To appear in series *Nonconvex Global Optimization and Its Applications*
Kluwer Academic Publishers

Termination criteria in the Moore-Skelboe Algorithm for Global Optimization by Interval Arithmetic

M.H. van Emden
Department of Computer Science
University of Victoria
Victoria
vanemden@cs.uvic.ca

B. Moa
Department of Computer Science
University of Victoria
Victoria,
bmoa@cs.uvic.ca.

Abstract

We investigate unconstrained optimization with an objective function that has an unknown and possibly large number of local minima. By varying the selection and termination criteria, we obtain several variants of the Moore-Skelboe algorithm for distinct tasks in nonconvex global optimization. All of these terminate after having found the best answer that is possible, given the precision of the underlying hardware and given the expression for the objective function. The first algorithm finds the best lower bound for the global minimum. This is then extended to a version that adds an upper bound.

Often not only the global minimum is required, but also possibly existing points that achieve near-optimality, yet are far from the points at which the global minimum occurs. In response to this requirement we define the δ*-minimizer*, the set of points at which the objective function is within δ of the global minimum.

We then present algorithms that return a set of boxes. In one of these, the union of the boxes in this set contains a δ-minimizer. If this union is small, then we know that there is a well-defined global minimum. In the other version, the union of the boxes returned is contained in a δ-minimizer. If this union is large, then we know that there is a wide choice of parameters that yield near-optimal objective function values.

Keywords: nonconvex unconstrained global optimization, Moore-Skelboe algorithm, minimizer, sensitivity analysis

1 Introduction

There are many features that contribute to the degree of difficulty of an optimization problem. As the wide applicability and the great flexibility of the optimization paradigm make it tempting to formulate models with ever increasing numbers of variables, all non-statistical global optimization methods are severely limited in the number of variables they can handle. This is an acutely felt limitation.

With the number of variables given, an important determinant of degree of difficulty is whether the objective function has a single local minimum. In addition to this favorable property, it is a powerful help if the matrix of second derivatives exists and is well-conditioned near the local minimum. At the other extreme, the objective function may have an unknown and possibly large number of local minima. It may be that neither second derivatives nor even first derivatives are available. In this paper we consider optimization problems of this latter type. The only assumption we make of the objective function is that it is bounded from below and that it can be computed by an expression that can be evaluated in interval arithmetic.

This class of optimization problems is solved, in sufficiently small instances, by the Moore-Skelboe algorithm [7, 8, 3, 11]. Many variants exist, mostly in the selection and termination criteria. In the literature [7, 8, 1, 9], these variants are compared on heuristic grounds. We show that in this respect one can move from the heuristic to the exact. Our first step in this direction is to clarify what optimization problem is to be solved.

One possible goal is to determine *what* the global minimum is. We call this the *fathoming problem*. Another possible goal is to determine *where* the global minimum occurs. We call this the *localization problem*.

These distinct goals determine different termination criteria. Within the fathoming problem, we first present an algorithm that finds the best lower bound for the global minimum. Next comes an algorithm for finding the best interval for the global minimum. To address the localization problem, we present an algorithm that yields a set of boxes containing the δ-minimizer. Another localization algorithm yields a set of boxes contained in a δ-minimizer.

2 Preliminaries

Definition 1 *An* optimization setting *consists of the following.*
(1) An objective function f*, which is a function of type* $\mathcal{R}^n \rightarrow \mathcal{R}$*.*
(2) A domain $\mathcal{D}$*, which is a non-empty subset of* $\mathcal{R}^n$*.*
(3) A set of conditions *of the form* $g_i(x_1, \ldots, x_n) \leq 0$*, for* $i \in \{1, \ldots, m\}$*. Here* $g_1, , \ldots, g_m$ *are functions of type* $\mathcal{R}^n \rightarrow \mathcal{R}$*.*

Logically, the domain (2) and the conditions (3) are mutually redundant. Yet it is convenient to have both. The domain is typically simply defined, for example as a Cartesian product of intervals. The conditions may not be trivial to solve. If $m = 0$ and $\mathcal{D}$ is a cartesian product of intervals, then we speak of an unconstrained optimization problem.

