Decomposition and Interval Arithmetic Applied to Global Minimization of Polynomial and Rational Functions

PIERRE HANSEN
GERAD and Ecole des Hautes Etudes Commerciales, Département des Méthodes Quantitatives et Systèmes d'Information, Montréal, Québec, H3T 1V6 Canada.

BRIGITTE JAUMARD
GERAD and Ecole Polytechnique de Montréal, Département de Mathématiques Appliquées, Montréal, Québec, H3C 3A7 Canada.

JUNJIE XIONG
Rutgers University, Department of Mathematics, U.S.A.

(Received: 8 April 1992; accepted: 3 January 1993)

Abstract. A recent global optimization algorithm using decomposition (GOP), due to Floudas and Visweswaran, when specialized to the case of polynomial functions is shown to be equivalent to an interval arithmetic global optimization algorithm which applies natural extension to the cord-slope form of Taylor's expansion. Several more efficient variants using other forms of interval arithmetic are explored. Extensions to rational functions are presented. Comparative computational experiences are reported.

Key words. Global optimization, decomposition, interval arithmetic.

Résumé. On montre que l'algorithme récent d'optimisation globale basé sur la décomposition du à Floudas et Visweswaran, lorsqu'on le spécialise au cas de fonctions polynômiales, est équivalent à une méthode d'optimisation globale basée sur l'arithmétique d'intervalles, qui applique l'extension naturelle à la forme de la pente de la corde du développement de Taylor. Plusieurs variantes plus efficaces utilisant d'autres formes de l'arithmétique d'intervalles sont explorées. On propose des extensions au cas des fonctions fractionnaires. On présente des résultats de calcul comparatifs.

Mots-clés. Optimisation globale, décomposition, arithmétique d'intervalles.

1. Introduction

In Floudas and Visweswaran ([5], [6], [13]), a decomposition-based global optimization algorithm (GOP) is proposed for solving constrained nonconvex NLP problems. After introducing new transformation variables if necessary, the original nonconvex problem is expressed as a bilinear (or biconvex) program. Its variables are thus partitioned into two sets. Fixing temporarily one or the other set of variables leads to define primal and relaxed dual subproblems. Solution of these problems provides upper and lower bounds on the global optimum. Recently, this algorithm was specialized to find efficiently the global minimum of univariate polynomial functions over an interval (Visweswaran and Floudas [14]). In this paper, the correspondence between this specialized version of the GOP algorithm and interval arithmetic is explored. It is shown that the application of the GOP algorithm to polynomial functions is equivalent to a version of an interval arithmetic algorithm which uses natural extension of the cord-slope form (defined below) of Taylor's expansion for univariate polynomial functions. Other versions of the interval arithmetic algorithm corresponding to different ways to evaluate the bounds on the cord-slope are investigated. Some of them lead to more efficient algorithms than others. Use of the nested form gives the smallest computing time and of the centered form (or sometimes the nested form) the least number of iterations. Furthermore the interval arithmetic algorithm can be easily extended to find the global optima of univariate rational functions on an interval by applying the cord-slope form on the numerator and the denominator of the fraction.

The paper is organized as follows: the GOP algorithm is presented in Section 2. Some background on interval arithmetic and its application to computing the ranges of polynomial functions is provided in Section 3. The corresponding interval arithmetic algorithm is presented in Section 4. An extension designed to find the global minimum of rational functions is given in Section 5. Computational results are reported in Section 6.

2. The GOP Algorithm for Polynomial Functions

2.1. FORMULATION OF THE PRIMAL AND RELAXED DUAL PROBLEMS

Consider the problem

$$\begin{aligned} \min \quad & F(y) = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n \\ \text{subject to} \quad & y^L \leq y \leq y^U \end{aligned} \tag{1}$$

where $a_0, a_1, \ldots, a_n$ are given real numbers. Introducing new variables $x_i = y^i$, $i = 0, 1, 2, \ldots, n$, Problem (1) can be reformulated as

$$\begin{aligned} \min \quad & f(x, y) = \sum_{i=0}^{n} a_i x_i \\ \text{subject to} \quad & x_0 = 1 , \\ & x_i - x_{i-1} y = 0 , \quad i = 1, 2, \ldots, n , \\ \text{and} \quad & y \in [y^L, y^U] . \end{aligned} \tag{2}$$

For any fixed $\bar{y} \in [y^L, y^U]$, the primal problem is defined as the following Problem $P_{\bar{y}}$:

$$\min_x \quad f(x, \bar{y}) = \sum_{i=0}^{n} a_i x_i$$

$$\text{subject to} \quad x_0 = 1 ,$$
$$x_i - x_{i-1}\bar{y} = 0 , \quad i = 1, 2, \ldots, n . \tag{3}$$

Since all variables x_i are uniquely determined once $\bar{y}$ is fixed, Problem $P_{\bar{y}}$ reduces to a function evaluation; its solution is an upper bound of the global minimum of Problem (2).

Define the function $v(y)$ for $y \in [y^L, y^U]$ to be

$$v(y) = \min_x \quad f(x, y)$$
$$\text{subject to} \quad x_0 = 1$$
$$x_i - x_{x-1}y = 0 , \quad i = 1, 2, \ldots, n \tag{4}$$

then Problem (2) is equivalent to

$$\min \quad v(y) \tag{5}$$

$$\text{subject to} \quad y \in [y^L, y^U] . \tag{6}$$

The minimization Problem in (4) is simply the primal Problem (P_y). For any $y \in [y^L, y^U]$, the strong duality theorem permits $v(y)$ to be written as

$$\sup_\mu \inf_x f(x, y) + \mu^T(x_i - x_{i-1}y) \tag{7}$$

where $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ is a vector of Lagrange multipliers. This leads to the following formulation, equivalent to (2):

$$\min \quad v(y)$$
$$\text{subject to} \quad v(y) \geqslant \min_x L(x, y, \mu) ,$$
$$y \in [y^L, y^U] \tag{8}$$

where $L(x, y, \mu) = f(x, y) + \mu^T(x_i - x_{i-1}y)$ is the Lagrange function.

