Korean J. of Chem. Eng., **14**(4), 270-276 (1997)

GLOBAL OPTIMIZATION OF CHEMICAL PROCESSES USING THE INTERVAL ANALYSIS

Ju Ran Han, Vasilios Manousiouthakis* and Soo Hyoung Choi†

School of Chemical Engineering and Technology, Chonbuk National University, Chonju 561-756, S. Korea
*Department of Chemical Engineering, University of California, Los Angeles, CA 90095, U.S.A.
(Received 4 March 1997 • accepted 9 June 1997)

Abstract – Optimization of chemical processes often leads to nonlinear programming problems that are nonconvex. Such problems may possess many local optima, whose objective function values vary significantly from one to another. Thus identifying the global optimum is an important, albeit difficult, endeavor. A deterministic algorithm based on interval analysis branch and bound is proposed in this paper to be suitable for global optimization of chemical processes.

Key words: *Global Optimization, Chemical Processes, Interval Analysis, Branch and Bound*

INTRODUCTION

Most chemical process optimization problems are nonlinear programming (NLP) problems in the following form.

$$\min f(\mathbf{x}) \qquad \text{(P)}$$

subject to

$$\mathbf{g}(\mathbf{x}) \le \mathbf{0}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

where $\mathbf{x} \in R^n$, $f : R^n \rightarrow R$, $\mathbf{g} : R^n \rightarrow R^l$, and $\mathbf{h} : R^n \rightarrow R^m$. If the objective function and the feasible region are convex, *e.g.*, when f and **g** are convex and **h** is linear, the problem is called a convex problem, which has only one local minimum that is the global minimum. Most of chemical process optimization problems, however, have a nonconvex feasible region because of nonlinear equality constraints. Therefore, they are nonconvex, and in many cases, have multiple local optima. However, most optimization techniques currently used are local methods, which easily crash, and at best, find only one local optimum. The goal of this study is to develop a method for global optimization of chemical processes.

Most global optimization algorithms belong to one of the two categories: (1) Stochastic approach and (2) Deterministic approach. Algorithms such as simulated annealing [Kirkpatrick et al., 1983] and genetic algorithm [Goldberg, 1989] are based on the stochastic approach. These algorithms aim high probability of finding the global optimum, and do not guarantee the finite ε-convergence (convergence to the global optimum in finite computation steps for a given finite error tolerance), or the global optimality of the obtained solution. Algorithms that use the deterministic approach such as branch and bound [Soland, 1971] guarantee the finite ε-convergence and the global optimality of the obtained solution.

†To whom correspondence should be addressed.

The deterministic approach to global optimization of chemical processes has actively been studied since the 1980's. Most of the proposed algorithms are based on one of the following methods.

1. Generalized Benders Decomposition (GBD) [Geoffrion, 1972]

This algorithm iterates between a primal and a master which give upper and lower bounds respectively for the global optimum. Based on this approach, Duran and Grossmann [1986] proposed an Outer Approximation algorithm for a particular class of Mixed Integer Nonlinear Programs (MINLP). Floudas et al. [1989] proposed Global Optimum Search (GOS) algorithm for general problems, which was considered very efficient, but Bagajewicz and Manousiouthakis [1991] indicated an error in this algorithm. Floudas and Visweswaran [1990] corrected the error, proposing Global Optimization Algorithm (GOP), but the computation time for convergence greatly increased.

2. Underestimator Branch and Bound

Soland [1971] presented an algorithm of the branch and bound type which solves a separable nonconvex problem by solution of a sequence of convex subproblems. Each convex subproblem, if feasible, gives a lower bound on the global minimum in its domain. Subproblems are discarded if they are infeasible or their lower bounds are higher than the upper bound which is set by the value of the objective function at a feasible point. Feasible subproblems are divided into more subproblems, and the algorithm continues until the lower bound converges to the upper bound. Based on this approach, Ryoo and Sahinidis [1995] and Adjiman et al. [1996] proposed algorithms for chemical process optimization.

