Generalized subinterval selection criteria for interval global optimization*

Tibor Csendes

University of Szeged, Institute of Informatics, Szeged, Hungary

Abstract

The convergence properties of interval global optimization algorithms are studied which select the next subinterval to be subdivided with the largest value of the indicator $pf(f_k, X) = \frac{f_k - \underline{F}(X)}{\overline{F}(X) - \underline{F}(X)}$. This time the more general case is investigated, when the global minimum value is unknown, and thus its estimation f_k in the iteration k has an important role. A sharp necessary and sufficient condition is given on the f_k values approximating the global minimum value that ensure convergence of the optimization algorithm. The new theoretical result enables new, more efficient implementations that utilize the advantages of the pf^* based interval selection rule, even for the more general case when no reliable estimation of the global minimum value is available.

Key words: Convergence properties, interval methods, global optimization, interval selection

AMS Subject Classification: 90C30, 65K05

Short title: New subinterval selection in interval global optimization

1 Introduction

The present paper extends the results of an earlier one [5] for the more general case, when the global minimum value is not previously known.

*This work was supported by the Grants OMFB D-30/2000, OMFB E-24/2001, OTKA T 032118 and T 034350.

Consider the bound constrained global optimization problem [10, 19]

$$\min_{x \in X} f(x) \tag{1}$$

where the n-dimensional interval X is the search region, and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is the objective function. We assume that there exists at least one global minimizer point in X, that is also a stationary point. Problems that have only not stationary global minimizer points on the boundary of the search region, can be recognized by interval optimization methods, and they can be solved usually in a relatively easy way, since then the monotonicity test is usually efficient.

The algorithm considered is based on inclusion functions calculated by interval arithmetic [15]:

Definition 1 *A function $F : \mathbb{I}^n \to \mathbb{I}$ is an* inclusion function *of the objective function f if for $\forall Y \in \mathbb{I}^n$ and $\forall y \in Y$ $f(y) \in F(Y)$, where $\mathbb{I}$ stands for the set of all closed real intervals.*

In other words, $f(Y) \subseteq F(Y)$ where $f(Y)$ is the range of f over Y. The lower and upper bounds of an interval $Y \in \mathbb{I}^n$ are denoted by $\underline{Y}$ and $\overline{Y}$, respectively. The width of an interval is $w(Y) = \overline{Y} - \underline{Y}$ if $Y \in \mathbb{I}$, and $w(Y) = \max_i(\overline{Y_i} - \underline{Y_i})$ if $Y \in \mathbb{I}^n$ is an n-dimensional interval vector (also called a box). $\mathbb{I}(X)$ stands for all intervals in X. Three important types or possible properties of inclusion functions are:

Definition 2 *F is said to be an* isotone inclusion function *over X if for $\forall Y, Z \in \mathbb{I}(X)$, $Y \subseteq Z$ implies $F(Y) \subseteq F(Z)$.*

Definition 3 *We say that the inclusion function F has the* zero convergence property, *if $w(F(Z_i)) \to 0$ holds for all the $\{Z_i\}$ interval sequences for which $Z_i \subseteq X$ for all $i = 1, 2, \ldots$ and $w(Z_i) \to 0$.*

Denote the global minimum value of the function $f(x)$ on the search region X by f^*. Assume that we have an isotone inclusion function $F(X)$ for $f(x)$.

Several Branch-and-Bound (B&B) type algorithms have been suggested and studied for the solution of (1) utilizing inclusion function information on the problem [9, 11, 15]. To allow a general discussion, we study the following algorithm framework that can incorporate most of the features of the present procedures.

Algorithm

Step 1 Let L be an empty list, the leading box $A := X$, and the iteration counter $k := 1$. Set $\tilde{f} = \overline{F}(X)$.

Step 2 Subdivide A into s subsets $A_i, (i = 1, \ldots, s)$ satisfying $A = \cup A_i$ so that $\mathrm{int}(A_i) \cap \mathrm{int}(A_j) = \emptyset$ for all $i \neq j$ where int denotes the interior of a set. Evaluate the inclusion function $F(X)$ for all the new subintervals, and update the upper bound of the global minimum: $\tilde{f} := \min\{\tilde{f}, \overline{F}(A_1), \ldots, \overline{F}(A_s)\}$.