Moreover, for any fixed $y = \bar{y}$, the Karush–Kuhn–Tucker conditions for the primal Problem $(P_{\bar{y}})$ can be written as

$$\nabla_{x_i} L_{\bar{y}}(x, y, \mu^{\bar{y}}) = a_i + \mu_i^{\bar{y}} - \mu_{i+1}^{\bar{y}}\bar{y} = 0 , \quad \forall i = 1, 2, \ldots, n , \tag{9}$$

with $\mu_0^{\bar{y}} = \mu_{n+1}^{\bar{y}} = 0$. Therefore the Lagrange multipliers can be found by backwards substitution:

$$\mu_n^{\bar{y}} = -a_n$$
$$\mu_{n-1}^{\bar{y}} = \mu_n^{\bar{y}}\bar{y} - a_{n-1} = -a_n\bar{y} - a_{n-1}$$
$$\mu_{n-2}^{\bar{y}} = \mu_{n-1}^{\bar{y}}\bar{y} - a_{n-2} = -a_n\bar{y}^2 - a_{n-1}\bar{y} - a_{n-2}$$
$$\vdots$$

Thus, in general

$$\mu_i^{\bar{y}} = -\sum_{j=i}^{n} a_j \bar{y}^{j-i}, \quad i = 1, 2, \ldots, n \tag{10}$$

and

$$L_{\bar{y}}(x, y, \mu^{\bar{y}}) = \sum_{i=0}^{n} a_i x_i + \sum_{i=1}^{n} \mu_i^{\bar{y}}(x_i - x_{i-1}y) . \tag{11}$$

Separating the terms in x, (11) can be rewritten as

$$L_{\bar{y}}(x, y, \mu^{\bar{y}}) = \sum_{i=0}^{n} (a_i + \mu_i^{\bar{y}} - \mu_{i+1}^{\bar{y}} y)x_i \tag{12}$$

and using (9) again

$$L_{\bar{y}}(x, y, \mu^{\bar{y}}) = a_0 - \mu_1^{\bar{y}}\bar{y} + \sum_{i=1}^{n} \mu_{i+1}^{\bar{y}}(\bar{y} - y)x_i . \tag{13}$$

Therefore the following problem is a relaxation of Problem (8):

$$\begin{aligned} \min \quad & v(y) \\ \text{subject to} \quad & v(y) \geq \min_x L_{\bar{y}}(x, y, \mu^{\bar{y}}) \\ & \bar{y} \in [y^L, y^U] \\ & y \in [y^L, y^U] . \end{aligned} \tag{14}$$

Restricting $\bar{y}$ to be y^L or y^U, this problem can be further relaxed to

$$\begin{aligned} \min \quad & v(y) \\ \text{subject to} \quad & v(y) \geq \min_x L_{y^L}(x, y, \mu^{y^L}) \\ & v(y) \geq \min_x L_{y^U}(x, y, \mu^{y^U}) \\ & y \in [y^L, y^U] \end{aligned} \tag{15}$$

Problem (15) is called the relaxed dual problem on the interval $[y^L, y^U]$ and denoted by Problem $RDP_{[y^L, y^U]}$.

Since $y \in [y^L, y^U]$ and $x_i = y^i$, we have the following bounds for x_i:

$$\begin{aligned} (y^L)^i \leq x_i \leq (y^U)^i , \quad & \text{if } i \text{ is odd;} \\ \min\{(y^L)^i, (y^U)^i\} \leq x_i \leq \max\{(y^L)^i, (y^U)^i\} , \quad & \text{if } i \text{ is even and } 0 \notin [y^L, y^U]; \\ 0 \leq x_i \leq \max\{(y^L)^i, (y^U)^i\} , \quad & \text{if } i \text{ is even and } 0 \in [y^L, y^U] . \end{aligned}$$

Let x_i^B be the upper bound of x_i if $\mu_{i+1}^{y^L} \geqslant 0$ and the lower bound otherwise. Because $y^L - y \leqslant 0$, we have

$$\min_x L_{y^L}(x, y, \mu^{y^L}) = L_1(y) = a_0 - \mu_1^{y^L} y^L + \sum_{i=1}^{n} \mu_{i+1}^{y^L}(y^L - y)x_i^B . \tag{16}$$

Similar results hold for the current Lagrange function $L_{y^U}(x, y, \mu)$, i.e.,

$$\min_x L_{y^U}(x, y, \mu^{y^U}) = L_2(y) = a_0 - \mu_1^{y^U} y^U + \sum_{i=1}^{n} \mu_{i+1}^{y^U}(y^U - y)x_i^{B'} , \tag{17}$$

where $x_i^{B'}$ is the lower bound of x_i if $\mu_{i+1}^{y^U} \geqslant 0$ and the upper bound otherwise. Therefore Problem $RDP_{[y^L, y^U]}$ is equivalent to

$$\begin{aligned} &\min && v(y) \\ &\text{subject to} && v(y) \geqslant L_1(y) \\ & && v(y) \geqslant L_2(y) \\ & && y^L \leqslant y \leqslant y^U . \end{aligned} \tag{18}$$

Because both $L_1(y)$ and $L_2(y)$ are linear functions in y, this problem can be easily solved by finding the intersection point of these two functions. Since this problem is a relaxation of the dual Problem (5), its solution is a lower bound of the global minimum on the interval $[y^L, y^U]$.

2.2. THE FRAMEWORK OF THE GOP ALGORITHM

The GOP algorithm for univariate polynomial functions starts from an initial interval $[y^L, y^U]$ and an initial interior point $y^1 \in (y^L, y^U)$. As the algorithm proceeds, the interval is dynamically bipartitioned into smaller subintervals. At each iteration, one primal and two relaxed dual problems are solved, providing more and more precise lower and upper bounds on the global minimum. The algorithm terminates when the gap between the lower and upper bounds is smaller than a given tolerance. It can be stated formally as follows

Step 1. Evaluate $F(y^L)$ and $F(y^U)$. Initialize the set of evaluation points to $PY = \{y^L, y^U\}$. Choose an initial point y^1. Set $v(y^1) = -\infty$ and store it in the set VY. Set the iteration counter $k = 1$, set the upper bound on the global minimum to $P^U = \min\{F(y^L), F(y^U)\}$, and the lower bound to $P^L = -\infty$.