3. Interval Analysis Branch and Bound

Ratschek and Rokne [1988, 1991] presented a branch and bound algorithm which uses interval analysis to calculate the lower bounds. Unlike the above mentioned algorithms, their algorithm can be applied to any type of problems, not requiring any problem transformation. This algorithm was modifi-

ed and applied to chemical process optimization by Vaidyanathan and El-Halwagi [1994].

Global optimization of a nonconvex NLP is an NP (Non-Polynomially)-hard problem, and thus, when a deterministic algorithm is used, the computation time drastically (generally, exponentially) increases with the size of the problem. This is the reason that all the example problems in the papers that propose deterministic algorithms are extremely small. Chemical process optimization problems are generally highly nonlinear, nonconvex, and very large. Therefore, the conventional deterministic algorithms are not suitable for chemical process optimization. Some people suggest that the stochastic approach is practically the only way to global optimization of large problems. However, as the computer hardware is rapidly being improved these days, it is still justifiable to study on the deterministic approach.

The deterministic approach is frequently based on the branch and bound technique. The problem is how to minimize the number of branching variables. A positive aspect of the branch and bound algorithm proposed by Soland [1971] is that branching is required only for the variables that cause nonconvexity of the problem. Let us refer to such variables as nonconvex variables. Unfortunately, however, in most chemical process optimization problems, all variables appear in nonlinear equality constraints, and thus, all variables are nonconvex variables. Therefore, the only method to reduce the number of branching variables for the conventional branch and bound technique is to reduce the total number of variables in the problem based on the symbolic solution of the equality constraints. However, it is not easy to symbolically solve a large system of highly nonlinear equations. This paper presents a method for reducing the number of branching variables using the equality constraints without any symbolic manipulations. Too many nonlinear equality constraints are generally a trouble, but in this work, they are the last hope.

THE PROPOSED AlGORITHM

The proposed algorithm is based on Algorithm 2 of Ratschek and Rokne [1988]. Their algorithm always treats all variables as the branching variables even though there exist many equality constraints. Other interval analysis algorithms such as proposed by Vaidyanathan and El-Halwagi [1994] also branch all variables. The algorithm proposed in this paper, however, exploits the equality constraints so that the intervals of some variables can be determined by those of other variables, and thus the number of branching variables can be reduced. Consider the following optimization problem in which variables are divided into two groups.

$$\min_{\mathbf{x},\,\mathbf{y}} f(\mathbf{x}, \mathbf{y})$$

subject to

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) \le \mathbf{0}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$$

$$\mathbf{x} \in X := \{\mathbf{x} \mid \mathbf{x} \in R^{n_x},\ -\infty < x_j^L \le x_j \le x_j^U < +\infty,\ j = 1, \cdots, n_x\}$$

$$\mathbf{y} \in Y := \{\mathbf{y} \mid \mathbf{y} \in R^{n_y},\ -\infty < y_j^L \le y_j \le y_j^U < +\infty,\ j = 1, \cdots, n_y\}$$

where **f**, **g**, and **h** are continuous functions. The variables **x** and **y** are selected in such a way that the intervals of variables **y** denoted by Y, can be determined from the intervals of variables **x** denoted by X, using the constraints in the above problem.

The proposed algorithm is as follows.

1. Interval Analysis Branch and Bound Algorithm

1-1. Step 0. Initialization

(1) Define $X^1 := X$.
(2) Initialize the set of unsolved subproblem indices $B \leftarrow \{1\}$ and the set of solved subproblem indices $D \leftarrow \emptyset$.
(3) Set the upper bound on the global minimum $U \leftarrow +\infty$ or $f(x^o, y^o)$, where (x^o, y^o) is any feasible point.

