using serial and parallel branch-and-reduce algorithms. Computational results demonstrate that the solve time for determining the global solution decreased significantly when the parallel contraction technique is implemented on the convex relaxation of the original nonconvex NLP prior to performing the global search for the solution.

Acknowledgments

The authors would like to acknowledge financial support from the National Science Foundation Early Career Development grant CTS-038663.

REFERENCES

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

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

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

R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66:327–349, 1994.

Message Passing Interface Forum. MPI-2: Extensions to the Message-Passing Interface. Technical report, University of Tennessee, Knoxville, TN, 1997.

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

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

ILOG. *ILOG CPLEX 8.0: User's Manual*. Mountain View, CA, July 2002.

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

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

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

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

Mohit Tawarmalani and Nikoloas V. Sahinidis. Global Optimization of Mixed-Integer Nonlinear Programs: A Theoretical and Computational Study. *Mathematical Programming*, October 2002.

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