Step 2. Delete $v(y^k)$ from VY. Evaluate $F(y^k)$ and update $P^U = \min\{P^U, F(y^k)\}$. Let $y^{LEFT} = \max\{y | y \in PY, y < y^k\}$, $y^{RIGHT} = \min\{y | y \in PY, y > y^k\}$. Solve the relaxed dual problem $RDP_{[y^{LEFT}, y^k]}$, let its solution be $(y^*, v(y^*))$. If $v(y^*) < P^U - \epsilon$ where ϵ is the given tolerance, store this solution, i.e., put y^* in the set PY and store the value $v(y^*)$ in VY; otherwise discard the subinterval $[y^{LEFT}, y^k]$.

Then solve the relaxed dual Problem $RDP_{[y^k, y^{RIGHT}]}$ and store its solution if the optimal value is less than $P^U - \epsilon$. Update the iteration counter, i.e. $k \leftarrow k + 1$.

Step 3. Choose $y^k \in PY - \{y^L, y^U\}$ so that $v(y^k) = \min\{v(y) | y \in Y - \{y^L, y^U\}\}$. Update $P^L = v(y^k)$. If $P^U - P^L < \epsilon$, stop. Otherwise, return to *Step 2.*

3. Interval Arithmetic

3.1. INTRODUCTION

Interval Arithmetic was introduced by Moore [10] as a basic tool for control of numerical errors in machine computations. Instead of approximating a real value x by a machine representable number a pair of machine representable numbers is used, representing an interval in which x lies. Arithmetic operations for intervals are defined as follows

$$[a, b] + [c, d] = [a + c, b + d] \tag{19}$$

$$[a, b] - [c, d] = [a - d, b - c] \tag{20}$$

$$[a, b] \cdot [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ab, ac, bc, bd\}] \tag{21}$$

$$[a, b]/[c, d] = [a, b] \cdot [1/d, 1/c] \text{ if } 0 \not\in [c, d] . \tag{22}$$

These definitions are readily used to compute intervals containing the range of rational functions $f(y)$ for y belonging to an interval Y. The simplest procedure is to use the *natural extension form* of $f(y)$. It consists in replacing each occurrence of variable y by the interval Y containing it and then applying the above rules of interval arithmetic. Special procedures for bounding trigonometric and transcendental functions allow the extension of this procedure to analytical functions. The bounds so obtained are not always precise, but often better ones can be obtained by various means, see Ratschek and Rokne [11] for a thorough survey and discussion.

While interval arithmetic was not initially designed for global optimization, it was soon understood that it could be effectively used to solve such problems. The many efforts made in that direction are summarized in the recent book of Ratschek and Rokne [12].

3.2. BOUNDING POLYNOMIAL FUNCTIONS USING INTERVAL ARITHMETIC

When applying interval arithmetic to obtain bounds on a function $f(y)$ over an interval Y, different results can be obtained if different expressions of the function are used. For the special case of a polynomial function $f(y) = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n$, the following forms are the most used ones:

Natural Extension Form. As explained in the previous subsection, an inclusion interval for $f(y)$ can be obtained by replacing each occurrence of the variable y in the expression $f(y) = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n$ by the interval Y and applying the rules of interval arithmetic.

Nested Form. An inclusion interval for the function $f(y)$ can also be obtained by writing $f(y)$ in the nested form $f(y) = ((\cdots(a_n y + a_{n-1})y + \cdots)y + a_1)y + a_0$, then replacing each occurrence of variable y in this expression by the interval Y and applying the rules of interval arithmetic.

Centered Form. An inclusion interval for the function $f(y)$ can be obtained by writing Taylor's expansion of $f(y)$ at the center c of Y: $f(y) = f(c) + f'(c)(y-c) + (f''(c)/2)(y-c)^2 + \cdots + (f^{(n)}(c)/n!)(y-c)^n$, then replacing each occurrence of y by the interval Y and applying the rules of interval arithmetic.

Remainder Forms. The idea of the remainder form (due to Cornelius and Lohner [2]) is to replace a part of the function $f(y)$ for which an exact range is available by this range and then to use natural extension or some other forms to obtain a range for $f(y)$ itself. This idea can be extended in the following two ways: (1) write the function $f(y)$ in the form $f(y) = (a_0 + a_1 y + a_2 y^2) + y^3((a_3 + a_4 y + a_5 y^2) + y^3(a_6 + \cdots)\cdots)$, then obtain an interval containing the range of $f(y)$ by replacing the quadratic parts $a_0 + a_1 y + a_2 y^2$, $a_3 + a_4 y + a_5 y^2, \ldots,$ by their exact range, replacing all y^3 by Y^3, and applying the rules of interval arithmetic. This form will be called *quadratic remainder form*; (2) write the function $f(y)$ in the form $f(y) = (a_0 + a_1 y + a_2 y^2 + a_3 y^3) + y^4((a_4 + a_5 y + a_6 y^2 + a_7 y^3) + y^4(a_8 + \cdots)\cdots)$, then obtain an interval containing the range of $f(y)$ by replacing all cubic parts $a_0 + a_1 y + a_2 y^2 + a_3 y^3$, $a_4 + a_5 y + a_6 y^2 + a_7 y^3, \ldots,$ by their exact range for $y \in Y$, replacing y^4 by Y^4, and applying the rules of interval arithmetic. This form will be called *cubic remainder form*.