1-2. Step 1. Interval Analysis and Bounding. For all $k \in B$,

(1) Remove k from B, and enter k into D.
(2) Determine Y^k corresponding to X^k.
(3) Set $Y^k \leftarrow Y^k \cap Y$.
(4) If $Y^k = \emptyset$ or lb $f(X^k, Y^k) >$ U or lb $g_i(X^k, Y^k) > \varepsilon_f$ or lb $h_i(X^k, Y^k) > \varepsilon_f$ or ub $h_i(X^k, Y^k) < -\varepsilon_f$ for some i, then remove k from D.
(5) Otherwise,
  (a) Set $x^k :=$ mid X^k.
  (b) Determine y^k corresponding to x^k.
  (c) If $y^k \in Y^k$ and $g_i(x^k, y^k) \le \varepsilon_f$ and $h_i(x^k, y^k) \in [-\varepsilon_f, +\varepsilon_f]$ for all i, then
    (i) Update $U \leftarrow \min(U, f(x^k, y^k))$.
    (ii) Remove all r from D such that lb $f(X^r, Y^r) >$ U.

1-3. Step 2. Convergence Test and Branching

(1) If $D = \emptyset$, terminate as the problem is infeasible.
(2) Otherwise,
  (a) Select $k \in D$ such that lb $f(X^k, Y^k) = \min_{r \in D}$ lb $f(X^r, Y^r)$.
  (b) If $y^k \in Y^k$ and $f(x^k, y^k) -$ lb $f(X^k, Y^k) \le \varepsilon_o$ and $g_i(x^k, y^k) \le \varepsilon_f$ and $h_i(x^k, y^k) \in [-\varepsilon_f, +\varepsilon_f]$ for all i, then terminate with (x^k, y^k) as a global solution.
  (c) Otherwise,
    (i) Select varable x_j which has the maximum length of interval in X^k.
    (ii) Bisect X^k normal to coordinate x_j, getting X^p and X^q such that $X^k = X^p \cup X^q$, where $p, q \notin D$.
    (iii) Remove k from D, and enter p and q into B.
    (iv) Go to Step 1.

The operators lb and ub in the above algorithm represent lower bound and upper bound respectively, and they can be calculated by any interval arithmetic method such as natural inclusion, centered-form inclusion, *etc.* [Ratschek and Rokne, 1988]. The natural inclusion has been used in this work, which is defined as follows.

$$[a, b] + [c, d] = [a + c, b + d]$$

$$[a, b] - [c, d] = [a - d, b - c]$$

$$[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$[a, b]/[c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)] \text{ if } 0 \notin [c, d]$$

The parameters ε_f and ε_o in the above algorithm are the feasibility tolerance and the optimality tolerance respectively. The values of ε_f and ε_o should be set to a moderately small positive number based on the numerical scale of the problem. The value of ε_f in Step 1(4), however, can be set to zero if the machine interval arithmetic [Ratschek and Rokne, 1988] is used to calculate the lower and upper bounds of the constraint functions.

The proposed algorithm applies branching (bisection) to the **x** variables only. The finite ε-convergence of the proposed algorithm is guaranteed if we select **x** and **y** such that the following property is satisfied.

Property (Y) : For any $X^k \subset X$, as $w(X^k) := \text{ub } X^k - \text{lb } X^k \to 0$, $w(Y^k) \to 0$.

This indicates that, for anywhere in the search space, if the intervals of the **x** variables approach zero, then the intervals of the **y** variables also approach zero. If this property is satisfied, we can safely apply branching to the **x** variables only. The convergence is guaranteed since the proof of convergence presented by Ratschek and Rokne [1988] still applies to our case. If the above property is not satisfied, the convergence is not guaranteed. Nevertheless, however, if only the procedure converges, then the global optimality of the obtained solution is still guaranteed. This is because the lower bound of the objective function is always valid with or without Property (Y).

The simplest case in which the intervals of **y** can be determined from those of **x** arises when the following type of equality constraints are available.

$$\mathbf{A(x)y = b(x)} \tag{1}$$

where **A** is an $n_y \times n_y$ matrix, **b** is an n_y-dimensional column vector, and their elements are constants or functions of **x** only. If the coefficient matrix **A** includes a singular matrix for given intervals for **x**, *i.e.*, $0 \in$ [lb det **A**(X), ub det **A** (X)], then Property (Y) is not satisfied. However, numerical experiments show that the proposed algorithm can still converge in such a case also. As mentioned before, if it converges, the global optimality of the obtained solution is guaranteed. If it fails to converge, we have to try a different set of the **y** variables.

