Bounding the Set of Solutions of a Perturbed Global Optimization Problem

E. R. HANSEN
654 Paco Dr., Los Altos, CA 94024, U.S.A.

(Received: 15 April 1991; accepted: 4 May 1991)

Abstract. Consider a global optimization problem in which the objective function and/or the constraints are expressed in terms of parameters. Suppose we wish to know the set of global solutions as the parameters vary over given intervals. In this paper we discuss procedures using interval analysis for computing guaranteed bounds on the solution set. This provides a means for doing a sensitivity analysis or simply bounding the effect of errors in data.

Key words. Global optimization, sensitivity analysis, error analysis, interval analysis, perturbations.

1. Introduction

Let c denote a vector of some number, s, of parameters. Consider a global optimization problem depending on c in which we wish to find the point(s) x in R^n to

$$\begin{aligned} &\text{minimize (globally) } f(x, c) \\ &\text{subject to } p_i(x, c) \leq 0 \; (i = 1, \ldots, m) , \\ &\qquad\qquad q_i(x, c) = 0 \; (i = 1, \ldots, r) . \end{aligned} \tag{1.1}$$

We allow $m = 0$ and/or $r = 0$. Therefore, the problem may be constrained or unconstrained.

Let $x^*(c)$ denote a solution point of (1.1); and let $f^*(c)$ denote the globally minimum value of $f(x, c)$. That is,

$$f^*(c) = f(x^*(c), c) .$$

We consider the case in which each component c_i $(i = 1, \ldots, s)$ is allowed to vary over an interval C_i. Thus, there is a set of solutions

$$x^*(C) = \{x^*(c): \quad c \in C\}$$

and a set of globally minimum values

$$f^*(C) = \{f^*(c); \quad c \in C\}$$

of the objective function. (The notation $c \in C$ indicates that $c_i \in C_i$ for all $i = 1, \ldots, s$.)

Define a *box* to be a vector of intervals. Let $X^*(C)$ denote the smallest box containing the set, $x^*(C)$, of solution points.

In this paper, we given an algorithm which computes lower and upper bounds on $f^*(C)$ and on the components of $X^*(C)$. These bounds are guaranteed to be correct despite the presence of roundoff and approximation errors. The lower bound on $f^*(C)$ is always sharp (except for small roundoff and approximation errors). We shall sometimes be able to prove that the upper bound on $f^*(C)$ and the lower and upper bounds on $X^*(C)$ are sharp, as well. The guarantees of correctness are made possible by the use of interval analysis.

The algorithm provides a means for bounding the solution of a problem in which parameters are uncertain because of measurement errors, for example.

It also provides a means for doing sensitivity analysis. By entering parameters as intervals, we determine how much the global solution varies (in position and value of the objective function) as the parameters vary over their bounding intervals.

In [3], the author described a procedure for solving the perturbed problem (1.1). In Section 2, we call this the "basic algorithm". In the present paper, we give a more general procedure. The essential part of the generalization is given in Section 5.

In Section 2, we describe a "basic algorithm" which uses interval methods to solve the global optimization problem. In Section 3 we give a condition which must be satisfied in order for the method described in this paper to be most useful. The Kuhn–Tucker conditions are discussed in Section 4. Section 5 is the raison d'être for this paper. It contains the procedure for bounding the solution set. Sections 6 and 7 give a means of validating the procedure. Section 8 discusses options for use of the procedure. Sections 9 and 10 contain numerical examples.

2. Global Optimization Algorithm Using Interval Analysis

In this paper, we assume the reader has a basic knowledge of interval analysis. For an introduction to the subject, see [9], for example.

Methods using interval analysis exist for solving the global optimization problem for constrained or unconstrained problems. These methods guarantee that the global solution has been found. All errors are taken into account including rounding and approximation errors.

For the simplest of such method, no differentiability is needed, see [10], [12]. However, efficiency is much greater for the algorithms which use derivatives.

For an introduction to interval global optimization algorithms, see [4]. A survey is given in [7]. Algorithms for the unconstrained, inequality constrained, and equality constrained cases are given in [2], [5], and [6], respectively. More detailed algorithms will appear in the author's forthcoming book *Global Optimization Using Interval Analysis*.

For the basic algorithm discussed later in this section, we assume that the constraint functions are continuously differentiable with respect to x and that the objective function is twice continuously differentiable with respect to x. Thus, we are able to use the most efficient algorithms. For example, we are able to apply an interval Newton method to solve the Kuhn–Tucker conditions. No differentiability with respect to the parameter vector, c, is required.

For the procedure to be introduced in Section 5, we require the constraint and objective functions to have three and four continuous derivatives, respectively, with respect to x. We also require (for this procedure) that these functions be twice continuously differentiable with respect to the parameter vector, c.