4. An Interval Arithmetic Algorithm for the Global Minimization of Polynomial Functions

4.1. THE CORD-SLOPE FORM OF POLYNOMIAL FUNCTIONS

We first introduce the following cord-slope form of polynomial functions

THEOREM 1. *For any fixed point $\bar{y}$ in the interval $[y^L, y^U]$,*

$$F(y) - F(\bar{y}) = (\bar{y} - y)\left(\sum_{i=0}^{n-1} \mu_{i+i}^{\bar{y}} y^i\right), \quad \forall y \in [y^L, y^U] \tag{23}$$

where the coefficients $\mu_i^{\bar{y}}$ are the same as the coefficients $\mu_i^{\bar{y}}$ defined in (10).

This form has been used by Alefeld [1] to compute the roots of polynomial functions. A straightforward proof of Theorem 1 is provided below.

Proof. From the definition of $\mu_i^{\bar{y}}$ in (10), we have

$$\begin{aligned}(\bar{y}-y)\sum_{i=0}^{n-1}\mu_{i+1}^{\bar{y}}y^i &= \sum_{i=0}^{n-1}\left(-\sum_{j=i+1}^{n}a_j\bar{y}^{j-i-1}\right)y^i(\bar{y}-y)\\ &= -\sum_{j=1}^{n}\sum_{i=0}^{j-1}(a_j\bar{y}^{j-i-1}y^i(\bar{y}-y))\\ &= -\sum_{j=1}^{n}a_j\sum_{i=0}^{j-1}(\bar{y}^{j-i}y^i-\bar{y}^{j-i-1}y^{i+1})\\ &= -\sum_{j=1}^{n}a_j(\bar{y}^j-y^j)\\ &= \sum_{j=1}^{n}a_jy^j-\sum_{j=1}^{n}a_j(\bar{y}^j)=F(y)-F(\bar{y})\,.\end{aligned}$$

■

We will write

$$F_{\bar{y}}(y)=\sum_{i=0}^{n-1}\mu_{i+1}^{\bar{y}}y^i \tag{24}$$

and call

$$F(y)=F(\bar{y})+(\bar{y}-y)F_{\bar{y}}(y) \tag{25}$$

the cord-slope form of the polynomial $F(y)$ with respect to $\bar{y}$. The cord-slope form for rational functions is discussed in Section 5. Cord-slope forms for general univariate functions and their use in global optimization are discussed in Hansen *et al.* [8].

4.2. BOUNDING THE GLOBAL MINIMUM ON AN INTERVAL USING INTERVAL ARITHMETIC

Now consider the problem of obtaining bounds on the global minimum of the polynomial function $f(y)=a_0+a_1y+a_2y^2+\cdots+a_ny^n$ on an interval $[y^L, y^U]$. An upper bound can be easily obtained by evaluating the function value at any point in this interval. To obtain a lower bound, using the cord-slope form of $F(y)$ with respect to y^L, we have

$$F(y)=F(y^L)+(y^L-y)F_{y^L}(y)\,. \tag{26}$$

Since $F_{y^L}(y)$ is a polynomial function in y, various methods discussed in Section 3 can be applied to obtain bounds on its range. Let $F_{y^L}^U$ be an upper bound so obtained; since $y^L-y\leqslant 0$ for $y\in[y^L, y^U]$, we have

$$F(y)\geqslant C_{y^L}^1(y)=F(y^L)+F_{y^L}^U(y^L-y)\,. \tag{27}$$

Let F^L_{yU} be a lower bound of $F_{yU}(y)$ also obtained by interval arithmetic techniques; since $y^U - y \geq 0$ for all $y \in [y^L, y^U]$, we have,

$$F(y) \geq C^2_{yU}(y) = F(y^U) + F^L_{yU}(y^U - y). \tag{28}$$

Since $C^1_{yL}(y)$ and $C^2_{yU}(y)$ are two underestimating linear functions of $F(y)$, a lower bound of $F(y)$ can be obtained by finding the intersection point y^* of $C^1_{y_L}(y)$ and $C^2_{yU}(y)$, then $F_l = \min\{F(y^L), F(y^U), C^1_{yL}(y^*)\}$ is a valid lower bound on the function value.

4.3. AN INTERVAL ARITHMETIC ALGORITHM TO FIND THE GLOBAL MINIMUM

The interval arithmetic algorithm starts from the initial interval $[y^L, y^U]$. The bounding techniques of the previous subsection are applied to obtain valid lower and upper bounds on the global minimum of $F(y)$ on this interval. If the gap between those two bounds is less than a given tolerance ϵ, the algorithm terminates. Otherwise, the interval is bipartitioned into two subintervals according to the intersection point of the two underestimating linear functions, and the algorithm proceeds to the next iteration. When there are more than one subinterval left, the subinterval for which the lower bound on the function value obtained is minimum is chosen to be further explored first. The procedure can be formally stated as follows:

Step 0. Let $Y^0 = [y^L, y^U]$. Evaluate $F(y^L)$ and $F(y^U)$ and initialize $y_{min} = y^L$ if $F(y^L) < F(y^U)$, $y_{min} = y^U$ otherwise. Initialize $F_{min} = F(y_{min})$. Use the interval arithmetic techniques to find the two underestimating linear function $C^1_{yL}(y)$ and $C^2_{yU}(y)$ as in the previous subsection. Find the intersection point $(y^0, v(y^0))$ of them. If $F_{min} - v(y^0) < \epsilon$, stop. Otherwise initialize a list L containing only the triplet $(Y^0, y^0, v(y^0))$. Set $k = 0$.

