Computer Assisted Mechanics and Engineering Sciences, 7: 699–705, 2000.

Optimization of mechanical structures using interval analysis

Andrzej Pownuk
Department of Theoretical Mechanics, Silesian University of Technology
ul. Krzywoustego 7, 44-100 Gliwice, Poland

(Received November 3, 1999)

The problem of optimal design consists in finding the optimum parameters according to a specified optimality criterion. Existing optimization methods [25, 27] usually are not reliable or cannot use the nondifferentiable, not continuous objective functions or constraints. An interval global optimization method is very stable and robust, universally applicable and fully reliable. The interval algorithm guarantees that all stationary global solutions have been found. In this paper the algorithm is applied to optimization of mechanical systems, calculation of extreme values of mechanical quantities and to optimization of structures with uncertain parameters.

1. Introduction

Algorithms for solving global optimization problems can be classified into heuristic methods that find the global optimum only with high probability, and methods that guarantee to find a global optimum with some accuracy. An important class belonging to the former type are the stochastic methods [25]. A number of techniques such as simulated annealing [25] and genetic algorithms [16] use analogies to physics and biology to approach the global optimum. The most important class of methods of the second type are branch and bound methods [11]. They originate from combinatorial optimization, where global optima are also wanted but the variables are discrete and take several values only. Branch and bound methods guarantee to find a global optimum with a desired accuracy after a predictable (though often exponential) number of steps. The basic idea is that the configuration space is split repeatedly by branching into smaller and smaller parts. This is not done homogenously, but instead some parts are preferred and others are eliminated. The details depend on bounding procedures. Lower bounds on the objective allow to eliminate large portions of the configuration space early in the computation so that only a (usually small) part of the branching tree has to be generated and processed. The lower bounds may be obtained using dc-method [25], techniques of the interval analysis [10, 23], or methods based on the knowledge of Lipshitz constants [19]. In many mechanical and civil engineering optimization problems, sensitivity and gradient methods are applied [2, 8, 15, 26]. If an objective function and constraints are differentiable, Kuhn–Tucker conditions can be applied [6]. Many information about optimization methods is available on the Internet [27, 28].

In this paper, applications of the interval global optimization method [10, 23] to the optimization problems in applied mechanics are considered. This method can also be applied to the following engineering problems:

– optimization of parameters of mechanical system [8];

– shape optimization [21];

– calculation of extreme values of mechanical system [20].

2. Interval analysis

2.1. Interval arithmetic

A real interval is a set of real numbers such that

$$[x] = [x^-, x^+] = \{x \in R: \ x^- \le x \le x^+\}. \tag{1}$$

The set of all intervals is denoted by IR [1] and called a real interval space. Operations and functions on real numbers are naturally extended to interval operands according to the general formula [1, 18]

$$[x] \oplus [y] = \{x \oplus y: \ x \in [x], \ y \in [y]\}, \quad \text{where} \quad \oplus \in \{+, -, \cdot, /\}, \tag{2}$$

$$f([x_1], \ldots, [x_n]) = \{f(x_1, \ldots, x_n): \ x_1 \in [x_1], \ldots, x_n \in [x_n]\}. \tag{3}$$

Multidimensional intervals can be defined in the following way

$$[x] = [x_1^-, x_1^+] \times [x_2^-, x_2^+] \times \cdots \times [x_n^-, x_n^+] \in IR^n. \tag{4}$$

We call a function f *programmable* if f can be built up from arithmetic, logical and comparison operators and some collection of standard transcendental functions (like *sin*, *cos*, *power*, etc.). Particularly, taking an argument x, the function value $f(x)$, can be calculated with a finite number of operations [9]. All the functions dealt with in this paper are assumed to be programmable.

Another important property of arithmetic operations on intervals is called *inclusion isotonicity*,

$$([a] \subseteq [c]) \wedge ([b] \subseteq [d]) \Rightarrow [a] \oplus [b] \subseteq [c] \oplus [d] \tag{5}$$

that is, the result of straightforward calculation of interval expression will always include the proper result ($\oplus$ is any interval arithmetic operation).