Step 3 Let $L := L \cup \{(A_i, pf(f_k, A_i))\}$ for all $i \in \{1, ..., s\}$.

Step 4 Use the accelerating devices: delete parts of the subintervals stored in L that cannot contain a global minimizer point.

Step 5 Set A to be the that subinterval from the list L which has the largest pf value, and remove the related pair from the list.

Step 6 While termination criteria do not hold let $k := k + 1$ and go to Step 2.

In Step 4, so called accelerating devices can be used that delete or shrink subintervals without discarding a global minimizer point. Such accelerating devices can be for example the cut-off test (for some implementations it is called midpoint test), the monotonicity test, the interval Newton step and the concavity test. It is important that no global minimizer point is lost in this process.

In [5] the effects of the application of

$$pf(f_k, Y) = \frac{f_k - \underline{F}(Y)}{\overline{F}(Y) - \underline{F}(Y)},$$

as an indicator that gives which interval is to be selected for subdivision were investigated. In the related algorithm that interval Y was chosen which had the maximal $pf(f_k, Y)$ value. Here f_k is the approximation of the global minimum value in the iteration k, and in this way $pf(f_k, Y)$ is a variant of the *RejectIndex*, $pf^* = \frac{f^* - \underline{F}(Y)}{\overline{F}(Y) - \underline{F}(Y)}$ [1, 2, 3, 4].

In that earlier paper [5] the author investigated the convergence properties, and it was found that the necessary and sufficient conditions for the convergence to a set of global minimizer point were that the sequence $\{f_k\}$ converges to the global minimum value f^* and there exist at most a finite number of f_k values below f^*.

Now we investigate such an algorithm that can utilize the new interval selection rule without knowing a priori the global minimum value. Its approximation, the real value f_k in the k-th iteration is between the known best lower and upper bounds of f^*:

$$\underline{f}_k = \min\{\underline{F}(Y^l), l = 1, ..., |L|\} \leq f_k < \tilde{f}_k = \overline{f}_k,$$

where $|L|$ stands for the cardinality of the elements of the list L. Here the list L is always the actual list, i.e. the one available at the iteration number k.

2 Convergence condition

To investigate the convergence properties of the introduced algorithm, we assume that the stopping conditions are either deleted, or they cannot be fulfilled, and that the subdivision direction selection is balanced [7].

Theorem 1 *Assume that the inclusion function of the objective function is isotone and it has the zero convergence property. Consider the interval branch-and-bound optimization algorithm that uses the cut-off test, the monotonicity test, the interval Newton step and the concavity test as accelerating devices, and that selects as next leading interval Z from the working list which has the maximal $pf(f_k, Z)$ value.*

1. The algorithm converges exclusively to global minimizer points if

$$\underline{f}_k \leq f_k < \delta(\overline{f}_k - \underline{f}_k) + \underline{f}_k$$

holds for each iteration number k, where $0 < \delta < 1$.

2. The above condition is sharp in the sense that $\delta = 1$ allows convergence to not optimal points.

Proof. 1. Notice first that the maximal $pf(f_k, Y)$ values are always nonnegative, since f_k is not less than the minimal lower bound of F. Due to $f_k < \tilde{f}$, the numerator of pf is less than $\tilde{f} - \min\{\underline{F}(Y^l), l = 1, ..., |L|\}$. $\underline{f}_k$ is conservative, i.e. it is monotonically nondecreasing (based on the isotone inclusion functions). A similar property is ensured for $\overline{f}_k$ by the isotonicity of $F(X)$, and by the updating of $\tilde{f}$. Thus $\underline{f}_k$ is monotonically nondecreasing, and $\overline{f}_k$ is monotonically nonincreasing. The sequence $\underline{f}_k$ converge to a value in $[\underline{F}(X), f^*]$ depending on the actual problem instance.