Step 1. If L is empty, stop. Otherwise take the last triplet in L and denote it by $(Y^k, y^k, v(y^k))$. Remove it from the list L. If $F_{min} - v(y^k) \leq \epsilon$, stop. Otherwise go to *Step 2*;

Step 2. Let y_1 be the left endpoint of Y^k, y_2 be the right endpoint of Y^k, and let $V^1 = [y_1, y^k]$, $V^2 = [y^k, y_2]$. For the interval V^1, find the intersection point of the two underestimating functions $C^1_{y_1}(y)$ and $C^2_{y^k}(y)$, denote it by $(y^{V^1}, v(y^{V^1}))$. Evaluate $F(y^{V^1})$, if $F(y^{V^1}) < F_{min}$, update $y_{min} = y^{V^1}$, $F_{min}F(y^{V^1})$. If $F_{min} - v(y^{V^1}) < \epsilon$, discard V^1; otherwise insert the triplet $(V^1, y^{V^1}, v(y^{V^1}))$ into the list L so that the third member of the elements in L is in a nondecreasing order. Repeat the same process for $V^2 = [y^k, y_2]$ and return to *Step 1*.

4.4. CORRESPONDENCE WITH THE GOP ALGORITHM

The procedure described in Section 4.2 is used to find a lower bound of $f(y)$ on an interval in the interval arithmetic algorithm. Suppose the interval under evaluation is $Y=[y^L, y^U]$. Various forms of polynomial functions can be applied to find the bounds $F^U_{y^L}$ and $F^L_{y^U}$. Let us consider in detail the *natural extension form*: the function $F_{y^L}(y)$ is written

$$F_{y^L}(y)=\sum_{i=1}^{n-1}\mu^{y^L}_{i+1}y^i\,. \tag{29}$$

Since $y\in Y=[y^L, y^U]$, the inclusion intervals for y^i are

$$y^i\in Y^i$$
$$=\begin{cases}[(y^L)^i,(y^U)^i] & \text{if } i \text{ is odd}\,,\\ [\min\{(y^L)^i,(y^U)^i\},\max\{(y^L)^i,(Y^U)^i\}] & \text{if } i \text{ is even and } \emptyset\not\in[y^L,y^U]\,,\\ [0,\max\{(y^L)^i,(y^U)^i\}] & \text{if } i \text{ is even and } 0\in[y^L,y^U]\,.\end{cases}$$

Therefore, using interval arithmetic

$$F_{y^L}(Y)=\sum_{i=1}^{n-1}\mu^{y^L}_{i+1}Y^i\,. \tag{30}$$

Then the upper bound so obtained is

$$F^U_{y^L}=\sum_{i=1}^{n-1}\mu^{y^L}_{i+1}y^B_i \tag{31}$$

where y^B_i is the right endpoint of Y^i if $\mu^{y^L}_{i+1}>0$ and the left endpoint otherwise. Substituting this into the expression of $C^1_{y^L}(y)$, it is clear that $C^1_{y^L}(y)=L_1(y)$ where $L_1(y)$ is defined in (16) (since $F(y^L)=a_0-\mu^{y^L}_1 y^L$ and $\mu^{y^L}_{n+1}=0$). Similarly, $C^2_{y^U}(y)=L_2(y)$, where $L_2(y)$ is defined in (17). Therefore the lower bound so obtained is the same as the one obtained by solving the relaxed dual Problem $RDP_{[y^L,y^U]}$. Since the evaluation of the function value at y^k is equivalent to solving the primal problem in GOP, and the framework of the interval arithmetic algorithm is the same as the GOP algorithm, this version of the interval arithmetic algorithm is equivalent to the GOP algorithm. (Except for the first iteration, as the GOP algorithm starts from a chosen starting point while the interval arithmetic algorithm starts by evaluating the initial interval. The difference is however minor and both algorithms can easily be modified to follow the other way.)

5. Extension of the Algorithm to Rational Functions

5.1. THE CORD-SLOPE FORM FOR RATIONAL FUNCTIONS

Consider a rational function

$$F(y)=\frac{A(y)}{B(y)}$$

defined on an interval $[y^L, y^U]$, where $A(y)$ and $B(y)$ are polynomial functions. Take any point $\bar{y} \in [y^L, y^U]$; the cord-slope forms for $A(y)$ and $B(y)$ are

$$A(y) - A(\bar{y}) = (\bar{y} - y)A_{\bar{y}}(y)\,, \quad B(y) - b(\bar{y}) = (\bar{y} - y)B_{\bar{y}}(y)\,. \tag{32}$$

Therefore

$$\begin{aligned} F(y) - F(\bar{y}) &= \frac{A(y)}{B(y)} - \frac{A(\bar{y})}{B(\bar{y})} = \frac{A(y)B(\bar{y}) - B(y)A(\bar{y})}{B(y)B(\bar{y})} \\ &= \frac{A(y)(B(\bar{y}) - B(y)) + B(y)(A(y) - A(\bar{y}))}{B(y)B(\bar{y})} \\ &= \frac{-(\bar{y} - y)A(y)B_{\bar{y}}(y) + (\bar{y} - y)B(y)A_{\bar{y}}(y)}{B(y)B(\bar{y})} \\ &= (\bar{y} - y)\frac{B(y)A_{\bar{y}} - A(y)B_{\bar{y}}(y)}{B(y)B(\bar{y})} \\ &= (\bar{y} - y)F_{\bar{y}}(y) \end{aligned}$$

where $F_{\bar{y}}(y)$ is defined as

$$F_{\bar{y}}(y) = \frac{B(y)A_{\bar{y}}(y) - A(y)B_{\bar{y}}(y)}{B(y)B(\bar{y})} \tag{33}$$

which is a rational function of y. The expression

$$F(y) = F(\bar{y}) + (\bar{y} - y)F_{\bar{y}}(y) \tag{34}$$

will be called the cord-slope form of the rational function $f(y)$ with respect to $\bar{y}$.

5.2. AN INTERVAL ARITHMETIC ALGORITHM FOR RATIONAL FUNCTIONS

The interval arithmetic techniques can be applied to obtain upper and lower bounds for the rational function $F_{\bar{y}}(y)$ in the cord slope form of the previous subsection. In particular, we can apply all the forms mentioned in Section 3 to its numerator and denominator to obtain bounds for them separately, then apply interval arithmetic rules to perform the division of the two inclusion intervals. Ar alternative is to apply the *mean value form* to the rational function itself. In this form, we first write the function in Taylor's expansion at the center of the interval.

$$F_{\bar{y}}(y) = F_{\bar{y}}(c) + F'_{\bar{y}}(\xi)(y - c) \tag{35}$$

where ξ belongs to the interval under consideration. An inclusion interval for $F_{\bar{y}}(y)$ can be obtained by replacing y by the interval containing it, $F'_{\bar{y}}(\xi)$ by an inclusion interval for $F'_{\bar{y}}(y)$ and then by applying interval arithmetic rules.

