INTERVAL ANALYSIS: UNCONSTRAINED AND CONSTRAINED OPTIMIZATION

Introduction.

Interval algorithms for constrained and unconstrained optimization are based on adaptive, exhaustive search of the domain. Their overall structure is virtually identical to *Lipschitz optimization* as in [4], since interval evaluations of an objective function ϕ over an interval vector $\boldsymbol{x}$ correspond to estimation of the range of ϕ over $\boldsymbol{x}$ with Lipschitz constants. However, there are additional opportunities for acceleration of the process with interval algorithms, and use of outwardly rounded interval arithmetic gives the computations the rigor of a mathematical proof.

The interval algorithms are both complicated and accelerated by the presence of constraints, as is explained below.

See **Interval analysis: Introduction, interval numbers, and basic properties of interval arithmetic** for background on interval computations. See [5], [2] or [3] for further details of concepts in this article.

The basic problem is

$$\boxed{\begin{array}{ll} \text{minimize} & \phi(x) \\ \text{subject to} & \left\{ \begin{array}{lcll} c(x) & = & 0 & \text{and} \\ g(x) & \leq & 0, & \end{array} \right\} \end{array}} \qquad (1)$$

where $\phi: \boldsymbol{x} \subset \mathbf{R}^n \to \mathbf{R}$, $c: \boldsymbol{x} \to \mathbf{R}^{m_1}$, and $g: \boldsymbol{x} \to \mathbf{R}^{m_2}$, where $\boldsymbol{x}$ is an interval vector

$$\boldsymbol{x} = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])^T.$$

The values $m_1 = 0$ and $m_2 = 0$ will be allowed, in which case the problem is considered to be unconstrained. It is emphasized here that, in problem 1, a *global* optimum, that is the lowest possible value of ϕ over the feasible set, is sought.

The basic branch and bound algorithm for unconstrained optimization.

The overall outline of an interval branch and bound algorithm for unconstrained global optimization is as follows:

INPUT: *an initial box* $\boldsymbol{x}_0$.

OUTPUT: *a list* `C` *of boxes that have been proven to contain critical points and a list* `U` *of boxes with small objective function values, but which could not otherwise be resolved.*

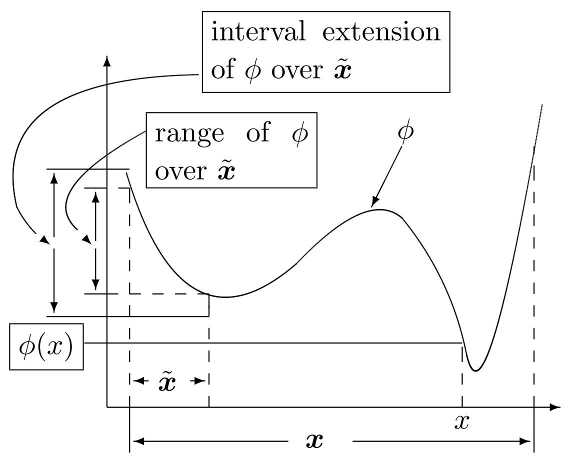
1. *Initialize a list of boxes* `L` *by placing the initial search region* $\boldsymbol{x}_0$ *in* `L`.
2. DO WHILE `L` $\neq \emptyset$.
   (a) *Remove the first box* $\boldsymbol{x}$ *from* `L`. *(The boxes in* `L` *are in general inserted in a particular order, depending on the actual algorithm.)*
   (b) (Process $\boldsymbol{x}$) *Do one of the following:*
      - *reject* $\boldsymbol{x}$*;*
      - *reduce the size of* $\boldsymbol{x}$*;*
      - *determine that* $\boldsymbol{x}$ *contains a unique critical point, then find the critical point to high accuracy;*
      - *subdivide* $\boldsymbol{x}$ *to make it more likely to succeed at rejecting, reducing, or verifying uniqueness.*
   (c) *Do the following to the box(es) resulting from Step 2b:*
      - *If* $\boldsymbol{x}$ *was rejected, do nothing.*
      - *If more than one box was derived from* $\boldsymbol{x}$*, insert all but one of them into* `L`. *Call the remaining box derived from* $\boldsymbol{x}$ $\tilde{\boldsymbol{x}}$.
      - *If there is a* $\tilde{\boldsymbol{x}}$ *that has been proven to contain a critical point, insert it into* `C`.
      - *If there is a* $\tilde{\boldsymbol{x}}$ *that is small, but has not been proven to contain a feasible point, insert it into* `U`.