Consider now an arbitrary point $x' \in X$ in such a way that $f(x') > f^*$, and that there is a subsequence $\{Y_{k_l}\}$ of the leading boxes that converges to x'. For this point x' the sequence of lower bounds $\underline{F}(Y_{k_l})$ converges to $f(x')$ due to the zero convergence property, and obviously the sequence of upper bounds $\tilde{f}_k = \overline{f}_k$ on the minimum value converges to a value not greater than $f(x')$. Now the f_k values must be below $f(x')$ from a certain iteration index, since they fulfill the condition

$$\underline{f}_k \leq f_k < \delta(\overline{f}_k - \underline{f}_k) + \underline{f}_k$$

with a $0 < \delta < 1$, and the difference between $\overline{f}_k$ and f_k is larger than $(1-\delta)(\overline{f}_k - \underline{f}_k)$ which is at least $(1-\delta)(f(x') - f^*) > 0$ (since an $\tilde{f}$ below $f(x')$ would inhibit a convergence to x'). Then the respective pf values are negative from an index.

If there are more such points $x_1, x_2, ... \in X$ for which $f(x_i) > f^*$, and for each of them there exist a subsequence of the leading boxes that converges to the given point, then the above reasoning holds for each of them. In other words, also in this case from a certain index all pf values are negative.

On the other hand, there is always at least one global minimizer point, a stationary point in one of the subintervals in the list L. The respective subinterval cannot be deleted by an accelerating step, and thus its $pf(f_k, Y)$ value is nonnegative. But this contradicts that a subinterval with a negative pf value is selected, i.e. no subsequence of the generated intervals can converge to a not optimal point of the search region.

2. The second statement is a consequence of Theorem 3 and Corollary 2 in [5] that require the convergence of $\tilde{f}$ to the global minimum value to have the interval B&B optimization algorithm with the $pf(f_k, Y)$ based interval selection rule to converge exclusively to global minimizer points. □

Notice that if we set $f_k = \min\{\underline{F}(Y^l), l = 1, ..., |L|\}$ then we have actually the Moore-Skelboe algorithm, since then always that subinterval is selected which has the minimal lower bound, since for these intervals the pf value will be zero while for each other interval it will be negative. In this sense the present theorem is a generalization of the convergence assertion on the Moore-Skelboe algorithm. Theorem 1 remains true when some or all of the accelerating devices are not used in the algorithm.

The required isotonicity is easy to achieve also for non-isotone inclusion functions by intersecting the new inclusion function value $F(Y^i)$ with that of the direct ancestor interval Y (for which $Y = \cup_i Y^i$).

3 Implementation and numerical test results

The goal of the present computational test was to demonstrate the effect of the new interval selection rule (utilizing the approximate optimum value known with 4 digit precision obtained by a traditional optimization algorithm as the starting $\tilde{f}$ value) compared to the old algorithm variants studied in [6]. This precision is usually not difficult to achieve, and it costs about 1,000 to 10,000 additional function evaluations (not counted in the corresponding table). To have a reliable method, the approximate value must be validated by an interval based evaluation. The algorithm can improve the set starting $\tilde{f}$ value.

In the paper of [6] the numerical efficiency of several algorithm variants was investigated on 40 standard test functions. Three procedures were compared:

- the Moore-Skelboe algorithm that selects that subinterval for subdivision which has the lowest lower bound on the objective function,
- the one that selects the subinterval with the maximal pf^* value, and
- a procedure that selects the subinterval with the maximal $pf(f_k, X)$ value, where $f_k = (\underline{f}_k + \overline{f}_k)/2$.

The studied algorithms utilized only the cutoff test. For the investigated procedures no derivative information was necessary. We used the traditional bisection and the subdivision was made along the coordinate direction with the longest edge. The algorithms were stopped when the diameter of a candidate interval was smaller than 0.01, or if the length of the working list reached 20,000. This memory limitation is far from the physical one, still above this level, a larger and larger part of the computation must be spent on administration in contrast to function evaluations.

The conclusion of the paper [6] was that when the global minimum value is known it is the best to use it in the form of the pf^* based interval selection criterion. If it is not available, then the $f_k = (\underline{f}_k + \overline{f}_k)/2$ estimation can be used, and the resulting algorithm will be more efficient for hard to solve problems than the Moore-Skelboe algorithm. The results based on the known global minimum value were so much better than the others, that it seems to be reasonable to use a good approximation of the minimum value (e.g. obtained by a real arithmetic based traditional optimization).