Assume that the range of the denominator of the rational function $F(y)$ over the initial interval does not contain zero. (This assumption can be checked by maximizing and minimizing the denominator with the algorithm of the previous section; if the assumption is not satisfied, a zero of the denominator can be found again using interval arithmetic and the function can be simplified, unless the problem is not well-defined). Using the tools described in the previous paragraph to compute the bounds of the cord-slope $F_{\bar{y}}(y)$, the algorithm of the previous section can be immediately extended to find the global minimum of rational functions on an interval. A special case that may happen during the resolution is that the inclusion interval of the denominator of the cord-slope $F_{\bar{y}}(y)$ may contain zero, in which case we can bipartition the interval at the middle point, assign $-\infty$ as the lower bounds on both subintervals and the algorithm continues. Because all the forms used here to obtain inclusion intervals have convergence order at least one, i.e., the difference between the inclusion interval and the real range converges to zero as the length of the subinterval converges to zero, the inclusion interval of the denominator of the cord-slope will not contain zero when the subinterval is sufficiently small under our assumption. Therefore the algorithm converges under this assumption.

It is worth noting that the GOP algorithm can be also applied to the minimization of rational functions, after the inverse of the denominator of the fraction has been replaced by a new variable (Hansen and Jaumard [9]). However in that case, there are two complicating variables instead of one for polynomial functions. The algorithm becomes quite different and more complicated than the interval arithmetic one for minimization of rational functions.

Of course both the GOP and the interval arithmetic algorithm apply to global optimization of multivariate functions. Comparison of the two methods in that case is beyond the scope of this paper.

6. Computational Experience

6.1. COMPARATIVE RESULTS FOR POLYNOMIAL FUNCTIONS

The algorithm in Section 4 has been implemented in FORTRAN 77 and tested on a SUN 3/50-12 workstation with 1.5 mips central processor. The program has various options depending on the form used to compute the bounds on the cord-slope (see Section 3). Seven test problems used by Visweswaran and Floudas [14] have been solved; they are recalled in Table I. The values of the tolerance considered are 10^{-3}, 10^{-7}, and 10^{-12}. The number of iterations and computing times are listed in Tables II, III and IV respectively.

The number of iterations obtained by the GOP algorithm as reported in Visweswaran and Floudas [14] are also listed for reference (computing times are not available). As shown above, the GOP algorithm works in the same way as the interval arithmetic algorithm when *natural extension form* is used. For most test problems the number of iterations reported in Floudas and Visweswaran [14] and

Table I. Test problems

Problem number	Objective function $f(x)$	Initial interval	n_1	n_2	Source from
1	$f_1(y) = \frac{1}{10} - y - \frac{79}{20}y^2 + \frac{71}{10}y^3 + \frac{39}{80}y^4 - \frac{52}{25}y^5 + \frac{1}{6}y^6$	[−2, 11]	3	1	[16]
2	$f_2(y)$	[1, 2]	1	1	[10]
3	$f_3(y) = 0.000089248y - 0.0218343y^2 + 0.998266y^3 - 1.6995y^4 + 0.2y^5$	[0, 10]	2	1	[15]
4	$f_4(y) = 4y^2 - 4y^3 + y^4$	[−5, 5]	2	2	[4]
5	$f_5(y) = 1.75y^2 - 1.05y^4 + \frac{1}{6}y^6$	[−5, 5]	3	1	[4]
6	$f_6(y) = 250 + 27y^2 - 15y^4 + y^6$	[−5, 5]	3	2	[7]
7	$f_7(y) = 10y - 1.5y^2 - 3y^3 + y^4$	[−5, 5]	2	1	[3]

$$f_2(y) = -500.0y + 2.5y^2 + 1.666666666y^3 + 1.25y^4 + 1.0y^5 + 0.833333333y^6 + 0.714285714y^7 + 0.625y^8 + 0.555555555y^9 + 1.0y^{10} - 43.63636363y^{11} + 0.416666666y^{12} + 0.384615384y^{13} + 0.357142857y^{14} + 0.333333333y^{15} + 0.3125y^{16} + 0.294117647y^{17} + 0.277777777y^{18} + 0.263157894y^{19} + 0.25y^{20} + 0.238095238y^{21} + 0.227272727y^{22} + 0.217391304y^{23} + 0.208333333y^{24} + 0.2y^{25} + 0.192307692y^{26} + 0.185185185y^{27} + 0.178571428y^{28} + 0.344827586y^{29} + 0.666666666y^{30} - 15.48387097y^{31} + 0.15625y^{32} + 0.151515151y^{33} + 0.147058823y^{34} + 0.142857142y^{35} + 0.138888888y^{36} + 0.135135135y^{37} + 0.131578947y^{38} + 0.128205128y^{39} + 0.125y^{40} + 0.121951219y^{41} + 0.119047619y^{42} + 0.116279069y^{43} + 0.113636363y^{44} + 0.111111111y^{45} + 0.108695652y^{46} + 0.106382978t^{47} + 0.208333333y^{48} + 0.408163265y^{49} + 0.80y^{50}.$$

n_1: number of local minima; n_2: number of global minima.