A function $F: IR \to IR$ which satisfies

$$\{f(x): \ x \in [x]\} \subseteq F([x]) \qquad \text{for all} \ \ [x] \in IR \tag{6}$$

will be called an inclusion function for f over $[x]$.

Let $[x] \in IR$; then the natural interval extension $\hat{f}([x])$ of the programmable function f to $[x]$ is defined as that expression which is obtained from the expression $f(x)$ by replacing each occurrence of the variable x by the interval $[x]$, the arithmetic operations of IR by the corresponding interval arithmetic operations, and each occurrence of pre-declared function g by the corresponding inclusion function [9, 17]. It follows from the inclusion isotonicity of interval arithmetic [1, 17] operations that:

$$x \in [x] \quad \text{implies} \quad f(x) \in \hat{f}([x]). \tag{7}$$

Property (7) is the key to almost all interval arithmetic applications and results and (7) is called the *fundamental property of interval arithmetic* [23].

For any bounded set of real numbers S we can define the *smallest interval enclosure* of the set [14]

$$\operatorname{hull} S = [\inf S, \sup S]. \tag{8}$$

2.2. Systems of linear interval equations

Let us consider a linear interval system of equations with an interval coefficient matrix $[\mathbf{A}] \in IR^{n\times n}$ and an interval right-hand vector $[\mathbf{B}] \in IR^n$ [18]

$$[\mathbf{A}]\mathbf{X} = [\mathbf{B}] \tag{9}$$

The solution set of equations (9) is defined as [18]

$$\sum([\mathbf{A}], [\mathbf{B}]) = \{\mathbf{X} \in R^n: \quad \exists \mathbf{A} \in [\mathbf{A}], \ \exists \mathbf{B} \in [\mathbf{B}], \ \mathbf{A} \cdot \mathbf{X} = \mathbf{B}\}. \tag{10}$$

2.3. Interval Newton method

Consider a system of nonlinear equations in the form $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{g} : R^n \supset [\mathbf{x}] \to R^n$. We define an iterative algorithm in the form [10]

$$\hat{\mathbf{N}}(\mathbf{x}_k, [\mathbf{x}_k]) = \text{hull} \sum \left(\frac{\partial \hat{\mathbf{g}}(\mathbf{x}_k, [\mathbf{x}_k])}{\partial \mathbf{x}}, -\mathbf{g}(\mathbf{x}_k) \right), \tag{11}$$

$$[\mathbf{x}_{k+1}] = \mathbf{x}_k + \hat{\mathbf{N}}(\mathbf{x}_k, [\mathbf{x}_k]), \qquad \text{where} \quad \mathbf{x}_k \in [\mathbf{x}_k], \tag{12}$$

for $k = 0, 1, \ldots$; ($[\mathbf{x}_0]$ is a given initial interval and $\partial \hat{\mathbf{g}}(\mathbf{x}_k, [\mathbf{x}_k])/\partial \mathbf{x}$ is a special interval extension of Jacobi matrix [10]). For example, in two dimensional cases the Jacobi matrix can be written as

$$\frac{\partial \hat{\mathbf{g}}(\mathbf{x}_k, [\mathbf{x}_k])}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial \hat{g}_1(x_1, [x_2])}{\partial x_1} & \dfrac{\partial \hat{g}_1([x_1], [x_2])}{\partial x_2} \\ \dfrac{\partial \hat{g}_2([x_1], [x_2])}{\partial x_1} & \dfrac{\partial \hat{g}_2(x_1, [x_2])}{\partial x_2} \end{bmatrix}. \tag{13}$$

Using the interval Newton method all solutions of a system of nonlinear equations $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ in a given initial interval $[\mathbf{x}_0]$ can be found [1, 10, 18].