Chemical process optimization problems generally include many equality constraints, and many of them are of the type of Eq. (1). Furthermore, other types of equality constraints and inequality constraints can also be used to calculate the intervals of **y** from those of **x**. Therefore, the proposed algorithm is considered suitable for chemical process optimization.

CASE STUDIES

The following examples were used in this work for performance evaluation.

(1) Design of a three stage process system with recycle [Stephanoploulos and Westerberg, 1975]
(2) Reactor network design [Manousiouthakis and Sourlas, 1992]
(3) Heat exchanger network synthesis [Floudas and Ciric, 1989]

All local optima of these problems were found by an exhaustive search using local optimizer GINO [Winston, 1995], and the interval analysis algorithm proposed in this paper was programmed in FORTRAN and implemented on a Pentium 100 MHz computer.

1. Example 1: Design of a Three Stage Process System with Recycle

This is an example problem of Stephanopoulos and Westerberg [1975] which minimizes the capital cost of a three stage process with recycle as shown in Fig. 1. The problem formulation is as follows.

$$\min\ x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6$$

subject to

$$-3x_1 + x_2 - 3x_4 = 0$$

$$-2x_2 + x_3 - 2x_5 = 0$$

$$4x_4 - x_6 = 0$$

$$x_1 + 2x_4 \le 4$$

$$x_2 + x_5 \le 4$$

$$x_3 + x_6 \le 6$$

$$\mathbf{0} \le \mathbf{x} \le (3, 4, 4, 2, 2, 6)$$

This problem has three local minima including the global minimum as listed in Table 1. Vaidyanathan and El-Halwagi [1994] indicated that the GBD algorithm of Floudas and Pardalos [1990] converged to a wrong solution, and reported that their interval analysis algorithm with the tolerance on the width of the solution box ε=0.001 and the accuracy of the objective function inclusion δ=0.01 located the global minimum consuming 436.4s on Sun SPARCstation 10. Ryoo and Sahinidis [1995] reported that their underestimator branch and bound algorithm with the optimality tolerance $\varepsilon=10^{-6}$ located the global minimum by solution of only one subproblem,

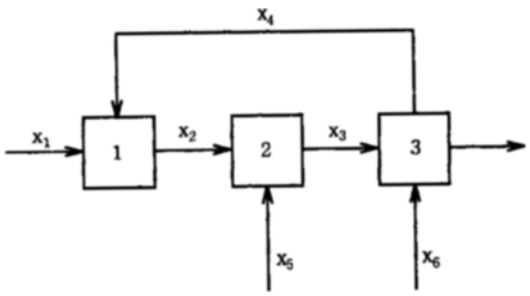

Fig. 1. A three stage process system with recycle.

Table 1. Computation results of example 1

Algorithm	Class	Number of branching variables	Convergence criterion	Objective function value	Number of subproblems	Computing time
Floudas and Pardalos [1990]	GBD	0	–	−11.96 (wrong solution)	–	–
Vaidyanathan and El-Halwagi [1994]	Interval analysis	6	$\varepsilon=10^{-3}$ $\delta=10^{-2}$	−13.402	–	436.4s on Sun SPARCstation 10
Ryoo and Sahinidis [1995]	Underestimator	6	$\varepsilon=10^{-6}$	−13.401904	1	0.5s on Sun SPARCstation 2
This work	Interval analysis	3	$\varepsilon_f=10^{-4}$ $\varepsilon_o=10^{-4}$	−13.40187	769	0.11s on Pentium 100 MHz

Local 1: f= −4.258899 at **x**=(0, 0, 4, 0, 2, 0)
Local 2: f= −12.507887 at **x**=(0, 1.8, 3.6, 0.6, 0, 2.4)
Local 3: f= −13.401904 at **x**=(0.166667, 2, 4, 0.5, 0, 2)

and the computation time was 0.5s on Sun SPARCstation 2.

