SOLVING MINIMAX PROBLEMS BY INTERVAL METHODS

S. ZUKER, A. PLEUHANS and M. C. ESHEMANN

Mathematics Department
University of Nancy
Vincennes, France
National Institute of China

Abstract

In the present paper, we consider minimax problems of a class of functions defined on a box $D^n = [a_1, b_1] 	imes [a_2, b_2] 	imes ... 	imes [a_n, b_n]$ in $n$-dimensional space, where $n > 1$. The functions are assumed to be continuous and to have continuous partial derivatives. The minimax problem is to find the maximum value of a function $f(x_1, x_2, ..., x_n)$ subject to the constraint $g(x_1, x_2, ..., x_n) = 0$. The problem is solved using a combination of interval arithmetic and optimization techniques, which ensure that the solution is correct within a user-defined bound.

AMS Classification: Primary 65G30; secondary 65K10, 65K15

Keywords: Minimax problem, interval methods

1. Introduction

Let $f(x) = f(x_1, x_2, ..., x_n)$ be a real-valued function defined on the box $D^n = [a_1, b_1] 	imes [a_2, b_2] 	imes ... 	imes [a_n, b_n]$. We say that $f^*(x_1, x_2, ..., x_n) = (f_{x_1}, f_{x_2}, ..., f_{x_n})$ is a minimum point of $f$ if $f_{x_i}(f^*) = 0$ for all $i = 1, 2, ..., n$.

Then $f^*(x_1, x_2, ..., x_n)$ is a minimax problem for $f$ and the function $f(x_1, x_2, ..., x_n)$ is said to be minimax.

The purpose of this paper is to present a method for solving the minimax problem (1.1). Such problems arise in both mathematics and physics, and the following example shows:

-- This paper has been written with the financial support of the Deutsche Forschungsgemeinschaft and the University of Frankfurt. It has been sponsored by the United States Office of Naval Research, grant N00014-85-C-0107, and the Office of Naval Research, Grant N00014-85-C-0107.

Received June 1985; Revised May 1986.
1.4. Chebyshev approximation:

Given a function \( g : \mathbb{R}^n \rightarrow \mathbb{R} \) and a function space \( \mathcal{P}_n \) of functions \( p_n : \mathbb{R}^n \rightarrow \mathbb{R} \), the Chebyshev approximation \( p_n \) of \( g \) in \( \mathcal{P}_n \) solves the following minimax problem:

\[
\min_{p_n} \max_{x \in \mathbb{R}^n} |p_n(x) - g(x)|^2.
\]

1.2. Game theory:

A game is a triple \((T, \mathcal{I}, \mathcal{K})\) where \( T \) is the set of strategies for player 1 and \( \mathcal{I} \), respectively, and \( \mathcal{K} \) is a real-valued payoff function of \( x \in T \) and \( y \in \mathcal{I} \). Under certain conditions, the optimal strategies for both players solve the saddle point problem:

\[
\max_{y \in \mathcal{I}} \min_{x \in T} f(x, y) = \min_{x \in T} \max_{y \in \mathcal{I}} f(x, y).
\]

1.3. Engineering design problems:

In many engineering design problems, one is interested in minimizing the largest eigenvalue of an \( n \times n \) symmetric matrix-valued function \( A(y) \) of variable \( y \) in \( \mathbb{R}^m \). Let \( A_{ij}(y) \) denote the \( i,j \) element of \( A(y) \). We obtain the minimax problem:

\[
\min_{y \in \mathbb{R}^m} \max_{1 \leq i, j \leq n} A_{ij}(y).
\]

Another example is a process which produces electronic parts within certain tolerances and consider the problem of minimizing the error in manufacture. Specifically, suppose that when a state \( x \) is specified, the process actually produces the state \( x + e \) where \( e \) is the tolerance set. We measure the resulting distortion. Since \( e \) is not known in advance, the worst-case distortion should be estimated, leading to the problem:

\[
\min_{x \in \mathbb{R}^m} f(x, e).
\]

The Mandelbrot problem which occurs in chemical circuit theory (see [2]) is of this kind. Specifically, let \( x = (x_1, \ldots, x_n) \) and

\[
f(x, e) = \sum_{i=1}^n \max \{ e_i, x_i \}; \quad e_i \in \mathbb{R}
\]

we wish to select a vector \( e \) such that

\[
\max_{x \in \mathbb{R}^m} f(x, e) = \min_{x \in \mathbb{R}^m} f(x, e).
\]
2. Interval methods and results.