We shall give only a bare outline of these algorithms. The important feature is that they produce an interval guaranteed to contain the globally minimum value of the objective function. They also produce a box (or boxes) guaranteed to contain the global solution point(s).

The ability to solve global optimization problems by interval methods is a result of the following theorem due to Moore. (See, for example, [9].) It has come to be called the fundamental theorem of interval analysis. See [9], for example, for definitions of the terms used.

THEOREM 2.1. *Let $F(x)$ be an inclusion monotonic interval extension of a real function, $f(x)$. Let X be an interval vector. Then $F(X)$ contains the range of $f(x)$ for all $x \in X$.*

The global optimization algorithms proceed as follows:
Begin with a box, $X^{(0)}$, large enough to contain any point of global minimum. Using fail-safe procedures, delete sub-boxes of this original box which are proved by interval procedures to not contain a point of global minimum. Stop when the remaining box of boxes satisfy given termination conditions.

During the solution process, we determine and continually improve an upper bound, $\bar{f}$, on the global minimum, f^*. This is done by sampling values of $f(x, c)$ at points in $X^{(0)}$ proved to be feasible.

If there are no equality constraints, it is a simple procedure to prove feasibility of a point using rounded interval arithmetic. Hansen and Walster [6] show how to prove feasibility when there are equality constraints and rounding is present.

We now list the primary procedures used for deleting a sub-box, X, of $X^{(0)}$.

(1) Delete X if $f(x, c) > \bar{f}$, for all $x \in X$.

(2) If X is in the interior of the feasible region, delete X if

(a) A component of the gradient of $f(x, c)$ is nonzero for every point, x, in X. (This assures that there is no stationary point of $f(x, c)$ in X.)

(b) A diagonal element of the Hessian of $f(x, c)$ is negative for every point, x, in X. (This assures that $f(x, c)$ is not convex at any point in X.)

(3) Delete X if no point in X is feasible.

(4) Apply a step of an interval Newton method to solve the Kuhn–Tucker equations over X. (This procedure either deletes all of X or, in effect, deletes the part of X not retained.)

Theorem 2.1 makes it possible to implement these steps as a practical algorithm.

The algorithm is efficient and outstandingly robust. In [13], an instance was reported in which failure occurred because only internal storage was used on the computer; and the algorithm ran out of space. A later version of the algorithm solved the problem with no difficulty. The author has never observed any kind of failure of the algorithm on any other problem.

It is possible (and has been done) to write a program which solves the general problem (1.1) or any special case of it (such as the unconstrained case). We shall assume such an algorithm is used and refer to it as "the basic algorithm". We shall not always be specific about whether the problem being discussed is unconstrained or constrained. For the constrained case we shall not be specific about what kind of constraints occur.

The sharpness of the bounds computed by the basic algorithm depends on tolerances specified by the user. In the algorithm discussed below, the basic algorithm is first used to solve (1.1) with loose tolerances to get crude bounds on the solution value and points. Then a separate computation (see Section 5) is used to obtain bounds on $X^*(C)$ and $f^*(C)$.

One tolerance which must be specified in the global optimization algorithm is a number, ε_X. The basic algorithm computes a box (or set of boxes) which contains a global solution point. If there is more than one global solution point, the algorithm computes a set of boxes containing each of them. Each component of each such box will be an interval whose width is less than or equal to ε_X.

The other tolerance is a bound, ε_F, on how much the objective function, $f(x, c)$, can change over an output box. That is, the basic algorithm assures that for each output box, X,

$$\max_{x \in X} f(x, c) - \min_{x \in X} f(x, c) \leq \varepsilon_F . \tag{2.1}$$

The basic algorithm can solve the perturbed problem in which the parameter vector, c, is replaced by an interval vector, C. The solution set will be covered by a set of boxes whose largest component is of width $\leq \varepsilon_X$. See [3].

In fact, the basic algorithm treats all problems as if they were perturbed. This is necessary if correctness is to be guaranteed. The reason is that problems often involve transcendental parameters, such as π, which cannot be represented in a computer's number system. Therefore, they are replaced by intervals which bound them. Also, irrational functions are approximated by rational functions; and the approximation error is bounded by an interval.

If the solution set is large or if ε_X is small, the output will be a large number of boxes and a lot of computing will be necessary to generate them.

In the perturbed case, the algorithm assures that

$$\max_{\substack{x \in X \\ c \in C}} f(x, c) - \min_{\substack{x \in X \\ c' \in C}} f(x, c') \leq \varepsilon_F . \tag{2.2}$$

Note that the left member is generally nonzero in the perturbed case even if X is a single point. Therefore, if ε_F is chosen too small, it is impossible to satisfy (2.2).