This problem has three equality constraints, and using these, the intervals of variables x_2, x_3, and x_6 can be determined by those of x_1, x_4, and x_5. Exploiting this property, *i.e.*, using x_1, x_4, and x_5 as the branching variables (the **x** variables in the algorithm), the proposed interval analysis branch and bound algorithm with $\varepsilon_f=10^{-4}$ and $\varepsilon_o=10^{-4}$ required 769 subproblems and computing time of 0.11s on Pentium 100 MHz to find the global optimum. The results are summarized in Table 1.

2. Example 2: Reactor Network Design

This is an example problem of Manousiouthakis and Sourlas [1992] which optimizes a two reactor system as shown in Fig. 2. The objective is to maximize the concentration of component B in the output stream of reactor 2, satisfying a given constraint on the capital cost. The problem formulation is as follows.

$$\min -x_4$$

subject to

$$x_1 - 1 + k_1 x_1 x_5 = 0$$

$$x_2 - x_1 + k_2 x_2 x_6 = 0$$

$$x_3 + x_1 - 1 + k_3 x_3 x_5 = 0$$

$$x_4 - x_3 + x_2 - x_1 + k_4 x_4 x_6 = 0$$

$$x_5^{0.5} + x_6^{0.5} \le 4$$

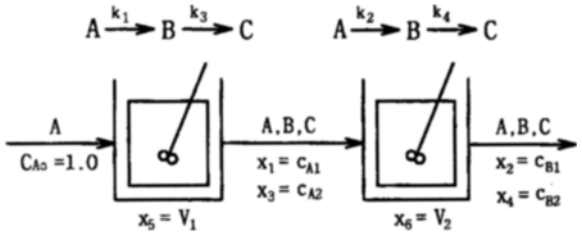

Fig. 2. A two reactor system.

$$\mathbf{0} \le \mathbf{x} \le (1, 1, 1, 1, 16, 16)$$

where $k_1=0.09755988$, $k_2=0.99\ k_1$, $k_3=0.0391908$, and $k_4=0.90\ k_3$. This problem has three local minima as listed in Table 2. Local 1 corresponds to using reactor 1 only, local 2 corresponds to using reactor 2 only, and local 3, which is the global optimum, corresponds to using both. Note that the difference in the objective function value between local 2 (−0.388102) and local 3 (−0.388812) is extremely small. Therefore, this problem will check the robustness of the tested algorithm. The results are summarized in Table 2.

Ryoo and Sahinidis [1995] reported that their algorithm with the optimality tolerance $\varepsilon=10^{-6}$ located the global optimum by solution of 179 subproblems, and the computation time was 21s on Sun SPARCstation 2. In this work, the variables x_5 and x_6 were selected as the **x** variables, and x_1, x_2, x_3, and x_4 as the **y** variables. The implementation of the proposed algorithm with $\varepsilon_f=10^{-4}$ and $\varepsilon_o=10^{-4}$ required 20,651 subproblems and computing time of 11.87s on Pentium 100 MHz to find

Table 2. Computation results of example 2

Algorithm	Class	Number of branching variables	Convergence criterion	Objective function value	Number of subproblems	Computing time
Ryoo and Sahinidis [1995]	Underestimator	6	$\varepsilon=10^{-6}$	−0.388812	179	21s on Sun SPARCstation 2
This work	Interval analysis	2	$\varepsilon_f=10^{-4}$ $\varepsilon_o=10^{-4}$	−0.38881	20,651	11.87s on Pentium 100 MHz
			$\varepsilon_f=10^{-4}$ $\varepsilon_o=10^{-3}$	−0.3888	6,281	1.81s on Pentium 100 MHz

Local 1: f= −0.374617 at **x**=(0.390479, 0.390479, 0.374617, 0.374617, 16, 0)
Local 2: f= −0.388102 at **x**=(1, 0.392874, 0, 0.388102, 0, 16)
Local 3: f= −0.388812 at **x**=(0.771462, 0.516997, 0.204234, 0.388812, 3.036504, 5.096052)

the global optimum. The algorithm also located the global optimum with $\varepsilon_f=10^{-4}$ and $\varepsilon_o=10^{-3}$, and in this case, the number of subproblem was 6,281, and the computing time was 1.81s.