The present numerical tests were also carried out on a Pentium-IV computer (1,4 Ghz, 1 Gbyte RAM) under the Linux operating system. The

Table 1: The maximal list length needed for the solution of the test problems.

Problem		MLL						
name	dim.	$\underline{F}$	$(\overline{f}_k + \underline{f}_k)/2$		4 digit approx.		pf^*	
				%		%		%
H3	3	20,000	20,000	100	3,256	16	2,383	12
H6	6	20,000	20,000	100	20,000	100	20,000	100
GP	2	20,000	20,000	100	2,106	11	2,145	11
SHCB	2	20,000	17,643	88	762	4	762	4
L3	2	20,000	20,000	100	72	0	72	0
L5	2	20,000	20,000	100	37	0	37	0
Sch27	3	5,706	16	0	16	0	5,706	100
EX2	5	20,000	20,000	100	20,000	100	20,000	100

programs were coded in C++. The inclusion functions were implemented via the PROFIL/BIAS routines [12], and the basis algorithm was that of the C++ Toolbox for Verified Computing [8]. The standard time unit (the CPU time required to evaluate the Shekel 5 test function 1000 times at $(4.0, 4.0, 4.0, 4.0)^T$) was 0.00076 seconds.

In contrast to our earlier paper discussing an extensive numerical study, our present computational experiments used only those test problems, that were the most difficult to solve among those in [6]. Thus now the problems Hartman-3 (H3), Hartman-6 (H6), Goldstein-Price (GP), Six-Hump-Camel-Back (SHCB), Levy-3 (L3), Levy-5 (L5), Schwefel-2.7 (Sch27), and EX2 from [7]. The search regions were the same as in other numerical tests [7, 9, 16, 19]. The numerical results are demonstrated in Tables 1 to 3.

In our test the most important indicator is the required number of list elements for the solution of the given problems. In instances when the respective value is 20,000, the related method was unable to solve the given problem, thus all further efficiency indicators are incomparable for these cases. According to the maximal list length required (MLL), it is definitely worth to use an approximation of the global minimum value, since with this overhead, our new algorithm was able to solve all those problems, which were otherwise solved only by the pf^* based method (that needs the a priori known global minimum value). The MLL values were close to those obtained by the pf^* method, in one instance (Schwefel 2.7 problem) it was even better.

According to Table 2, the CPU times needed for the solution proved to be a success story for the new method (denoted by 4 digit approx.): it could

Table 2: The CPU time in seconds required for the solution.

Problem		CPUt						
name	dim.	$\underline{F}$	$(\overline{f}_k + \underline{f}_k)/2$		4 digit approx.		pf^*	
				%		%		%
H3	3	347.64	431.98	124	8.46	2	5.59	2
H6	6	444.75	439.99	99	375.53	84	368.55	83
GP	2	474.79	1,760.60	371	3.09	1	3.48	1
SHCB	2	362.53	298.12	82	0.45	0	0.54	0
L3	2	387.02	443.24	115	0.07	0	0.09	0
L5	2	381.78	319.82	84	0.03	0	0.05	0
Sch27	3	114.40	0.06	0	0.04	0	115.27	101
EX2	5	358.43	354.16	99	311.11	87	328.91	92

provide the minimal requirements of the follow up two techniques, that of the $(\overline{f}_k + \underline{f}_k)/2$ based, and the pf^* based methods. It is remarkable, that all solved problems were handled within 10 seconds.

The number of function evaluations and the number of iterations (NFE) go understandably in parallel, thus the later is not demonstrated here. Although the new method (4 digit approx.) is the best according to NFE in all comparable cases, still, remember, that the new method involves an additional amount of function evaluations for the approximate optimization to obtain an estimated global minimum value. Notice that the solution time difference between the best technique not using f^* and the new method allows the preliminary approximate optimization, and provides a substantial saving in the whole test set.