3. Basic algorithm of interval global optimization

The interval global optimization method is based on the fundamental property of interval arithmetic (7). If the following inequality holds,

$$\hat{f}([\mathbf{x}_1])^+ < \hat{f}([\mathbf{x}_2])^-, \tag{14}$$

where

$$[\hat{f}([\mathbf{x}])^-, \hat{f}([\mathbf{x}])^+] = \text{hull}\, \hat{f}([\mathbf{x}]), \qquad [\mathbf{x}_1], [\mathbf{x}_2] \in IR^n, \qquad f : R^n \supset [\mathbf{x}] \to R,$$

then the global minimum does not lie in the interval $[\mathbf{x}_2]$, and hence $[\mathbf{x}_2]$ can be omitted in future calculations. For example, let $f(\mathbf{x}) = x_1^2 + x_2^2$ and $[\mathbf{x}_1] = [0, 1] \times [0, 1]$ and $[\mathbf{x}_2] = [2, 3] \times [2, 3]$. Then $\hat{f}([\mathbf{x}_1]) = [0, 2]$, $\hat{f}([\mathbf{x}_2]) = [8, 18]$. Because $\hat{f}([\mathbf{x}_1])^+ < \hat{f}([\mathbf{x}_2])^-$, then from the fundamental property of interval arithmetic it follows that

$$\forall \breve{x} \in [\mathbf{x}_1]\ \forall \tilde{x} \in [\mathbf{x}_2] \quad f(\breve{x}) < f(\tilde{x}), \tag{15}$$

hence the global minimum of the function f does not lie in the interval $[\mathbf{x}_2]$ and $[\mathbf{x}_2]$ can be omitted in future calculations.

Let $[\mathbf{x}] \in IR^n$ be an initial interval. The basic algorithm is as follows [4, 10, 23]:

Step 0 Set $[\mathbf{y}] = [\mathbf{x}]$ and $y = \hat{f}([\mathbf{x}])^-$. Initialize the list $\mathrm{L} = (([\mathbf{y}], y))$ and the cut-off level $z = \hat{f}([\mathbf{x}])^+$.

Step 1 Choose a coordinate direction $k \in \{1, 2, \ldots, n\}$.

Step 2 Bisect $\mathbf{y}$ in direction k: $[\mathbf{y}] = [\mathbf{v}_1] \cup [\mathbf{v}_2]$.

Step 3 Calculate $\hat{f}([\mathbf{v}_1])$ and $\hat{f}([\mathbf{v}_2])$ and set $v_i = \hat{f}([\mathbf{v}_i])^-$ for $i = 1, 2$ and $z = \min\{z, \hat{f}([\mathbf{v}_1])^+, \hat{f}([\mathbf{v}_2])^+\}$.

Step 4 Remove $([\mathbf{y}], y)$ from the list L.

Step 5 Cutoff test: discard the pair $([\mathbf{v}_i], v_i)$ if $v_i > z$ (where $i = 1, 2$).

Step 6 Add any remaining pair(s) to the list L. If the list becomes empty then STOP.

Step 7 Denote the pair with the smallest second element by $([\mathbf{y}], y)$.

Step 8 If the width of $\hat{f}([\mathbf{y}])$ is less than ε, then print $\hat{f}([\mathbf{y}])$ and $[\mathbf{y}]$, STOP.

Step 9 Go to step 1.

4. ACCELERATION DEVICES – TECHNIQUES FOR SPEEDING UP THE COMPUTATION

In order to improve the quality of the interval global optimization algorithm, several special procedures should be applied.

Monotonicity test – if

$$0 \notin \frac{\partial \hat{f}([\mathbf{x}])}{\partial x_j} \quad \text{for some } j \in \{1, \ldots, n\} \tag{16}$$

then there is no stationary point of f in $[\mathbf{x}]$. In particular, $[\mathbf{x}]$ cannot contain the global minimum.

Midpoint test – if the following inequality holds (compare [14])

$$\hat{f}(\mathbf{x})^+ < \hat{f}([\mathbf{x}_2])^- \quad \text{where } \mathbf{x} \in [\mathbf{x}_1] \tag{17}$$

then global minimum cannot be in the interval $[\mathbf{x}_2]$.