Thus, when solving the perturbed case in practice, it is best to choose ε_F relatively large and let ε_X drive the termination process. Hereafter, we assume this is the case.

The main purpose of this paper is to present an algorithm which computes guaranteed bounds on the smallest box containing the solution set (or a set of such boxes if the solution set consists of disjoint subsets). The virtue of the method is that it avoids computing a large number of small boxes covering the solution set. We show that in certain cases, we can prove that the bounding box is the smallest one containing the solution set.

The drawback is that the solution set is not as well defined. However, the actual shape of the solution set is probably of little interest in practice. If the shape is important, the basic algorithm can be used with a small box size tolerance. If only the size of the solution set is of interest, the algorithm to be given in Section 5 provides it with what is generally less computing.

3. A Condition

As pointed out above, the basic algorithm will solve the perturbed problem. Suppose we do so using a relatively large value of ε_X. The result will be one or more relatively large boxes covering the solution set. Denote the box(es) by X'.

In order for all the bounds on $f^*(C)$ and $X^*(C)$ to be sharp, X' must not contain a local (non-global) solution of the optimization problem (1.1) for any $c \in C$. For a given example, we may or may not be able to prove that this is correct. See Section 7. However, there are reasons to expect that the bounds will be fairly sharp in all cases.

It is probably true that, in most problems in practice, the local solutions are sufficiently far from the global solution that X' will not contain a local solution unless ε_X is much too large. Even as it does, any local solution point lying near the global solution point should be "nearly global" and our error is small. For a numerical example, see Section 10.

Despite this uncertainty, our algorithm produces a sharp lower bound on $f^*(C)$.

4. The Kuhn–Tucker Conditions

A necessary condition for a point, x, to be a solution to the global optimization problem (1.1) is that the Kuhn–Tucker conditions be satisfied at x. That is, the function $F(x, c)$ must be zero, where

$$F(x, c) = \begin{bmatrix} \nabla f(x, c) + \sum_{i=1}^{m} u_i \nabla p_i(x, c) + \sum_{i=1}^{r} v_i \nabla q_i(x, c) \\ u_1 p_1(x, c) \\ \vdots \\ u_m p_m(x, c) \\ q_1(x, c) \\ \vdots \\ q_r(x, c) \end{bmatrix} \tag{4.1}$$

and u_i $(i = 1, \ldots, m)$ and v_i $(i = 1, \ldots, r)$ are Lagrange multipliers. We make use of this condition later.

The Kuhn–Tucker conditions also include the requirement that the Lagrange multipliers, u_i $(i = 1, \ldots, m)$, be nonnegative.

When using the Kuhn–Tucker conditions, we must also consider constraint qualifications. An interval global optimization algorithm using the Fritz John conditions is given in [8]. See also [7]. This avoids the need for assuming the constraint qualifications hold. However, for the interval algorithm using the John conditions, we require special attention to normalization of the Lagrange multipliers. To avoid discussing normalization in this paper, we simply assume the constraint qualifications hold.

5. The Procedure

We obtain bounds on $f^*(C)$ and $X^*(C)$ by solving a set of problems. We solve one problem to obtain an upper bound on $f^*(C)$; we solve a different one to obtain the lower bound. We solve separate problems to obtain an upper or a lower bound on each component of $X^*(C)$.

Thus, to bound both $f^*(C)$ and $X^*(C)$, we must solve $2n + 2$ problems. However, they are solved over the small region, X', obtained by the basic algorithm when solving (1.1) with c replaced by the interval vector, C. Therefore, convergence is very fast.

The problems to be solved are unperturbed. Therefore, they can be solved quite sharply by the basic algorithms.

To obtain a lower bound on $f^*(C)$, we solve the following problem:

$$\begin{aligned} &\underset{x,\,c}{\text{minimize}} \text{ (globally) } f(x, c) \\ &\text{subject to } p_i(x, c) \leq 0 \quad (i = 1, \ldots, m)\,, \\ &\qquad\qquad q_i(x, c) = 0 \quad (i = 1, \ldots, r)\,, \\ &\qquad\qquad c \in C\,. \end{aligned} \tag{5.1}$$

Note that the parameters, c_i $(i = 1, \ldots, s)$ now enter as variables. Therefore, the new problem is of higher dimension than the original one. However, C will generally be a small box since perturbations or uncertainties in c are usually small. As a result, the search for a global minimum is made over a small set of values for both x and c; and convergence is rapid.

The condition $x \in X'$ is not imposed as a formal constraint. It merely restricts the search made by the basic algorithm.