3. Example 3: Heat Exchanger Network Synthesis

This is an example problem of Floudas and Ciric [1989] which determines the optimal configuration of two heat exchangers for two hot streams and one cold stream. The superstructure of the heat exchanger network is as shown in Fig. 3. The problem formulation for minimizing the capital cost is as follows.

$$\min \; 1200\left[\frac{800}{2.5\left(\frac{2}{3}\sqrt{(320-t_2)(300-t_1)}+\frac{(320-t_2)+(300-t_1)}{6}\right)}\right]^{0.6} +1200\left[\frac{1000}{0.2\left(\frac{2}{3}\sqrt{(340-t_4)(300-t_3)}+\frac{(340-t_4)+(300-t_3)}{6}\right)}\right]^{0.6}$$

subject to

$$f_1+f_2=10 \tag{2}$$

$$f_1+f_6=f_3 \tag{3}$$

$$f_2+f_5=f_4 \tag{4}$$

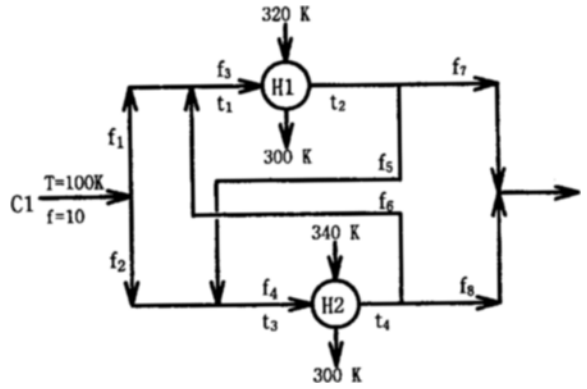

Fig. 3. A heat exchanger network superstructure.

$$f_5+f_7=f_3 \tag{5}$$

$$f_6+f_8=f_4 \tag{6}$$

$$100f_1+t_4f_6=t_1f_3 \tag{7}$$

$$100f_2+t_2f_5=t_3f_4 \tag{8}$$

$$f_3(t_2-t_1)=800$$

$$f_4(t_4-t_3)=1000$$

$$\mathbf{0\le f\le 10}$$

$$\mathbf{100\le t\le}(290,\ 310,\ 290,\ 330)$$

In order to improve the numerical stability of this problem, all the constraints were scaled using new variables defined by the original variables divided by their upper limits, and adjusting the coefficients of the equations in terms of the new variables to the order of magnitude of one. The objective function was not scaled, because the optimality tolerance can be adjusted independent of the feasibility tolerance in the proposed algorithm. This problem has 3 local minima including the global minimum as shown in Table 3. Local 1 corresponds to the parallel configuration, local 2 corresponds to the 1-2 serial configuration, and local 3, which is the global optimum, corresponds to the 2-1 serial configuration.

The above problem has 12 variables and 9 equality constraints, and thus 3 degrees of freedom. However, it was impossible in this work to find 3 **x** variables and 11 **y** variables with which the algorithm converges. While many alternatives exist, the following 6 variables were selected as the **x** variables: f_3, f_4, t_1, t_2, t_3, and t_4. The intervals of the other 6 variables (the **y** variables) are determined as tight as possible taking advantage of all of the equality and inequality constraints whenever possible. For example, when the intervals of f_3, f_4, t_1, t_2, t_3, and t_4 are known, the intervals of f_1, f_2, f_5, and f_6 can be determined by Eqs. (3), (4), (7), and (8), and those of f_7 and f_8 by Eqs. (5) and (6). The intervals of f_1 and f_2 can further be reduced by Eq. (2) as follows.

$$\max(\mathrm{lb}\,f_1,\ 1-\mathrm{ub}\,f_2)\le f_1\le \min(\mathrm{ub}\,f_1,\ 1-\mathrm{lb}\,f_2)$$

$$\max(\mathrm{lb}\,f_2,\ 1-\mathrm{ub}\,f_1)\le f_2\le \min(\mathrm{ub}\,f_2,\ 1-\mathrm{lb}\,f_1)$$