   END DO

One way that a box is rejected in step 2b is by using a bound on the range of the function ϕ over the interval vector (box) $\boldsymbol{x}$. In particular, suppose the value $\phi(x)$ at a point x is known. Then $\phi(x)$ is an upper bound for the global optimum. (In fact, if ϕ has been evaluated at various points, then the minimum of the resulting values is a usable upper bound on the global optimum.) Now suppose a lower bound $\underline{\phi}$ on the range of ϕ over a box (or more generally, a region) $\boldsymbol{x} \subset \mathbf{R}^n$

Lipschitz optimization
Interval analysis: Introduction, interval numbers, and basic properties of interval arithmetic

can be computed, and that $\underline{\phi} > \phi(x)$. Then there cannot be any global optimizers of ϕ within $\boldsymbol{x}$. The value $\underline{\phi}$ can be obtained through an interval function value. This process is illustrated in the following figure.



The midpoint test: Rejecting $\tilde{\boldsymbol{x}}$ because of a high objective value

The lower bound $\underline{\phi}$ for the objective over the box $\boldsymbol{x}$ need not be obtained via interval computations. Indeed, if a Lipschitz constant $L_{\boldsymbol{x}}$ for ϕ is known over $\boldsymbol{x}$, and $\phi(\check{x})$ is known for $\check{x}$, the center of $\boldsymbol{x}$, then, for any $\tilde{x} \in \boldsymbol{x}$,

$$\phi(\tilde{x}) \geq \phi(\check{x}) - \frac{1}{2} L_{\boldsymbol{x}} \, \|\mathrm{w}(\boldsymbol{x})\| \, ,$$

where $\mathrm{w}(\boldsymbol{x})$ is the vector of widths of the components of the interval vector $\boldsymbol{x}$. However, getting rigorous bounds on Lipschitz constants can require more human effort than the interval computation, and often results in bounds that are not as sharp as those from interval computation. Similarly, automated computations for Lipschitz constants as presently formulated result in bounds that are provably not as sharp as interval computations. Furthermore, use of properly rounded interval arithmetic, if used both in computing $\underline{\phi}$ and $\phi(x)$, allows one to conclude *with mathematical rigor* that there are no global optima of ϕ within $\boldsymbol{x}$.

Use of this lower bound for ϕ is sometimes called the *midpoint test*, since the points x at which $\phi(x)$ is evaluated are often taken to be the vectors of midpoints of the boxes $\boldsymbol{x}$ produced during the subdivision process. (Actually, some implementations use the output of an approximate or local optimizer as x, to get an upper bound on the global optimum that is as low as possible.)

The simplest possible branch and bound algorithms need to contain both a box rejection mechanism and a subdivision mechanism. A common subdivision mechanism is to form two sub-boxes by bisecting the widest coordinate interval of $\boldsymbol{x}$ (with possible scaling factors). Heuristics and scaling factors, as well as several references to the literature, appear in [3, §4.3.2, p 157 ff.]. Alternatives to bisection, such as trisection, forming two boxes by cutting other than at a midpoint, etc. have also been discussed at conferences and studied empirically [1].

Acceleration tools.