To obtain an upper bound on $f^*(C)$, we solve the following problem:

$$\begin{aligned} &\underset{x,c}{\text{maximize}}\ (\textit{globally})\ f(x, c) \\ &\text{subject to } c \in C\,, \\ &\qquad\quad F(x, c) = 0\,. \end{aligned} \tag{5.2}$$

Again, the search is restricted to the region X'.

Note that we have introduced the Kuhn–Tucker condition $F(x, c) = 0$ as a constraint. In so doing, we are assuming that any Kuhn–Tucker point in X' is a global solution to (1.1). The smaller we make ε_X when "solving" the original problem (1.1), the more likely it is that this assumption is correct. In Section 10, we given an example in which this assumption is incorrect.

If some Kuhn–Tucker point in X' is not a global solution to (1.1), solving (5.2) still yields an upper bound on $f^*(C)$. However, it may not be sharp enough to be useful.

In Section 6, we discuss how it may be possible to prove that a solution in X' is unique and, therefore, global.

To obtain the left endpoint of the i-th component, $X_i^*(C)$ $(i = 1, \ldots, n)$, of $X^*(C)$, we solve the problem:

$$\begin{aligned} &\underset{x,\,c}{\text{minimize}}\ (\text{globally})\ x_i \\ &\text{subject to } c \in C\,, \\ &\qquad\quad F(x, c) = 0\,. \end{aligned} \tag{5.3}$$

Again, we restrict the search to the region X'.

To obtain the right endpoint of the i-th component of $X^*(C)$, we solve the problem:

$$\begin{aligned} &\underset{x,\,c}{\text{maximize}}\ (\text{globally})\ x_i \\ &\text{subject to } c \in C\,, \\ &\qquad\quad F(x, c) = 0 \end{aligned} \tag{5.4}$$

with $x \in X'$.

6. Uniqueness

In this section, we discuss a theorem which will be useful in later sections. The theorem provides a means for proving computationally that the solution to an optimization problem is unique.

We shall consider the problem of solving the Kuhn–Tucker condition $F(x, c) = 0$. Since the Lagrange multipliers are also variables in such a problem, we should write the equation as $f(x, u, v, c) = 0$. However, the Lagrange multiplier variables enter in the same way as the variable, x. Hence, for simplicity, we omit explicit reference to them and simply write $f(x, c) = 0$.

Note that when we solve (1.1) using the basic algorithm, we obtain guaranteed bounds on the Lagrange multipliers. See [8].

Suppose we apply an interval Newton method to solve the Kuhn–Tucker condition $F(x, c) = 0$ where $F(x, c)$ is given by (4.1). Assume we seek a solution in a given box, X, for all $c \in C$. Let $J(x, c)$ denote the Jacobian of $F(x, c)$.

Following standard practice in interval analysis, we linearize $F(x, c)$ about some point $x' \in X$ and obtain the equation

$$F(x', c) + J(X, c)(y - x') = 0 . \tag{6.1}$$

to be solved for y. As is well known (e.g., see [11]), the set of solution points, y, of (6.1) contains any solution of $F(x, c) = 0$ in X.

The following theorem is also well known. See, for example, [11].

THEOREM 6.1. *If $J(X, c)$ does not contain a singular matrix, then any solution of $F(x, c) = 0$ in X is unique.*

If, in the Jacobian, we replace the real vector, c, by the box, C, containing it, then Theorem (6.1) becomes

THEOREM 6.2. *If $J(X, C)$ does not contain a singular matrix, then any solution of $F(x, c) = 0$ in X is unique for a given value of $c \in C$.*

Let $m[J(X, C)]$ denote the center of $J(X, C)$. That is, if the element in position (i, j) of $J(X, C)$ is $[a, b]$, then the element in position (i, j) of $m[J(X, C)]$ is $(b + a)/2$.

When solving (6.1) it is common practice (and generally necessary for accuracy) to precondition the equation by multiplying by an approximate inverse, say B, of $m[J(X, C)]$. The preconditioned equation is

$$M(X, C)(y - x') = r(x', C) \tag{6.2}$$

where $M(X, C) = BJ(X, C)$ and $r(x', C) = -BF(x', C)$. The center of $M(X, C)$ approximates the identify matrix.

The solution set of (6.2) contains the solution set of (6.1). See [10]. If $M(X, C)$ does not contain a singular matrix, then neither does $J(X, C)$.

Preconditioning "drives" the coefficient matrix toward the identity matrix and, thus, toward diagonal dominance. If $M(X, C)$ is strictly diagonally dominant, then it does not contain a singular matrix.