Mathematical methods can be used to reliably solve optimization problems. The reader may find a list of related references in [13, 12]. A thorough introduction to interval mathematics is given in the book of Moore [8]. In this section we briefly list some concepts of interval methods and related notions which are needed for the problem (1).

Denote by $1B^{1}(iri, ito)$ the set of real intervals, $n$-dimensional interval vectors, and $n$ a real interval matrix respectively, by $x, y, z$ and $n(x)$ the lower bound, upper bound, endpoint, and radius of $x$, respectively, and by $n(x)$ the interior of $x$.

2.1. Interval enclosure.

We call a function $f(x, y) : IR^{1n} \rightarrow IR$ a Lipschitz interval extension of $f : D \subseteq IR^{1n} \rightarrow IR$ on $X \subseteq IR^{1n}$ if

(1) $f(x, y) = f(x) + f(y)$

(2) $f(x, y)$ is an interval matrix for all $x, y \in X$.

(3) $f(x, y)$ is an interval matrix for every $x \in X$.

where $c$ is a positive number. Clearly, each enclosure is an inclusion of the range $f(x) = f(x) + 0 = f(x) + 0 \times x \subseteq f(x) + 0 \times x$.

Throughout this paper we assume that the first and second derivatives of $f$ have Lipschitz interval extensions on $D$. Hence, we get higher order extensions from the second differences.

(1) $f(x, y) \leftarrow f(x) - f(y) = f(x) + f(y) = f(x) + f(y) = f(x) + f(y)$

(2) $f(x, y) = f(x) + f(y) = f(x) + f(y) = f(x) + f(y)$

By Remark 11, we see that the curves in (2) are optimal. For this purpose write $f(x) = (f, f)$ and define the vectors $\ell$ and $\ell$ as

(7) $\ell = f(x) - f(y)$

(8) $\ell = f(x) - f(y)$

(9) $\ell = f(x) - f(y)$

(10) $\ell = f(x) - f(y)$

(11) $\ell = f(x) - f(y)$

then $x = \ell$ is the lower bound and $x = \ell$ the upper bound in (2). For $\ell = f(x) - f(y)$, we see in fact that $f(x)$ is a lower triangular matrix instead of a symmetric matrix, and some of the conclusions of the

2.2. Discontinuity rules.

As the algorithm for the problem (1.1) proceeds we dynamically subdivide $X$ into subintervals, as much as subinterval $X$ is generated we want to discard those points of $X$ in which no minimum point is contained. In this section we will describe the discarding rules.

2.2.1. Necessity test (cf. Step 5 of the algorithm).

Suppose that for some interval $X = (x, y)$ and some index $i 

2.2.2. Necessity test (cf. Step 6 of the algorithm).

Suppose that for some index $i$ and some index $j 

2.3. Evaluation test (cf. Step 7 of the algorithm).

For $i = 1, 2, \ldots, n$, let $f(x)$ be the minimum of $f(x)$ on $X$. Then $f(x)$ is the minimum of $f(x)$ on $X$. We use this to refine or discard $X$ by one or two edge points $X^\prime$.

Note: We have theoretically that there is no minimum point at the boundary of the initial interval $X_0$, we discards $X$ if the necessary test is not satisfied.

Lemma 1. Let $X = (x, y)$ and $f$ is a function.

For all $i, j \in \{1, 2, \ldots, n\}$, then there is no minimum point in the only $X_0$.
certain step of the solution process, then we will solve the linear interval equation

\[ M(x) y + f(y) = 0 \]

To this end we examine the step of the preprocessed Gauss-Seidel iteration,

\[ F = G \cdot C \cdot \frac{\partial g}{\partial y}(y) \cdot f(y) \]

with

\[ A = \frac{\partial g}{\partial y}(y) \]

\[ B = -\frac{\partial g}{\partial y}(y) \cdot \frac{\partial h}{\partial y}(y) \cdot f(y) \]

\[ C = 0 \] (with \( C = 1 \) when \( m = 1 \)).

where \( \frac{\partial g}{\partial y}(y), \frac{\partial h}{\partial y}(y) \) is the Gauss-Seidel operator defined to be the end of the nonnegative bounded set \( \{ x \in X \mid \lambda = n \lambda \} \), obtained by solving in turn the \( i \)th equation for the \( j \)th variable \((j = 0,1, \ldots, n)\), taking interactions with the previous unknowns, cf. Hammer [13], Nesterov [20, 22, 23].