Given an optimization setting, the following additional definitions suggest themselves.

Definition 2 *(1) The* feasible set*, which is the intersection of the domain with the subset of* $\mathcal{R}^n$ *that satisfies the conditions.*

(2) The global minimum μ*, which is defined as the greatest lower bound of* f *restricted to a non-empty feasible set.*
(3) The δ-minimizer*, which is defined as the intersection of the feasible set with* $\{\langle x_1, \ldots, x_n\rangle \mid f(x_1, \ldots, x_n) - \mu \leq \delta\}$*, for some* $\delta \geq 0$*. The* minimizer *without qualification is the 0-minimizer.*

2.1 Intervals

We suppose a finite set of floating-point numbers, for example as specified in IEEE standard 754. We consider the finite floating-point numbers as reals; therefore they can serve as bounds for closed connected sets of reals. We consider as intervals pairs $[a, b]$ of finite floating-point numbers such that $a \leq b$. These denote the closed connected sets of reals bounded by a and b. We extend this notation to the infinite floating-point numbers by letting $[-\infty, b]$ and $[a, +\infty]$ denote the obviously suggested unbounded closed connected sets of reals.

We call a the *left bound* and b the *right bound*, writing $a = lb([a, b])$ and $b = rb([a, b])$. The *width* of $[a, b]$ is $w([a, b]) = b - a$.

An interval of the form $[a, a]$ is called a *point interval*. We denote the empty interval by $\emptyset$. As we consider the finite floating-point numbers as reals, the distinct bit patterns $+0$ and -0 denote the same real. Thus $[-0, -0]$, $[-0, +0]$, $[+0, -0]$, and $[+0, +0]$ are all equal and are equal to the point interval $[0, 0]$.

The fact that there are a finite number of floating-point numbers has important consequences. There is a greatest finite floating-point number M. Adjacent floating-point number have a positive distance between them. We have *atomic* intervals, which are defined as intervals $[a, b]$ with $a = b$ or $a < b$ and a and b adjacent floating-point numbers. Atomic intervals typically have small width. However, $[-\infty, -M]$ and $[M, \infty]$, where M is the greatest finite floating point number, are also atomic.

The *split* operation is defined on non-empty, non-atomic intervals $[a, b]$ and yields two intervals $[a, m]$ and $[m, b]$, where m is a floating-point number such that $a < m < b$. Thus, the split operation, if defined, results in narrower intervals.

Our algorithms typically continue splitting as long as possible. As a result, we can claim that they result in the best that can be obtained, given the limitations of the underlying arithmetic.

2.2 The objective function

The Moore-Skelboe algorithm depends on lower bounds for the objective function. It obtains these by interval arithmetic [1]. As a result, it is essential that the objective function f be given by an expression that can be evaluated in interval arithmetic. We assume that this expression is in terms of rational operations in which the standard functions (exponential, logarithm, trigonometric) may also occur.

This requirement rules out, for example, objective functions that are given as sets of observational data. For such data to become usable for the construction of an objective function, approximation or interpolation techniques can often be applied to obtain an objective function of the required form.

[1]Lower bounds can also be obtained if one has a Lipschitz condition on f. This method is used by J. Pintér [10].

Definition 3 *We assume an expression E is given that contains variables $x_1, \ldots, x_n$. E computes f in the sense that $f(a_1, \ldots, a_n)$ has as value E with $a_1, \ldots, a_n$ are substituted in E for $x_1, \ldots, x_n$. We assume that E can be evaluated in interval arithmetic.*

The same symbol f is used to denote the following three functions, which are distinguished by the types of their arguments:
(1) The objective function of type $\mathcal{R}^n \to \mathcal{R}$.
(2) The function that maps intervals $X_1, \ldots, X_n$ to the result of evaluating in interval arithmetic E with $X_1, \ldots, X_n$ substituted for $x_1, \ldots, x_n$.
(3) The function that maps a box $B = X_1 \times \cdots \times X_n$ to $f(X_1, \ldots, X_n)$, as defined above.