An interval matrix, A, of order n with elements $a_{ij} = [a_{ij}^L, a_{ij}^R]$ is said to be *strictly diagonally dominant* if

$$| a_{ii}^L | > \sum_{\substack{i=1 \\ j \neq i}}^{n} | a_{ij} |$$

where

$$| a_{ij} | = \max\{| a_{ij}^L |, | a_{ij}^R |\} .$$

Every real matrix contained in a strictly diagonally dominant interval matrix is strictly diagonally dominant in the usual sense. Note that every such real matrix is nonsingular.

Suppose we have a solution box, X', obtained using the basic algorithm and, thus, known to contain the global minimum of $f(x, c)$ for all $c \in C$. Therefore, it contains a solution to $F(x, c) = 0$ (i.e., a Kuhn–Tucker point) for all $c \in C$.

Suppose we linearize the Kuhn–Tucker function, $F(x, c)$, and obtain (6.1) (with X replaced by X'). Suppose the preconditioned coefficient matrix (see (6.2)) is strictly diagonally dominant. Then, by Theorem 6.2, the Kuhn–Tucker point is unique for each $c \in C$.

That is, we know that X' does not contain a local minimum of $f(x, c)$. The only minimum it contains is the one known to be global.

7. Assuring Validity

In Section 5, we gave problems whose solutions yielded bounds on $f^*(C)$ and $X^*(C)$ provided certain conditions were satisfied. In this section, we discuss how we can check in practice whether these conditions are satisfied.

We first consider the case in which the output, X', when solving (1.1) is a single box. Alternatively, it may consist of a set of abutting boxes which cannot be separated into two or more strictly disjoint subsets.

In the latter case, we shall replace X' by the smallest box containing X' (and again call the result X'). Thus, the case we are considering is when X' is (or has become) a single box.

This box contains the global minimum point (or points) for all $c \in C$. Therefore, solving problem (5.1) over X' will always yield a correct lower bound on $f^*(C)$. This is true even if X' contains a local minimum or more than one point of global minimum.

However, to get the other desired bounds by solving (5.2), (5.3), and (5.4), we must assure that X' does not contain a Kuhn–Tucker point which is not the global minimum. This assurance can be obtained by computing $M(X', C)$ (see Section 6) and verifying that it is strictly diagonally dominant. If $M(X', C)$ is not strictly diagonally dominant, we do not know whether the bounds from (5.2), (5.3), and (5.4) will be sharp or not.

In this failed case, we have two options. First, we can go ahead and solve equations (5.2), (5.3), and (5.4) over X'. Equation (5.2) will yield a lower bound on $f^*(C)$ and (5.3) and (5.4) will yield lower and upper bounds, respectively, on $x^*(C)$. However, we will not know whether these bounds are sharp.

Secondly, we can split X' into two sub-boxes. For a sub-box, X, of X', the interval elements of $M(X, C)$ will generally be narrower than those of $M(X', C)$. Therefore, it may be possible to verify that $M(X, C)$ does not contain a singular matrix. By covering X' with sub-boxes such as X, we may be able to obtain the desired results. Note that if $M(X', C)$ does contain a singular matrix, then $M(X, C)$ will contain a singular matrix for at least one of the boxes, X, covering X'.

However, subdividing X' introduces a new difficulty. We now consider the case in which X' is either composed of disjoint subsets or is subdivided as just suggested. For simplicity, assume X' consists of two boxes, Y and Z.

It may be that, for a given value of c, the box Y contains a global solution and Z contains another Kuhn–Tucker point (or vice versa). In such a case, the basic algorithm assures that the value of the objective function at the Kuhn–Tucker point in Z will differ little from the globally minimum value. Assurance is provided by step (1) of the basic algorithm given in Section 2.

Recall that, for the perturbed problem we are considering, we use loose termination criteria in order to prevent the basic algorithm from computing too large a number of small boxes to cover the solution set. Unfortunately, the looser the termination tolerances, the greater the difference between the value of $f(x, c)$ at the global solution in Y and at the Kuhn–Tucker point in Z.

If X' is composed of more than one subset, we must accept the possibility that we may not obtain a sharp upper bound on $f^*(C)$ when solving (5.2) or sharp bounds on $x^*(C)$ when solving (5.3) and (5.4). The sharpness will depend on the tolerances chosen when solving (1.1) using the basic algorithm. We have no guidelines to offer in choosing the tolerances.

If both Y and Z contain global solutions and no other Kuhn–Tucker point, then (5.3) and (5.4) will yield sharp bounds on $X^*(C)$. Unfortunately, we will not know whether this is true or not.

Even if the minima in Y and Z are both global, we cannot know that the upper bound on $f^*(C)$ obtained from (5.2) is sharp. It may be that a point in Y which is global for a given value of $c \in C$ is only local for another value of c. This can occur if the global solution point jumps to Z as c changes. We given an example of this kind in Section 10.