Early and simple algorithms contain *only* the midpoint test mechanism and bisection mechanism described above. Such algorithms produce as output a large list `U` of small boxes (with diameters smaller than a stopping tolerance) and no list `C` of boxes that contain verified critical points. The list `U` in such algorithms contains clusters of boxes around actual global optimizers. Some Lipschitz constant-based algorithms are of this form. Note, however, that such algorithms are of limited use in high dimensions, since the number of boxes produced increases exponentially in the dimension n.

Interval computations provide more powerful tools for accelerating the algorithm. For a start, if an interval extension of the gradient $\nabla\phi(\boldsymbol{x})$ is computable then $0 \notin \nabla\phi(\boldsymbol{x})$ implies that $\boldsymbol{x}$ cannot contain a critical point, and $\boldsymbol{x}$ can be rejected. This tool for rejecting a box $\boldsymbol{x}$ is sometimes called the *monotonicity test*, since $0 \notin \left(\nabla\phi(\boldsymbol{x})\right)_i$ implies ϕ is monotonic over $\boldsymbol{x}$ in the i-th component x_i, where $\left(\nabla\phi(\boldsymbol{x})\right)_i$ represents the i-th component of the interval evaluation of the gradient $\nabla\phi$.

However, heuristically obtained approximate Lipschitz constants, as employed in the calculations in [4], have been highly successful at solving practical problems, albeit not rigorously.

midpoint test

monotonicity test

Perhaps the most powerful interval acceleration tool is *interval Newton methods*, applied to the system $\nabla\phi = 0$. Interval Newton methods can result in quadratic convergence to a critical point in the sense that the widths of the coordinates of the image of $\boldsymbol{x}$ are proportional to the square of the widths of the coordinates of $\boldsymbol{x}$. Interval Newton methods also can prove existence and uniqueness of a critical point or non-existence of a critical point in $\boldsymbol{x}$. Thus, the need to subdivide a relatively large $\boldsymbol{x}$ is often eliminated, making a previously impractical algorithm practical. See **Interval Analysis: Interval Newton methods** and **Interval analysis: Interval Fixed Point Theory**.

For a more detailed algorithm, and for a discussion of parallelization of the branch and bound process, see **Interval analysis : Parallel methods for global optimization**.

Differences between unconstrained and constrained optimization.

If $m_1 > 0$ or $m_2 > 0$ in problem 1, then the problem is one of *constrained optimization*. The midpoint test cannot be applied directly to constrained problems, since $\phi(x)$ is guaranteed to be an upper bound on the global optimum only if the constraints $c(x) = 0$ and $g(x) \le 0$ are also satisfied at x. If there are only inequality constraints and none of the inequality constraints are active at x, then an interval evaluation of $g(x)$ will rigorously verify $g(x) < 0$, and x can be used in the midpoint test. However, if there are equality constraints (or if one or more of the inequality constraints is active), then an interval evaluation will yield $0 \in \boldsymbol{c}(x)$ (or $0 \in \boldsymbol{g}_i(x)$ for some i), and it cannot be concluded that x is feasible. In such cases, a small box $\check{\boldsymbol{x}}$ can be constructed about x, and it can be verified with interval Newton methods that $\check{\boldsymbol{x}}$ contains a feasible point. The upper bound of the interval evaluation $\phi(\check{\boldsymbol{x}})$ then serves as an upper bound on the global optimum, for use in the midpoint test. For details and references, see **Interval analysis: Verifying feasibility**.

On the other hand, constraints can be beneficial in eliminating infeasible boxes $\boldsymbol{x}$. In particular, $0 \notin c(\boldsymbol{x})$ or $g(\boldsymbol{x}) > 0$ implies that $\boldsymbol{x}$ can be rejected.

It is sometimes useful to consider *bound constraints* of the form $x_i \ge \underline{x}_i$ and $x_j \le \overline{x}_j$ separately from the general inequality constraints $g(x) \le 0$. Such bound constraints can generally coincide with the limits on the search region $\boldsymbol{x}_0$, but are distinguished from simple search bounds. (It is possible for an unconstrained problem to have no optima within a search region, but it is not possible if all of the search region limits represent bound constraints.) See **Interval analysis : Bound constrained problems** and [3, §5.2.3, p. 180 ff] for details.