3 Interval arithmetic for global optimization

The presence of an unknown and possibly large number of local minima may seem to preclude the possibility of finding a lower bound for the global minimum. After all, even if one has identified a thousand local minima, how does one know that there is not yet another one with an objective-function value lower than any found so far? Moreover, it is possible that a spike-shaped global minimum exists that fits entirely between two consecutive floating-point numbers.

The answer is that interval arithmetic has the property of producing intervals that *contain all possible values*. More precisely, we have the following theorem.

Theorem 1 *(The* Fundamental Law of Interval Arithmetic*).*
Let f be a function of type $\mathcal{R}^n \to \mathcal{R}$ and let $X_1, \ldots, X_n$ be intervals. We have

$$\{f(x_1, \ldots, x_n) \mid x_1 \in X_1, \ldots, x_n \in X_n\} \subset f(X_1, \ldots, X_n).$$

For the different meanings of the two occurrences of f, see Definition 3.

The fundamental theorem guarantees that the lower bound of the interval for f computed by interval arithmetic is a lower bound for the global minimum in $X_1 \times \cdots \times X_n$.

However, neither the fundamental theorem nor anything else ensures that this is *useful*: it may be that the lower bound is far away from the global minimum. Usually, the narrower the intervals in $X_1 \times \cdots \times X_n$, the closer the left bound of the interval $f(X_1 \times \cdots \times X_n)$ is to the greatest lower bound of $\{f(x_1, \ldots, x_n) \mid x_1 \in X_1, \ldots, x_n \in X_n\}$. This is far from guaranteed; it is only typical. What we do know is that splitting does not make the interval for the global minimum worse. This is because of the monotonicity of canonical set extensions of functions in general. In the case of a function $f : \mathcal{R}^n \to \mathcal{R}$ extended to an f that maps n intervals to an interval as defined in Definition 3, monotonicity is defined as follows.

Definition 4 *Let E be an expression that contains variables $x_1, \ldots, x_n$, and let f be the interval function associated with it. This function is said to be* monotonic *iff for all intervals $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$, such that $X_1 \subset Y_1, \ldots, X_n \subset Y_n$, we have $f(X_1, \ldots, X_n) \subset f(Y_1, \ldots, Y_n)$*

This suggests subdividing $X_1 \times \cdots \times X_n$ into smaller boxes and evaluating f over each one of these. The result of subdividing is a set of boxes that covers the set in question.

Definition 5 *Let $N > 0$ be an integer. If $\langle S_0, \ldots, S_{N-1} \rangle$ is a sequence of non-empty subsets of a set S, then the sequence is called a* cover in S. *If, in addition, the union of $S_0, \ldots, S_{N-1}$ is S, then the sequence $\langle S_0, \ldots, S_{N-1} \rangle$ is a* cover for S.

let the cover be $\{B_0\}$
while $(w(f(B_0)) > \epsilon)$ {
    // $\mu \in f(B_0)$
    remove B_0 from the cover
    split B_0 and insert the results into the cover in non-decreasing order of
    $lb(f(B_i))$, for $i = 0, \ldots, N-1$
}
// $\mu \in f(B_0)$ and $w(f(B_0)) \leq \epsilon$
output $f(B_0)$

Figure 1: The algorithm MS_0. It is intended to compute an interval for the global minimum with width less than or equal to a positive ϵ.

Assume that we have a cover $\langle B_0, B_1, \ldots, B_{N-1}\rangle$ for $X = X_1 \times \cdots \times X_n$ and that the cover is ordered by non-decreasing lower bounds of $f(B_i)$, for $i \in \{0, \ldots, N-1\}$. Initially, we can take $B_0 = X$ and $N = 1$. Such a cover contains the minimizer. We will consider algorithms that change a given cover containing the minimizer to one that has a smaller union and still contains the minimizer.