Note that, despite all the uncertainty in the case in which X' is composed of disjoint sets, some certainties remain. The bounds will always be correct. They will be reasonably sharp unless the tolerances used in the basic algorithm are large.

It may be that Z (say) does not contain a minimum for any value of $c \in C$. The uncertainty introduced by using the interval vector, C, in place of a real vector, c, when applying the basic algorithm could cause such a box to be retained. This causes no difficulty. The constraints in problems (5.2), (5.3), and (5.4) assure that the only solutions to these problems are Kuhn–Tucker points of the original problem (1.1).

8. Procedure

The solutions to the problems in Section 5 do not always provide sharp bounds on $f^*(C)$ and $X^*(C)$. The exception is problem (5.1). Its solution always provides a sharp lower bound on $f^*(C)$. How we proceed in getting the bounds must depend on how sharp we wish the bounds to be. We now consider the options.

No matter how sharp we want the bounds to be, we begin by applying the basic algorithm to solve (1.1). We do so using relatively large values of the termination parameters.

Since we do not know the nature of the solution in advance, it is probably impossible to know, in general, the best choices for values of the parameters. In practice, we have taken the unsatisfactory step of using human intervention. However, our procedure could probably be automated satisfactorily.

We first solve problem (1.1) with large values of the tolerances. If there are too few solution boxes, we reduce the tolerances and let the basic algorithm continue. In effect, we use the output of the first run as input for the second. Thus, there is not any extra computing involved in stopping and restarting.

Suppose the solution consists of a "satisfactory" number of boxes. Suppose the boxes cannot be separated into strictly disjoint subsets. Let X' denote the smallest box containing all of them. If $M(X', C)$ (see Section 6) is strictly diagonally dominant, we solve (5.1) through (5.4) knowing that they will produce sharp bounds.

If $M(X', C)$ is not strictly diagonally dominant, we have two options. We have the same options in the case in which the output of the basic algorithm consists of boxes which form strictly disjoint subsets. We can solve (5.1) through (5.4) as before. In this case, we are accepting bounds which may not be sharp.

Alternatively, we can reduce the termination tolerances to small values and cover the solution set by many small boxes using the basic algorithm. For small tolerances, the basic algorithm will thus provide relatively sharp bounds on $f^*(C)$ and $X^*(C)$.

We have no guidelines to recommend on which option a user should select.

9. First Numerical Example

We now consider an example of a perturbed problem which arose in practice as a chemical mixing problem. It is an equality constrained least squares problem given by

$$\begin{aligned}
&\text{minimize (globally) } f(x) = \sum_{i=1}^{18} (x_i - p_i)^2 \\
&\text{subject to} \quad x_1x_9 = x_2x_{14} + x_3x_4 \,, \\
&\qquad x_1x_{10} = x_2x_{15} + x_3x_5 \,, \\
&\qquad x_1x_{11} = x_2x_{16} + x_3x_6 \,, \\
&\qquad x_1x_{12} = x_2x_{17} + x_3x_7 \,, \\
&\qquad x_1x_{13} = x_2x_{18} + x_3x_8 \,, \\
&\qquad x_4 + x_5 + x_6 + x_7 + x_8 = 1 \,, \\
&\qquad x_9 + x_{10} + x_{11} + x_{12} + x_{13} = 1 \,, \\
&\qquad x_{14} + x_{15} + x_{16} + x_{17} + x_{18} = 1 \,, \\
&\qquad x_{14} = 66.67x_4 \,, \\
&\qquad x_{15} = 50x_5 \,, \\
&\qquad x_{16} = 0.015x_6 \,, \\
&\qquad x_{17} = 100x_7 \,, \\
&\qquad x_{18} = 33.33x_8 \,.
\end{aligned}$$

The parameters and their uncertainties are given in the following table.

i	p_i	
1	100	± 1.11
2	89.73	± 1.03
3	10.27	± 0.51
4	0.0037	± 0.00018
5	0.0147	± 0.0061
6	0.982	± 0.032
7	0	± 0
8	0.0001	± 0
9	0.22	± 0.0066
10	0.66	± 0.017
11	0.114	± 0.0046
12	0.002	± 0.0001
13	0.004	± 0.00012
14	0.245	± 0.0067
15	0.734	± 0.02
16	0.0147	± 0.0061
17	0.0022	± 0.0001
18	0.0044	± 0.0013

We solved this problem on a CYBER 175. To do so, we used the constraints to explicitly eliminate variables and reduced the problem to an unconstrained problem in five variables. We first solved the unperturbed case. We obtained the solution point with 12 digits of guaranteed accuracy in 0.54 seconds. The minimum value of f was found to be $f^* = 3.07354796 \times 10^{-7} + 2 \times 10^{-15}$.