Summarizing the numerical experiences, we can conclude that according to the tests made it is definitely worth to run a traditional, real arithmetic based optimization algorithm to obtain an approximate minimum value, since it can well be utilized with the $pf(f_k, X)$ indicator, and in the corresponding subinterval selection rule. As the efficiency indicators show, it seems that the approximate minimum value can be well used in the interval selection rule. We have repeated our computational tests with other precision values, and we have found that the savings were similar in a relatively large range of precision, with 100% to 0.000001% relative error in the estimation of the global minimum value. The large complexity savings open the way to heuristic reliable procedures, which use iteratively estimated optimum values together with reliable interval optimization techniques.

Table 3: The number of objective function evaluations needed for the solution.

Problem		NFE						
name	dim.	$\underline{F}$	$(\overline{f}_k + \underline{f}_k)/2$		4 digit approx.		pf^*	
				%		%		%
H3	3	66,817	120,055	180	6,519	10	7,159	11
H6	6	64,105	68,074	106	40,135	63	60,202	94
GP	2	70,663	636,550	901	4,211	6	6,433	9
SHCB	2	78,883	129,211	164	1,553	2	2,329	3
L3	2	70,774	118,153	167	189	0	283	0
L5	2	69,223	87,520	126	81	0	121	0
Sch27	3	60,535	88	0	59	0	60,535	100
EX2	5	62,407	78,151	125	42,547	68	63,823	102

References

[1] L.G. Casado and I. García, New Load Balancing Criterion for Parallel Interval Global Optimization Algorithm, in Proceedings of the 16th IASTED Int. Conf., Garmisch-Partenkirchen, Germany, 1998, 321–323.

[2] L.G. Casado, I. García, and T. Csendes, A new multisection technique in interval methods for global optimization, Computing 65(2000), 263-269.

[3] L.G. Casado, I. García, and T. Csendes, A Heuristic Rejection Criterion in Interval Global Optimization, BIT 41(2001), 683-692.

[4] L.G. Casado, I. García, and T. Csendes, Heuristic Rejection in Interval Global Optimization, JOTA 118(2003).

[5] T. Csendes, Convergence properties of interval global optimization algorithms with a new class of interval selection criteria, J. Global Optimization 19(2001), 307-327.

[6] T. Csendes, Numerical experiences with a new generalized subinterval selection criterion for interval global optimization. Reliable Computing, Computing, 9(2003) 109-125.

[7] T. Csendes and D. Ratz, Subdivision direction selection in interval methods for global optimization, SIAM J. Numerical Analysis 34(1997), 922-938.

[8] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz, C++ Toolbox for Verified Computing, Springer, Berlin, 1995.

[9] E. Hansen, Global optimization using interval analysis, Marcel Decker, New York, 1992.

[10] R. Horst and P.M. Pardalos (eds.), Handbook of Global Optimization, Kluwer, Dordrecht, 1995.

[11] R. B. Kearfott, Rigorous global search: continuous problems, Kluwer, Dordrecht, 1996.

[12] O. Knüppel, BIAS — Basic Interval Arithmetic Subroutines, Technical Report 93.3, University of Hamburg, 1993.

[13] M.Cs. Markót, T. Csendes, and A.E. Csallner, Multisection in Interval Branch-and-Bound Methods for Global Optimization II. Numerical Tests, J. Global Optimization 16(2000), 219-228.

[14] R.E. Moore and H. Ratschek, Inclusion functions and global optimization II, Mathematical Programming 41(1988), 341–356.

[15] H. Ratschek and J. Rokne, New Computer Methods for Global Optimization, Ellis Horwood, Chichester, 1988.

[16] D. Ratz and T. Csendes, On the Selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization, J. Global Optimization 7(1995), 183–207.

[17] S. Skelboe, Computation of rational functions, BIT 14(1974), 87–95.

[18] R. Stateva and S. Tsvetkov, A Diverse Approach for the Solution of the Isothermal Multiphase Flash Problem, Application to Vapor-Liquid-Liquid Systems, The Canadian J. of Chemical Engineering 72(1994), 722–734.

[19] A. Törn and A. Žilinskas, Global Optimization, Springer, Berlin, 1987.