This process can decrease the extent of underestimation of the objective function, and thus greatly improves the efficiency of the branch and bound procedure without loss of glo-

Table 3. Computation results of example 3

Algorithm	Class	Number of branching variables	Convergence criterion	Objective function value	Number of subproblems	Computing time
Ryoo and Sahinidis [1995]	Underestimator	8	$\varepsilon=10^{-6}$	12292.467132	1	2.2s on Sun SPARCstation 2
This work	Interval Analysis	6	$\varepsilon_f=10^{-4}$ $\varepsilon_o=10^{+2}$	12290.	57,041	34.70s on Pentium 100 MHz
			$\varepsilon_f=10^{-2}$ $\varepsilon_o=10^{+3}$	12200.	4,591	2.64s on Pentium 100 MHz

Local 1: obj=24,172 at **f**=(3.8095, 6.1905, 3.8095, 6.1905, 0, 0, 3.8095, 6.1905), **t**=(100, 310, 100, 261)
Local 2: obj=22,970 at **f**=(10, 0, 10, 10, 10, 0, 0, 10), **t**=(100, 180, 180, 280)
Local 3: obj=12,292 at **f**=(0, 10, 10, 10, 0, 10, 10, 0), **t**=(200, 280, 100, 200)

bal optimality.

Ryoo and Sahinidis [1995] reported that their algorithm with the optimality tolerance $\varepsilon=10^{-6}$ solved one subproblem consuming 2.2s on Sun SPARCstation 2 to find the global minimum. The proposed algorithm with $\varepsilon_f=10^{-4}$ and $\varepsilon_o=10^{+2}$ required 57,041 subproblems and computing time of 34.70s on Pentium 100 MHz. When more relaxed tolerances of $\varepsilon_f=10^{-2}$ and $\varepsilon_o=10^{+3}$ were used, the number of subproblems was 4,591, and the computing time was 2.64s. The results are summarized in Table 3.

DISCUSSION AND SUGGESTIONS

The underestimator branch and bound algorithms such as the algorithm of Soland [1971] or Ryoo and Sahinidis [1995] use a local optimizer to solve each subproblem. For such algorithms, a very small value for the optimality tolerance can be used without significant loss of efficiency, as shown by Ryoo and Sahinidis [1995]. For the interval analysis branch and bound algorithms such as proposed in this work, however, it is extremely inefficient to rely solely on interval analysis to locate the global optimum within a strict accuracy. This is because the objective function is evaluated at the midpoint of the box for each subproblem. The experimental data in Tables 2 and 3 indicate that significant improvement in the efficiency of the proposed algorithm is obtained by relaxing the feasibility and optimality tolerances. Therefore, a two tier approach to global optimization is suggested, in which the proposed interval analysis algorithm is used with relatively large feasibility and optimality tolerances, and then the result is used as a starting point for a local optimizer so that the accurate global solution is obtained.

A significant advantage of the interval analysis algorithms such as proposed in this paper is that the original problem formulation can directly be used without any modifications unlike the other algorithms such as GBD or the underestimator algorithms which require major reformulation of the original problem to the form that is allowed, sometimes introducing many new substitution variables. Therefore, the proposed algorithm is considered suitable for maintaining the size of the given problem as small as possible.

A disadvantage of the interval analysis algorithms is that the number of required subproblems is much larger than that of the underestimator algorithms. The reason is that the lower bounds on the objective function calculated by interval analysis are not as tight as determined by the underestimator algorithms. However, it is a great advantage that each subproblem can easily be solved without any local solver requiring only small amounts of computer memory and CPU time. Therefore, the proposed algorithm is considered suitable for implementation by the massively parallel computing technique, which is suggested as future work. Futhermore, development of a new method for calculating tighter lower bounds is also suggested.

Most importantly, the efficiency of any deterministic algorithm based on the branch and bound technique mainly depends on the number of branching variables because of the inevitable NP-hardness of the nonconvex NLP problems. The proposed algorithm is considered the most suitable for reducing the number of branching variables. However, the method of reducing the number of branching variables presented in this paper, which uses the equality constraints in the form of Eq. (1), is just the simplest case. Development of methods for using other types of equality and inequality constraints is suggested as future work.