An illustrative example.

Consider

$$\begin{array}{|lrcl|}\hline \text{minimize} & \phi(x) & = & -(x_1+x_2)^2 \\ \text{subject to} & c(x) & = & x_2+2x_1=0. \\ \hline\end{array} \qquad (2)$$

Example 2 represents a constrained optimization problem with a single equality constraint and no bound constraints or inequality constraints. To apply the midpoint test in a rigorously verified algorithm, a box must first be found in which a feasible point is verified to exist. Suppose that a point algorithm, such as a generalized Newton method, has been used to find an approximate feasible point, say $\check{x} = \left(-\frac{1}{4}, \frac{1}{2}\right)^T$. Now observe that $\nabla c \equiv (-2,1)^T$. Therefore, as suggested in **Interval analysis: Verifying feasibility**, x_2 can be held fixed

interval Newton methods
Interval Analysis: Interval Newton methods
Interval analysis: Interval Fixed Point Theory
Interval analysis : Parallel methods for global optimization
constrained optimization
Interval analysis: Verifying feasibility
bound constraints
Interval analysis : Bound constrained problems
Interval analysis: Verifying feasibility

at $x_2 = \frac{1}{2}$. Thus, to prove existence of a feasible point in a neighborhood of $\check{x}$, an interval Newton method can be applied to $f(x_1) = c(x_1, 0.5) = 0.5 - 2x_1$. We may choose initial interval $\boldsymbol{x}_1 = [-0.25 - \epsilon, -0.25 + \epsilon]$ with $\epsilon = 0.1$, to obtain

$$\begin{aligned} \boldsymbol{x}_1 &= [-.35, -.15], \\ \tilde{\boldsymbol{x}}_1 &= -0.25 - \frac{0}{-2} \\ &= [-0.25, -0.25] \subset \boldsymbol{x}_1, \end{aligned}$$

This computation proves that, for $x_2 = 0.5$, there is a feasible point for $x_1 \in [-0.25, -0.25]$. (See **Interval Analysis: Interval Newton methods** and **Interval analysis: Interval Fixed Point Theory**.) We may now evaluate ϕ over the box $([-0.25, -0.25], [0.5, 0.5])^T)$ (that is degenerate in the second coordinate, and also happens to be degenerate in the first coordinate for this example). We thus obtain

$$\phi\big([-0.25, -0.25], [0.5, 0.5]\big) = \left[-\frac{1}{16}, -\frac{1}{16}\right],$$

and $-1/16$ has been proven to be an upper bound on the global optimum for example problem 2.

References

[1] Csallner, A. E., and Csendes, T., Convergence Speed of Interval Methods for Global Optimization and the Joint Effects of Algorithmic Modifications, 1995, Talk given at SCAN'95, Wuppertal, Germany, Sept. 26 – 29, 1995.

[2] Hansen, E. R.: *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., New York, 1992.

[3] Kearfott, R. B.: *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, Netherlands, 1996.

[4] Pinter, J. D.: *Global Optimization in Action: Continuous and Lipschitz Optimization*, Kluwer, Dordrecht, Netherlands, 1995.

[5] Ratschek, H., and Rokne, J.: *New Computer Methods for Global Optimization*, Wiley, New York, 1988.

R. Baker Kearfott
Department of Mathematics
University of Southwestern Louisiana
U.S.L. Box 4-1010, Lafayette, LA 70504-1010 USA
E-mail address: rbk@usl.edu

Interval Analysis: Interval Newton methods
Interval analysis: Interval Fixed Point Theory

AMS 1991 Subject Classification: 65G10, 65H20.
Key words and phrases: constrained optimization, automatic result verification, interval computations, global optimization.