Table II. Comparative computational results with tolerance 10^{-3}

Problem	NEF		NF		CF		QRF		CRF		GOP
	N	T	N	T	N	T	N	T	N	T	N
1	22	0.54	13	0.26	13	0.76	13	0.40	17	0.50	11
2	40	8.02	36	5.18	27	58.52	37	8.42	36	6.90	34
3	18	0.36	12	0.18	12	0.46	14	0.32	17	0.44	13
4	27	0.38	18	0.26	17	0.50	26	0.52	16	0.34	30
5	21	0.44	21	0.36	17	0.92	39	1.02	17	0.50	27
6	37	0.86	25	0.42	23	1.26	33	0.88	37	1.02	68
7	11	0.16	8	0.10	11	0.32	10	0.20	9	0.22	24
Average	25.14	1.54	19.00	0.97	17.14	8.96	24.57	1.68	21.29	1.42	29.57

NEF: Natural Extension Form; NF: Nested Form; CF: Centered Form; QRF: Quadratic Remainder Form; CRF: Cubic Remainder Form; N: number of iterations; T: computing time in seconds (on SUN 3/50).

Table III. Comparative computational results with tolerance 10^{-7}

Problem	NEF		NF		CF		QRF		CRF		GOP
	N	T	N	T	N	T	N	T	N	T	N
1	32	0.74	21	0.38	20	1.12	20	0.58	26	0.78	19
2	46	9.22	44	6.30	34	72.96	44	9.90	44	8.62	45
3	27	0.50	19	0.30	18	0.78	21	0.46	25	0.62	28
4	47	0.70	32	0.40	31	1.16	46	0.94	29	0.70	54
5	27	0.58	21	0.34	23	1.22	41	1.06	17	0.54	34
6	57	1.32	37	0.70	37	2.04	47	1.30	54	1.66	
7	18	0.30	16	0.22	16	0.50	17	0.34	14	0.32	216
Average	36.29	1.91	27.14	1.23	25.57	11.40	33.71	2.08	29.86	1.89	66.00

See legend of Table II for explanation.

Table IV. Comparative computational results with tolerance 10^{-12}

Problem	NEF		NF		CF		QRF		CRF		GOP
	N	T	N	T	N	T	N	T	N	T	N
1	44	1.00	28	0.52	27	1.58	28	0.76	35	1.00	26
2	59	11.96	49	7.32	43	95.06	52	11.72	52	10.60	45
3	37	0.68	27	0.42	27	1.18	30	0.68	39	1.10	38
4	77	1.16	48	0.60	47	1.42	73	1.58	46	1.04	62
5	35	0.72	21	0.34	31	1.76	49	1.28	17	0.56	43
6	87	1.96	54	0.96	53	2.92	75	2.04	73	2.06	
7	27	0.38	21	0.26	26	0.78	26	0.54	24	0.52	
Average	52.29	2.55	35.43	1.49	36.29	14.96	47.57	2.66	40.86	2.41	42.80

See legend of Table II for explanation.

the number of iterations obtained with the natural extension form option are close. Differences appear to be due to the choice of initial point used (and not available for comparison). Exceptions are problem 6 and 7 for which Floudas and Visweswaran [14] report much larger numbers of iterations. Some other unidentified factor (perhaps numerical stability) may be at play here.

It appears that:

(1) For all cases, the smallest computing time is achieved by using the *nested form*. All the problems except Problem 2 are solved to within a tolerance of 10^{-12} in less than 1 second using this form; Problem 2 for which the objective function is of 50th order is solved in 7.34 seconds. On the average, the second smallest computing time is obtained by using the *cubic remainder form*, however the computing times obtained by using the *natural extension form* and the *quadratic remainder form* are both very close. The average computing time obtained by using the *centered form* is much larger, mainly because of the computing time required to solve Problem 2.

(2) For the tolerance values of 10^{-3} and 10^{-7}, the smallest number of iterations on the average is obtained by using the *centered form*. However for the tolerance of 10^{-12}, it is obtained by using the *nested form*. Their difference is within 10%. The largest number of iterations is obtained by using the *natural extension form*.

6.2. COMPARATIVE RESULTS FOR RATIONAL FUNCTIONS

For rational functions, the cord-slope $F_{\bar{y}}(y)$ is also a rational function. As mentioned in Section 5, bounds on this cord-slope can be obtained by applying all the forms mentioned in Section 3 to the numerator and to the denominator separately and then applying interval arithmetic rules of division. An alternative is to use the mean-value form on this rational function directly. A set of 7 test

Table V. Test problems with rational functions

Problem number	Objective function $f(x)$	Initial interval	n_1	n_2
1	$r_1(y) = f_3(y)/(f_4(y) + 5.0)$	[−5, 5]	1	1
2	$r_2(y) = f_4(y)/f_6(y)$	[−5, 5]	2	2
3	$r_3(y) = f_6(y)/(f_4(y) + 10)$	[−5, 5]	2	1
4	$r_4(y) = f_2(y)/(f_1(y) + 3000)$	[1, 2]	1	1
5	$r_5(y) = f_3(y)/f_6(y)$	[−5, 5]	3	1
6	$r_6(y) = f_5(y)/(f_7(y) + 10)$	[−5, 5]	3	1
7	$r_7(y) = f_3(y)/(f_5(y) + 10)$	[−5, 5]	2	1

problems with rational objective functions constructed from polynomial functions used in the previous subsection is provided in Table V. Comparative computational results are provided in Tables VI, VII and VIII:

It appears that:

Table VI. Comparative computational results with tolerance 10^{-3}

Problem	NEF		NF		CF		QRF		CRF		MVF	
	N	T	N	T	N	T	N	T	N	T	N	T
1	42	2.48	13	0.76	15	1.94	28	1.82	31	2.10	17	2.28
2	270	16.32	82	4.20	25	3.82	153	9.18	103	6.56	101	13.50
3	41	2.70	19	1.26	14	2.10	21	1.50	34	2.66	21	3.30
4	49	34.38	35	22.50	26	75.58	35	24.26	40	28.08	23	32.84
5	266	18.96	82	5.06	24	4.24	150	10.48	107	8.16	92	13.56
6	53	3.80	25	1.56	25	3.84	25	3.88	33	2.82	28	4.40
7	32	2.58	13	1.00	16	3.04	18	1.48	23	2.28	18	3.72
Average	107.57	11.60	38.43	5.19	20.71	13.51	61.43	7.51	53.00	7.52	42.86	10.51

See legend of Table II for explanation.