Theorem 2 *Let us consider an unconstrained global optimization problem. Let $\langle B_0, B_1, \ldots, B_{N-1}\rangle$ be a cover containing the global minimizer that is ordered according to nondecreasing order of the left bounds of $f(B_i)$ for $i = 0, 1, 2, \ldots, N-1$. Let U be the smallest among the right bounds of the intervals $\langle f(B_0), f(B_1), \ldots, f(B_{N-1})\rangle$. The interval $[lb(f(B_0)), U]$ contains the global minimum μ.*

Proof. Consider $lb(f(B_0))$. None of the other boxes in the cover has a smaller left bound. As the global minimum has to be achieved in at least one of the boxes (because the number of sets in the cover is finite), the left bound of $f(B_0)$ is a lower bound for μ. Let us now consider U. Suppose it is the right bound of $f(B_i)$ with $i \in \{0, \ldots, N-1\}$. As B_i is nonempty, $f(B_i)$ contains at least one value y of f. We have $U = rb(f(B_i)) \geq y \geq \mu$. We conclude that $[lb(f(B_0)), U]$ contains the global minimum μ.

4 The Moore-Skelboe algorithm

The considerations in the previous section suggest the possibility of solving the fathoming and the localization problem by constructing a suitable cover.

4.1 The fathoming problem for unconstrained global optimization

The original Moore-Skelboe algorithm can be regarded as addressing the fathoming problem. Essentially, the algorithm is as shown in Figure 1. Here it is desired to find an interval for μ of width at most ϵ, for some real $\epsilon > 0$.

The algorithm in Figure 1 has some positive features. In the first place, it may find a sufficiently narrow interval for μ. Second, it does this by subdividing X in an adaptive way, as explained below.

Let us consider the operation "split". By its definition, the results are nonempty, are both proper subsets, and have a union that is equal to the box that was split. The number of boxes in the cover created by the algorithm typically becomes so large that the cover cannot be stored. So one should be careful which box to split[2]. It is desirable to split a box most likely to contain the minimizer. The heuristic chosen by the algorithm in Figure 1 is to split the box B_0 for which $f(B_0)$ has the lowest lower bound. The subdivision resulting from the splits in this algorithm is *adaptive*: boxes far away from the global minimum tend not to be split. This goes some way towards avoiding covers with more sets than can be stored.

However, algorithm MS_0 needs improvement. For example, what happens if one chooses a too small positive ϵ? If the algorithm does not abort because the number of sets in the cover has become too large to be stored, B_0 will become atomic[3]. In that case the effect of split in the algorithm in Figure 1 is undefined.

To prevent this, we need to include a test whether the box B_0 of the cover is atomic, as is done in Figure 2. The interval returned contains μ. The width of the returned interval is either at most ϵ, and then we get what we asked for. If the returned interval is wider than ϵ, then we know that it has the best lower bound that is possible with the given arithmetic and expression for the objective function. That is, if we ask too much of the algorithm in the form of an ϵ that is too small, then we get as a consolation prize a very valuable result. Hence we call it the "Consolation Prize Algorithm". Its distinctive feature is stated in theorem 3.

Theorem 3 *The Consolation Prize algorithm in Figure 2 terminates and, in case of $w(f(B_0)) > \epsilon$, it returns the best lower bound.*

The termination of the Consolation Prize Algorithm is based upon the fact that the number of floating-point numbers is finite; hence the total number of the boxes that can be defined is finite. Every split changes a non-atomic box into two strictly smaller boxes. A non-termination loop would therefore generate an infinite sequence of different boxes. Termination of the algorithm follows. It is possible that $\mu = lb(f(B_0))$. $f(B_0)$ is therefore the only interval known to contain μ. The only way to improve $lb(f(B_0))$ as lower bound of μ is to split B_0. When B_0 is atomic, this lower bound cannot be further improved.