We then solved the perturbed case with loose tolerances using the same initial box. The algorithm took 3.0 seconds to obtain a set of 59 boxes covering the solution set. The smallest box (call it X') containing these 59 boxes is given in the following table.

i	X'_i
1	[96.2, 103.8]
2	[87.7, 91.7]
3	[8.47, 12.07]
4	[0.0035, 0.00384]
5	[0.0142, 0.0152]
6	[0.98142, 0.98147]
7	[−0.000205, 0.000249]
8	[−0.000384, 0.000645]
9	[0.1983, 0.2440]
10	[0.6046, 0.7207]
11	[0.0603, 0.1638]
12	[−0.0174, 0.0237]
13	[−0.0109, 0.0205]
14	[0.2338, 0.2559]
15	[0.7122, 0.7556]
16	[0.0147, 0.0148]
17	[−0.0205, 0.0249]
18	[−0.0128, 0.0215]

We obtained the crude interval $[0, 2.556]$ bounding $f^*(C)$.

By applying the method described in Section 7 using the above box, X', we showed that any solution (of the original minimization problem) in X' was unique for each $c \in C$. Therefore, we know that the problems in Section 5 will yield precise bounds on $f^*(C)$ and $X^*(C)$.

We did not do the computations. It seems certain that they could be carried to a successful completion. In the author's ten years of experience on global optimization, no failure an interval algorithm on any problem has occurred. A previously reported failure was apparently due to a programming error. See [10].

10. Second Example

As a second numerical example, we consider a two-dimensional unconstrained problem. The objective function is

$$f(x, c) = 12x_1^2 - 6.3x_1^4 + cx_1^6 + 6x_2(x_1 + x_2) .$$

For $c = 1$, this is the (negative of) the so-called three hump camel function. See, for example, [1].

We perturb the problem by letting c vary over the interval $C = [0.9, 1]$. For all c for which $0.945 < c \leq 1$, the global minimum is the single point at the origin; and $f^* = 0$ at this point. For $c = 0.945$, there are three global minima. One is at the origin. The others are at $\pm(a, -a/2)$ where $a = (10/3)^{1/2}$. For $c < 0.945$, there are two global minima at $\pm(b, -b/2)$ where

$$b = \{[4.2 + (17.64 - 14c)^{1/2}]/(2c)\}^{1/2} .$$

At each of these two points, f has the negative value

$$f^* = [22.05c - 18.522 + (3.5c - 4.41)(17.64 - 14c)^{1/2}]/c^2 .$$

The smallest value of f^* for $c \in [0.9, 1]$ occurs for $c = 0.9$ where $f^* = -1.8589$, approximately.

Consider perturbing the problem continuously by letting c increase from an initial value of 0.9 to a final value of 1. Initially, there are two global solution points. Each moves along a (separate) straight line in the x_1, x_2-plane until $c = 0.945$. As c passes through this value, the global solution jumps to the origin and remains there for all $c \in [0.945, 1]$.

The discontinuous movement of the points of global solution creates no difficulty for the basic algorithm. Each point which is a global solution for any $c \in C$ is contained in the solution set.

The solution set consists of three parts. One part is the origin. Another is the line segment joining the points $(a, -a/2)$ and $(b', -b'/2)$ where $b' = \{[21 + (126)^{1/2}]/9\}^{1/2}$. The third is the reflection of this line segment in the origin.

The smallest (disjoint) boxes which can cover the three parts of the solution set are

$$\begin{bmatrix} [0, 0] \\ [0, 0] \end{bmatrix}, \quad \begin{bmatrix} [a, b'] \\ [-b'/2, -a/2] \end{bmatrix}, \quad \begin{bmatrix} [-b', -a] \\ [a/2, b'/2] \end{bmatrix}.$$

We solved this problem with c replaced by the interval $[0.9, 1]$ using the basic algorithm. The search was made in the box defined by $-2 \leq x_i \leq 4$ $(i = 1, 2)$. We chose the box size tolerance, ε_X, to have the relatively large value 0.1 so that the solution set would be covered by only a few boxes. We chose the tolerance, ε_F, to have the very large value 10^5 so that it would play no part in the termination process (for reasons discussed in Section 2).

The basic algorithm (see [2]) took 45 steps to produce a set of 11 boxes covering the solution set. The computing time was 0.18 seconds on a CYBER 175 computer. One output box was the degenerate box equal to the point at the origin. Thus, one subset of the solution set was obtained exactly.