CONCLUSION

An interval analysis branch and bound algorithm has been proposed which is suitable for large nonconvex nonlinear programs which have many equality constraints. Although the nonlinear equality constraints are generally a burden to the other algorithms, this algorithm exploits them to reduce the number of branching variables, which is the actual measure of the size of a problem. Therefore, the algorithm is considered suitable for chemical process optimization problems which are large but have relatively small degrees of freedom. The proposed algorithm should be used with moderately small feasibility and optimality tolerances, and the final solution can be refined by a local optimizer. Case studies indicate that the proposed algorithm can be applied to global optimization of small chemical processes. Application to larger processes will be possible in the future when significant improvement in the computing power is achieved.

ACKNOWLEDGEMENT

This paper was supported by NON DIRECTED RESEARCH FUND, Korea Research Foundation. The authors also gratefully acknowledge partial support from Korea Science and Engineering Foundation through Automation Research Center at Pohang University of Science and Technology.

REFERENCES

Adjiman, C. S., Androulakis, I. P., Maranas, C. D. and Floudas, C. A., "A Global Optimization Method, αBB, for Process Design", *Computers & Chem. Eng.*, **20**, Suppl., S419 (1996).

Bagajewicz, M. and Manousiouthakis, V., "On the Generalized Benders Decomposition", *Computers & Chem. Eng.*, **15** (10), 691 (1991).

Duran, M. A. and Grossmann, I. E., "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs", *Mathematical Programming*, **36**, 307 (1986).

Floudas, C. A., Aggarwal, A. and Ciric, A. R., "Global Optimum Search for Nonconvex NLP and MINLP Problems", *Computers & Chem. Eng.*, **13**(10), 1117 (1989).

Floudas, C. A. and Ciric, A. R., "Strategies for Overcoming Uncertainties in Heat Exchanger Network Synthesis", *Computers & Chem. Eng.*, **13**(10), 1133 (1989).

Floudas, C. A. and Pardalos, P. M., "A Collection of Test Problems for Constrained Global Optimization Algorithms", Lecture Notes in Computer Science, Springer-Verlag, New York, **455**, 29 (1990).

Floudas C. A. and Visweswaran, V., "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs-I. Theory", *Computers & Chem. Eng.*, **14**(12), 1397 (1990).

Geoffrion, A. M., "Generalized Benders Decomposition", *J. Opt. Theory Applic.*, **10**, 237 (1972).

Goldberg, D. E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.

Kirkpatrick, S., Gelatt, C. D. Jr. and Vechhi, M. P., "Optimization by Simulated Annealing", *Science*, **220**, 671 (1983).

Manousiouthakis, V. and Sourlas, D., "A Global Optimization Approach to Rationally Constrained Rational Programming", *Chem. Eng. Comm.*, **115**, 127 (1992).

Ratschek, H. and Rokne, J., "Interval Tools for Global Optimization", *Computers Math. Applic.*, **21**(6/7), 41 (1991).

Ratschek, H. and Rokne, J., "New Computer Methods for Global Optimization", Ellis Horwood Limited, England, 1988.

Ryoo, H. S. and Sahinidis, N. V., "Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design", *Computers & Chem. Eng.*, **19**(5), 551 (1995).

Soland, R. M., "An Algorithm for Separable Nonconvex Programming Problems II: Nonconvex Constraints", *Management Science*, **17**(11), 759 (1971).

Stephanopoulos, G. and Westerberg, A. W., "The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization", *J. Opt. Theory Applic.*, **15**, 285 (1975).

Vaidyanathan, R. and El-Halwagi, M., "Global Otimization of Nonconvex Nonlinear Programs via Interval Analysis", *Computers & Chem. Eng.*, **18**(10), 889 (1994).

Winston, W. L., "Introduction to Mathematical Programming, Applications and Algorithms", 2nd ed., Duxbury Press, 1995.