If the solution set \( F \) is empty we discard the interval \( X \). Otherwise we reduce the interval \( X \) by \( \{ \ldots, x_{n-1}, x_0 \} \). \( \ldots, \}

2.4. Data management and update (of step 10 of the algorithm)

From the initial interval \( X_0 = (x_0, x_2)^+ \) we generate a list of subsets \( \mathcal{M}^i \). Each node \( m \) belongs to sets \( \mathcal{M}^i \) common intervals \( x_0 < x_1, x_0 < x_2 \) i.e., they have a common second part \( x_2 \). Hence it suffices to store \( \mathcal{M}^i \) for each interval \( x_0 \) and \( x_2 \) for each subset \( \mathcal{M}^i \). We store furthermore

\[ \mathcal{M}^i(x_0) = \begin{cases} \mathcal{M}^i(x_0) = \mathcal{M}^i(x_0) \cup \mathcal{M}^i(x_2), & j \geq 1 \end{cases} \]

\[ \mathcal{M}^i(x_2) \supseteq \max \mathcal{M}^i(x_0, \max \mathcal{M}^i(x_2)) \]

in the following algorithm:

2.5. Interval Newton method [22, 7]

Suppose that the solution point \( x_1 \) is a minimum point of \( V(x) \) and the interval \( X_1 \) contains \( x_1 \) as a stationary point of \( X_1 \) with respect to \( \gamma \). This last condition is to be satisfied at the solution point \( (x_0, x_2) = X_0 \). By the Taylor expansion we have

\[ \gamma(x_1) = \frac{\partial x_1}{\partial x_1} (x_0, x_2) \cdot \gamma \]

\[ \gamma(x_1) = \frac{\partial x_1}{\partial x_1} (x_0, x_2) \cdot \gamma \]

Suppose that \( X_0 = (x_0, x_2)^+ \) is a new interval generated from \( X_0 = (x_0, x_2)^+ \) at

\[ x_1 = \mathcal{M}^i(x_0, \max \mathcal{M}^i(x_2)) \]
If an interval $X^{\text{old}}$ was not refined enough in the solution process by the discarding rules and the internal Newton method, then we inserted the interval in the direction of maximal width of $X^{\text{old}}$. If $X^{\text{old}} = (x_{\text{low}}, x_{\text{high}})$ is blocked in the z direction we get the interior intervals $X_{\text{low}}^{\text{new}}, X_{\text{high}}^{\text{new}}$, and $(x_{\text{low}}, x_{\text{high}})$ and $(x_{\text{low}}, x_{\text{high}})$, otherwise, we have the entire interval $X^{\text{old}}$, i.e., we get the intervals $X_{\text{low}}$ and $X_{\text{high}}$ with intervals $X_{\text{low}} = (x_{\text{low}}, x_{\text{high}})$ and $(x_{\text{low}}, x_{\text{high}})$, $j = 1, 2$. We insert the resulting intervals and subintervals in the list $L$. 

Finally, at the end of each loop, if necessary, we update the value $p(x)$ as

$$p(x) = \max[p(x), \min(p(x_{\text{low}}), \min(p(x_{\text{low}}), \min(p(x_{\text{high}}), \min(p(x_{\text{low}}), \max(p(x_{\text{high}})))))].$$

This, $p(x)$ and $\min(p(x))$ are always the greatest lower and smallest upper bound of $\mathcal{J}$ currently known.

3. Algorithm.

Step 1. Initialization. Set $X = X^{\text{old}}$. Define a tolerance $\epsilon > 0$. To begin with, let the list $L$ contain one subinterval with the interval $X$. Go to step 3.

Step 2. Begin loop. Select a subinterval $X$ and an interval $X'$ of the list. Set Reduce to false.

Step 3. Evaluation test. $|X| < \epsilon$.

If $f(x) < p(x)$, then discard the interval $X$ and go to step 12.

Step 4. Evaluation test. $f(x) > p(x)$, then discard the subinterval $X$ and go to step 12.

Step 5. Monotonicity test. $f(x_{\text{low}}) > 0 > f(x_{\text{high}})$.