The solution subset consisting of the line segment joining the points $(a, -a/2)$ and $(b', -b'/2)$ was covered by five "solution" boxes. Let $X^{(1)}$ denote the smallest box which can cover these five boxes. We obtained

$$X^{(1)} = \begin{bmatrix} [1.739, 1.896] \\ [0.856, 0.968] \end{bmatrix}.$$

The smallest box covering the line segment of true solution points has components $X_1 = [1.825, 1.893]$ and $X_2 = [0.9128, 0.9462]$. Thus, in a sense, the computed covering was not far from optimal.

The solution subset consisting of the line segment joining the points $(-a, a/2)$ and $(-b', b'/2)$ was quite similarly covered by five "solution" boxes. Let $X^{(2)}$ denote the smallest box containing these five boxes.

The interval computed to bound $f^*(C)$ was $[-4.781, 0]$. The correct interval is $[-1.859, 0]$. This result is far from sharp because we chose the tolerance ε_F to be so large.

The computed "solution" set has disjoint subsets. Therefore, we must be concerned that one or more component may contain a local (non-global) minimum. This is, in fact, the case.

For this example, the origin is the only global solution point when $c > 0.945$. The boxes $X^{(1)}$ and $X^{(2)}$ contain points which are global minima for $c \leqslant 0.945$ and local minima for $c > 0.945$. Therefore, problems (5.2), (5.3), and (5.4) cannot yield sharp bounds.

If we were to solve (5.2) in the box $X^{(1)}$ to obtain an upper bound on $f^*(C)$, we would obtain a value of 1.7918, approximately. This is far from sharp. The correct value is zero.

If we were to solve (5.3) and (5.4) over $X^{(1)}$ for bounds on $X^*(C)$, we would obtain the box

$$\begin{bmatrix} [1.7475, 1.8923] \\ [-0.94612, -0.87377] \end{bmatrix}$$

approximately. The correct box for this subset of $X^*(C)$ to five decimals is

$$\begin{bmatrix} [1.8257, 1.8923] \\ [-0.94612, -0.91287] \end{bmatrix}.$$

Similar results would be obtained using the subset $X^{(2)}$.

Solving (5.1) gives the correct lower bound on $f^*(C)$, which is approximately -1.8589.

Thus, we see that solving for bounds on $f^*(C)$ and $X^*(C)$ using the procedure in Section 5 will give correct bounds. However, some of them are far from sharp.

When the basic algorithm produces disjoint subsets of solution boxes (such as in this example), we have no way of knowing whether the procedure in Section 5 will yield sharp bounds on $f^*(C)$ and $X^*(C)$ or not. The user must decide whether to use this procedure and perhaps obtain less than sharp results or to cover the solution set with small boxes using the basic algorithm.

References

1. Dixon, L. C. W. and Szegö, G. P. (1975), *Towards Global Optimization*, North Holland/ American Elsevier, New York.
2. Hansen, E. R. (1980), Global Optimization Using Interval Analysis – the Multidimensional Case, *Numerische Mathematik*, 247–270.
3. Hansen, E. R. (1984), Global Optimiation with Data Perturbations, *Comput. Ops. Res.* **11**, 97–104.

4. Hansen, E. R. (1988), An Overview of Global Optimization Using Interval Analysis, pp. 289–307 in Moore, R. E. (ed.), *Reliability in Computing*, Academic Press, Boston.
5. Hansen, E. R. and Sengupta, S. (1980), Global Constrained Optimization Using Interval Analysis, pp. 25–47 in Nickel, K. L., (ed.), *Interval Mathematics* 1980, Academic Press, New York
6. Hansen, E. R. and Walster, G. W. (1992), Equality Constrained Global Optimization, accepted for publication.
7. Hansen, E. R. and Walster, G. W. (1992), Nonlinear Equations and Optimization, to appear in the Second Special Issue on Global Optimization, Control and Games of *Comput. Math. Appl.*
8. Hansen, E. R. and Walster, G. W. (1992), Bounds for Lagrange Multipliers and Optimal Points, to appear in the Second Special Issue on Global Optimization, Control and Games of *Comput. Math. Appl.*
9. Moore, R. E., (1979), *Methods and Applications of Interval Analysis*, SIAM Publ., Philadelphia.
10. Moore, R. E., Hansen, E. R., and Leclerc, A. (1992), Rigorous Methods for Parallel Global Optimization, to appear in *Recent Advances in Global Optimization,* Princeton Univ. Press.
11. Neumaier, A. (1990), *Interval Methods for Systems of Equations*, Cambridge Univ. Press.
12. Ratshek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization*, Halstead Press, New York.
13. Walster, G. W., Hansen, E. R., and Sengupta, S. (1985), Test Results for a Global Optimization Algorithm, pp. 272–287 in Boggs, P. T., Byrd, R. H., and Schnabel, R. B. (eds.), *Numerical Optimization 1984*, SIAM Publ.