Finding a function value as small as possible – midpoint test gives better results if the number $f(\mathbf{x})$ is as small as possible. Minimum of the function f in $[\mathbf{x}_1]$ can be found using any local optimization method.

The interval Newton method – the interval Newton method (see 2.3) is applied to $[\mathbf{x}]$ to determine existence or uniqueness of a zero of the gradient of the function f.

Use a good inclusion function – in a calculation it is better to use an inclusion function rather than the natural interval extension [1, 22].

A nonconvexity check – if the function f has unconstrained minimum at $\mathbf{x}^*$, then f must be convex in some neighborhood of $\mathbf{x}^*$. Hence, the Hessian $\mathbf{H}$ of f must be positive semidefinite at $\mathbf{x}^*$. A necessary condition for this is that the diagonal elements H_{ii} $(i = 1, \ldots, n)$ be nonnegative. Consider an interval $[\mathbf{x}]$. If $\hat{H}_{ii}([\mathbf{x}])^+ < 0$ for some $i = 1, \ldots, n$ then $H_{ii}(\mathbf{x}) < 0$ for all $\mathbf{x} \in [\mathbf{x}]$. Hence, $\mathbf{H}$ cannot be positive semidefinitive for any point in $[\mathbf{x}]$. Therefore, f cannot have a stationary minimum in $[\mathbf{x}]$ and $[\mathbf{x}]$ can be deleted [10].

Recursive (automatic) differentiation – using automatic differentiation we can calculate derivatives of very complicated functions [18].

The Fritz–John conditions – this procedure is very similar to Kuhn–Tucker conditions [6] and we use them in constrained global optimization problems.

Use of constraints – using fundamental property of interval arithmetic (7) we can verify if in a given interval $[\mathbf{x}]$ any points that satisfy all constraints exist [10].

5. Applications of interval global optimization

The MacNeal–Schwendler company applies interval global optimization to design rocket nozzles [3, 8]. Sun Microsystems developed software and applications for interval techniques for global solutions of non-linear systems and optimization [5]. There are many applications of interval methods in economy (Bank One [24], Swiss National Bank [3]). GE Medical System uses interval methods to control NMR signal and "black-box" optimization [7]. Delisoft Ltd. commercialises and sells interval global optimization software [12].

6. Optimization of mechnical system using interval global optimization

Interval global optimization method can be applied to unconstrained and constrained optimization problems of the following types,

$$\begin{cases} \min f(\mathbf{x}) \\ \mathbf{x} \in [\mathbf{x}] \end{cases} \quad \text{or} \quad \begin{cases} \min f(\mathbf{x}) \\ p_i(\mathbf{x}) \geq 0 \quad \text{for } i = 1, \ldots, m \\ q_j(\mathbf{x}) = 0 \quad \text{for } j = 1, \ldots, n \\ \mathbf{x} \in [\mathbf{x}] \end{cases} \tag{18}$$

where $[\mathbf{x}] \in IR^n$. We can apply this algorithm to engineering problems which have the form (18). Because the interval global optimization method is usually NP-hard [13] the dimensionality of the problem (18) should be sufficiently low.

6.1. Shape optimization of a truss

Consider a problem of shape optimization of a truss structure shown in Fig. 1. The objective function will be the volume (weight) of the truss i.e.

$$f(\mathbf{x}) = \sum_{i=1}^{4} A_i \cdot L_i = \frac{1}{\sigma_0} \sum_{i=1}^{4} |N_i| \cdot L_i \tag{19}$$

where $\mathbf{x} = (x_w, y_w, N_1, N_2, N_3, N_4)$, N_i are axial forces, A_i are areas of cross sections, L_i are lengths of the rods and σ_0 is an allowable stress. This is a global optimization problem in the following form,

$$\begin{cases} \min f(\mathbf{x}) \\ \mathbf{Eq}(\mathbf{x}) = \mathbf{0} \\ \mathbf{x} \in [\mathbf{x}] \end{cases} \tag{20}$$