For $i = 1, \ldots, n$ do the following:

Evaluate $f(x_i)$.

If $f(x_i) > 0$, then set $X = \{x_i\}$ and go to Reduce true, $X_{\text{low}} = x_i$, $X_{\text{high}} = x_i$, discard the interval $X$ and go to step 12 otherwise.

Step 6. Non-monotonicity test. $f(x_{\text{low}}) = 0$

For $i = 1, \ldots, n$ do the following:

Evaluate $f(x)$.

If $f(x_i) < 0$ then set $X = \{x_i\}_{i=1}^n$ and Reduce to true if $f_i = 0$.

Step 7. Internal Newton method: Solve $\mathcal{C} = \mathcal{C}(x)$ (for $X = \{x\}$).

Compute $\mathcal{C} = \mathcal{C}(x)$ (for $X = \{x\}$). Set Reduce to true if $X = \{x\}$.

For $i = 1, \ldots, m$ do the following.

Step 8. Evaluation test. Solve $f(x_i) \geq p(x_i)$ for $x_i$.

For $i = 1, \ldots, m$ do the following:

Solve the quadratic equation $\mathcal{C}(x) = 0$.

Step 9. Evaluation test. Solve $f(x_{\text{high}}) \leq p(x_{\text{high}})$ for $x_{\text{high}}$.

For $i = 1, \ldots, m$ do the following:

Solve the quadratic equation $\mathcal{C}(x) = 0$.

Step 10. Evaluation test. Solve $f(x_{\text{low}}) \leq p(x_{\text{low}})$ for $x_{\text{low}}$.

For $i = 1, \ldots, m$ do the following:

Solve the quadratic equation $\mathcal{C}(x) = 0$.

If $X_{\text{low}} < X_{\text{high}}$, then discard the subinterval $X$ and go to step 12.

If $X_{\text{low}} = X_{\text{high}}$, then the solution $X$ is the union of the intervals $X_{\text{low}}$ and $X_{\text{high}}$, set Reduce to true if $X_{\text{low}} < X_{\text{high}}$, set $X_{\text{low}}$ and $X_{\text{high}}$.

If $X_{\text{low}} > X_{\text{high}}$, then the solution $X$ is the union of the intervals $X_{\text{low}}$ and $X_{\text{high}}$, set Reduce to true if $X_{\text{low}} < X_{\text{high}}$, set $X_{\text{low}}$ and $X_{\text{high}}$.
If in the steps 3–10 the interval $X$ was not enough reduced (Reduce is failed),
then block the subinterval $S$ in the direction of maximal width of $X$ and insert
the resulting intervals or subintervals in the list $L$.
Step 13. End loop.
Update the value $Q(l)$ if $Q(l) < Q(l')$, then go to Step 2.

Step 14. Print $Q(l)$
Print all $Q(l)$ in enclosures of minimal value, print remaining intervals of
subintervals of list $L$:  in the message of the branches process.

If a reasonable tolerance of the algorithm is given, the computation will always stop after a finite number of steps. This holds since the order of the list is of decreasing width and the forward rule of the intervals is in the function of maximum width, so that the width of the interval generated by the algorithm above from the initial interval $X_0$ tends to zero as the process proceeds. Through the descending rules and the interval Newton method, locally only a few intervals and subintervals remain in the list. It is not difficult to show that this implies that $Q(l) - x^{*}$ must become arbitrarily small if the tolerance is small enough.

4.1. Example.
Solve the minimization problem with
$$f(x,0) = g_1 - y_1 x - y_2 - y_3,$$
which has the exact solution
$$x^* = (1, 0, 1, 0, 1).$$
With a specified error tolerance of $x = (E = 10^{-8})$, the minimum value was estimated by the interval $233418144183182 (E = 2)$, that a width of 11 machine numbers; the total width of the boxes enclosing the two minimum points was $3318 - 27,321$ loops taken with maximally 23 boxes and 4 subintervals at a time.

4.2. Example.
(Mandelbrot's Problem). Solve the minimization problem with
$$f(x,0) = x^N + \sum_{i=1}^{n} x_i \sin \left( \frac{x_i}{2} \right)$$
subject to
$$y \in Y^0 \cap \{ -1, 1 \},$$
for $0 < 0.002 \times 2^{N} = (1, x, 0)_{12} < 1.2$.

References