Table VII. Comparative computational results with tolerance 10^{-7}

Problem	NEF		NF		CF		QRF		CRF		MVF	
	N	T	N	T	N	T	N	T	N	T	N	T
1	52	3.24	19	1.20	21	2.82	40	2.70	38	2.76	23	3.36
2	289	17.92	88	4.64	39	6.38	169	10.56	110	7.26	117	16.64
3	53	3.74	24	1.58	23	3.58	32	2.44	44	3.54	27	4.40
4	60	42.08	42	26.92	31	89.80	43	29.76	46	32.28	28	39.84
5	277	20.14	90	5.80	31	5.74	158	11.28	115	8.96	98	14.86
6	66	4.74	32	2.02	37	5.82	41	2.94	40	3.44	42	6.86
7	41	3.50	20	1.16	24	4.80	28	2.50	31	3.18	26	5.68
Average	119.71	13.62	45.00	6.19	29.43	16.99	73.00	8.88	60.57	8.77	51.57	13.09

See legend of Table II for explanation.

Table VIII. Comparative computational results with tolerance 10^{-12}

Problem	NEF		NF		CF		QRF		CRF		MVF	
	N	T	N	T	N	T	N	T	N	T	N	T
1	68	4.50	28	1.76	29	4.02	55	3.82	45	3.40	32	4.92
2	313	19.91	97	5.28	52	8.78	187	12.08	119	8.08	137	20.58
3	67	4.92	34	2.34	31	4.96	42	3.26	54	4.46	33	5.50
4	70	48.98	49	31.34	39	112.96	52	35.88	54	37.90	38	53.94
5	291	21.52	98	6.46	39	7.40	167	12.12	128	10.34	107	16.92
6	83	6.08	41	2.54	55	8.82	58	4.28	48	4.18	58	9.66
7	55	4.92	27	2.26	32	6.56	37	3.44	46	4.80	35	7.86
Average	135.29	15.83	53.43	7.43	39.57	21.93	85.43	10.70	70.57	10.45	62.86	17.05

See legend of Table II for explanation.

(1) Again for all problems, the smallest computing time is obtained by using the *nested form*. The computing time obtained by using the *quadratic remainder form* and the *cubic remainder form* are very close, the difference being within 3%. However, the computing time obtained by using the *natural extension form* is significantly larger This is because for two of the problems (Problem 2 and Problem 5), many bisections are needed to reduce the size of the interval so that inclusion intervals obtained by the *natural extension form* of the denominator do not contain the value zero. The *centered form* still requires the largest computing times on the average.

(2) On average, the smallest number of iterations is achieved by using the *centered form*, the second smallest is obtained by using the *nested form*. Their difference is now between 25% and 45%. The largest number of iterations is obtained using the *natural extension form*, and is about 4 times as large as the smallest number of iterations obtained on the average.

Acknowledgments

The first and second authors have been supported by FCAR (Fonds pour la Formation de Chercheurs et l'Aide à la Recherche) grant 92EQ1048 and AFOSR grant 90-0008 to Rutgers University. The first author has also been supported by NSERC (Natural Sciences and Engineering Research Council of Canada) grant to H.E.C. and NSERC grant GP0105574. The second author has been supported by NSERC grant GP0036426, FCAR grant 90NC0305 and a NSF Visiting Professorship for Women in Science at Princeton University for the academic year 1990–1991.

References

1. Alefeld, G. (1981), Bounding the Slope of Polynomial Operators and Some Applications, *Computing* **26**, 227–237.

2. Cornelius, H. and R. Lohner, (1984), Computing the Range of Values of Real Functions with Accuracy Higher Than Second Order, *Computing* **33**, 331–347.
3. Dixon, L. C. W. (1990), On Finding the Global Minimum of a Function of One Variable, Presented at the SIAM National Meeting, Chicago.
4. Dixon, L. C. W. and G. P. Szegö (1975), *Towards Global Optimization*, North Holland, Amsterdam.
5. Floudas, C. A. and V. Visweswaran (1990), A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs - I. Theory, *Computers and Chemical Engineering* **14**, 1397–1417.
6. Floudas, C. A. and V. Visweswaran (1993), A Primal-Relaxed Dual Optimization Approach, *Journal of Optimization Theory and Applications* (to appear).
7. Goldstein, A. A. and J. F. Price (1971), On Descent from Local Minima, *Mathematics of Computation* **25**, 569–574.
8. Hansen, P., Jaumard, B., and J. Xiong (1993), The Cord-Slope Form of Taylor's Expansion in Univariate Global Optimization, *Journal of Optimization Theory and Applications* (to appear).
9. Hansen, P. and B. Jaumard (1992), Reduction of Indefinite Quadratic Programs to Bilinear Programs, *Journal of Global Optimization* **2**, 41–60.
10. Moore, R. (1966), *Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.
11. Ratschek, H. and J. Rokne (1984), *Computer Methods for the Range of Functions*, Chichester, Ellis Horwood.
12. Ratschek, H. and J. Rokne (1988), *New Computer Methods for Global Optimization*, Chichester, Ellis Horwood.
13. Visweswaran, V. and C. A. Floudas (1990), A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs - II, Application of Theory and Test Problems, *Computers and Chemical Engineering* **14**, 1419–1434.
14. Visweswaran, V. and C. A. Floudas (1992), Global Optimization of Problems with Polynomial Functions in One Variable, in *Recent Advances in Global Optimization*, pp. 165–199, Floudas, C. A. and Pardalos, P. M. (eds.), Princeton, Princeton University Press.
15. Wilkinson, J. H. (1963), *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, New Jersey.
16. Wingo, D. R. (1985), Globally Minimizing Polynomials without Evaluating Derivatives, *International Journal of Computer Mathematics* **17**, 287–294.