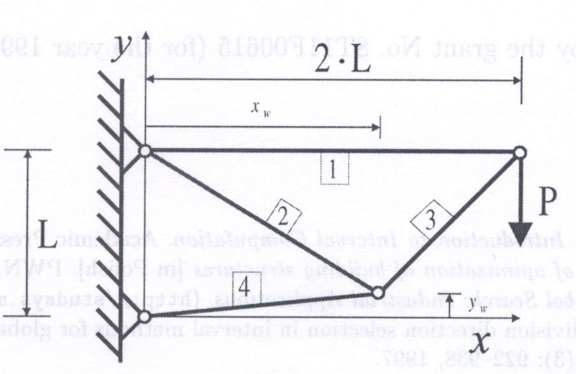

Fig. 1. Optimal shape of a truss

where $\mathbf{Eq}(\mathbf{x}) = \mathbf{0}$ is a system of equilibrium equations ($\mathbf{Eq}$ is a nonlinear function of $\mathbf{x}$) [21]. After solving equilibrium equations we can transform constrained global optimization problem (20) into nconstrained global optimization problem in the form

$$\min f(x_w, y_w) \quad \text{where } x_w \in [x_{w0}],\ y_w \in [y_{w0}]. \tag{21}$$

The objective function (21) is nondifferentiable and we cannot use traditional optimization algorithms [2]. In calculation, we assume that $L = 1$ [m], $\sigma_0 = 190$ [MPa], $P = 10$ [kN], $[x_{w0}] = [0, 2]$ [m], $[y_{w0}] = [0, 1]$ [m]. Numerical results are shown in Table 1.

Table 1. Optimal characteristics of the truss structure

x_w [m] (the best point solution)	1.19
$[x_w]$ [m] (the interval solution)	[1.124272, 1.218754]
y_w [m] (the best point solution)	0.22
$[y_w]$ [m] (the interval solution)	[0.183575, 0.245975]
A_1 [m^2]	$5.4 \cdot 10^{-5}$
A_2 [m^2]	$6.01 \cdot 10^{-5}$
A_3 [m^2]	$0.6 \cdot 10^{-5}$
A_4 [m^2]	$1.0 \cdot 10^{-4}$
Optimal volume f^* [m^3]	$4.102 \cdot 10^{-4}$

The point solutions were calculated in each iteration of the interval algorithm (using the midpoints of appropriate intervals). The best point solution is presented in Table 1. Calculation was carried out by a program written in C++ language. Overloaded operators were used in programming interval arithmetic operations (2).

7. Conclusions

The preceding numerical results indicate that the presented algorithm is an effective and efficient method of global optimization. However, it will probably be quite slow if many local minima have values of f differing very little from the global value. The algorithm guarantees that all stationary global solutions (in the initial interval) have been found. The bounds on the solution(s) are guaranteed to be correct. Error from all sources are accounted for. Algorithm will be faster when interval arithmetic is available in hardware [10]. The algorithm can solve the global optimization problem also when the objective function is nondifferentiable or even not continuous.

Acknowledgements

This work was supported by the grant No. 8T11F00615 (for the year 1999) from the Committee of Scientific Research (KBN).

References

[1] G. Alefeld, J. Hertzberger. *Introduction to Interval Computation.* Academic Press, New York, 1983.
[2] A.M. Brandt. *Fundament of opimization of building structures* [in Polish]. PWN, Warsaw 1978.
[3] G.F. Corliss. *Rigorous Global Search: Industrial Applications.* (http://studsys.mscs.mu.edu/~georgec), 1998.
[4] T. Csendes, D. Ratz. Subdivision direction selection in interval methods for global optimization. *SIAM Journal on Numerical Analysis*, **34**(3): 922–938, 1997.
[5] *I Delivering Performance on Sun: Optimizing Applications for Solaris.* Technical White Paper, SUN Microsystems (http://www.sun.com), 1999.

[6] K. Dems, W. Gutkowski. 2D shape optimization with static and dynamic constraints. *Structural Optimization*, **15**: 201–207, 1998.

[7] X. Feng, R. Yang, Y. Zhu, G.F. Corliss, R.B. Kearfott. GlobSol Case Study: Parameter Optimization for the Eddy Current Compensation of MRI Coils (GE Medical). *Rigorous Global Search Working Note 8* (`http://www.mscs.mu.edu/~globsol/papers.html`), 1998.

[8] F. Fritz, G.F. Corliss, A. Johnson, D. Prohaska, J. Hart. GlobSol Case Study: Rocket Nozzle Design (MacNeal–Schwendler). *Rigorous Global Search Working Note 5* (`http://www.mscs.mu.edu/~globsol/papers.html`), 1998.

[9] Ch. Görges, H. Ratschek. Global interval methods for local nonsmooth optimization. *Journal of Global Optimization*, **14**: 157–179, 1999.

[10] E.R. Hansen. *Global optimization using interval methods.* Marcel Dekker, New York 1992.

[11] W. Huyer, A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, **14**: 331–355, 1999.

[12] E. Hyvönen, S. De Pascale. *GIA InC++ Global Interval Arithmetic and Optimization Library.* Delisoft Ltd., (`http://www.delisoft.fi`), 1997.

[13] V. Kreinovich, A. Lakeyev, J. Rohn, P. Kahl. *Computational Complexity and Feasibility of Data Processing and Interval Computations.* Kluwer Academic Publishers, Dordrecht, 1997.

[14] Z. Kulpa, A. Pownuk, I. Skalna. Analysis of linear mechanical structures with uncertainties by means of interval methods. *CAMES*, **5**(4): 443–477, 1998.

[15] E. Lund. *Finite Element Based Design Sensitivity Analysis and Optimization.* Ph.D. Dissertation. Institute of Mechanical Engineering, Alborg University, 1994.

[16] Z. Michalewicz. *Genetics Algorithms + Data Structures = Evolution Programs.* WNT, Warsaw, 1996.

[17] R.E. Moore. *Interval Analysis.* Prentice-Hall, New Jersey, 1966.

[18] A. Neumaier. *Interval methods for systems of equations.* Cambridge University Press, New York, 1990.

[19] J.D. Pinter. *Global Optimization in Action.* Kluwer, Dordrecht 1995.

[20] A. Pownuk. Application of interval global optimization to calculation of extreme values of mechanical quantities. *20th International Scientific Symposium of Students and Young Research Workers*, Vol. IV: *Mechanics*, pp.211–215. Zielona Góra, 1998.

[21] A. Pownuk. Optimization of mechanical structures using interval analysis. *Proc. 14th Polish Conference on Computer Methods in Mechanics (PCCMM'99)*, Rzeszów, Poland, May 26–29, 1999, pp. 301–302.

[22] H. Ratschek, J. Rokne. *Computer Method for the Range of Function.* John Wiley and Sons, New York, 1984.

[23] H. Ratschek, J. Rokne. *New Computer Methods for Global Optimization.* John Wiley and Sons, New York, 1988.

[24] P.J. Thalacker, K. Julien, P.G. Toumanoff, J.P. Daniels, G.F. Corliss, R.B. Kearfott. GlobSol Case Study: Portfolio Management (Banc One), *Rigorous Global Search Working Note 6* (`http://www.mscs.mu.edu/~globsol/papers.html`), 1998.

Internet resources

[25] `http://solon.cma.univie.ac.at/~neum/glopt.html` – A. Neumaier.

[26] `http://ubmail.ubalt.edu/~harsham/refop/Refop.htm` – H. Arsham.

[27] `http://plato.la.asu.edu/guide.html` – H.D. Mittelmann, P. Spellucci.

[28] `http://www.mscs.mu.edu/~globsol/sites.html` – Research project sponsored by Sun Microsystems.