However, when we ask too much of the algorithm in the form of an ϵ that is too small, it would also be reasonable to get the narrowest possible *interval* for μ. This is not necessarily the case with the algorithm in Figure 2: the box B_0 with the smallest left bound for $f(B_0)$ may not have the smallest right bound. Thus we see that the algorithm in Figure 2, though ostensibly its purpose is to find a B with $w(f(B)) \leq \epsilon$, it does not try very hard. If it does not achieve its goal by improving the lower bound, then it should continue with improving the upper bound. Hence we call it the "interval-valued fathoming algorithm". It is shown in Figure 3. Its distinctive feature is the following.

Theorem 4 *The Algorithm in Figure 3 terminates and, for sufficiently small ϵ, outputs the best interval for μ.*

[2]The box to be split is a Cartesian product of n intervals. So we not only get to choose which box to split, but often also which projection to split.

[3]A box is atomic if all of its projections are atomic intervals.

let the cover be $\{B_0\}$
while $(w(f(B_0)) > \epsilon$ and $\neg$*atomic*$(B_0))$ {
    // $\mu \in f(B_0)$
    remove B_0 from the cover
    split B_0 and insert the results into the cover in non-decreasing order of
     lb$(f(B_i))$, for $i = 0, \ldots, N-1$
}
// $\mu \in f(B_0)$ and $(w(f(B_0)) \leq \epsilon$ or *atomic*$(B_0))$
output *lb*$(f(B_0))$;

Figure 2: The algorithm MS_1 ("Consolation Prize Algorithm"). The function *atomic* specifies whether its box argument is atomic.

let the cover be $\{B_0\}$
while $(w(f(B_0)) > \epsilon$ and $\neg$*atomic*$(B_0))$ {
    // $\mu \in f(B_0)$
    remove B_0 from the cover
    split B_0 and insert the results into the cover in non-decreasing order of
     lb$(f(B_i))$, for $i = 0, \ldots, N-1$
}
// $\mu \in f(B_0)$ and $(w(f(B_0)) \leq \epsilon$ or *atomic*$(B_0))$
if $(w(f(B_0)) \leq \epsilon)$ output $f(B_0)$; exit;

// *atomic*(B_0)
let L equal *lb*$(f(B_0))$
order the cover by non-decreasing *rb*$(f(B_i))$, for $i = 0, \ldots, N-1$
let U equal *rb*$(f(B_0))$
while $((U - L) > \epsilon$ and $\exists i$ such that $\neg$*atomic*$(B_i))$ {
    // $L \leq \mu \leq U$
    remove from the cover a non-atomic B_j with lowest *rb*$(f(B_j))$
    split B_j and insert the results of splitting into the cover
    maintaining the cover's order of non-decreasing *lb*$(f(B_i))$,
     for $i = 0, \ldots, N-1$
    $U =$ *rb*$(f(B_0))$ // update the upper bound of μ
    remove from the cover all boxes B such that *lb*$(f(B)) > U$ // Ichida-Fujii
}
// $(U - L) \leq \epsilon$ or all the boxes left in the cover are atomic
output $[L, U]$

Figure 3: The algorithm MS_2("interval-valued fathoming algorithm"). The line commented "Ichida-Fujii" marks the application of branch-and-bound added to the Moore-Skelboe algorithm in [6].

let the cover be $\{B_0\}$
let U equal $rb(f(B_0))$
while ($\exists j$ in the cover such that $\neg$*atomic*(B_j)){
    // the union of the boxes in the cover is an outer approximation to the δ-minimizer
    choose a nonatomic B_i with least upper bound $rb(f(B_i))$
    remove B_i from the cover
    split B_i and insert the results into the cover in non-decreasing order of
    $lb(f(B_k))$, for $k = 0, \ldots, N-1$
    update U
    remove all boxes B from the cover with $lb(f(B)) > (U + \delta)$
    }
}
// the union of the boxes in the cover is an outer approximation to the δ-minimizer
output the boxes in the cover

Figure 4: The algorithm MS_3. After termination, the best outer approximation to the δ-minimizer is the union of $\{B_0, \ldots, B_i\}$ where i is the greatest i such that $lb(f(B_i)) \leq (U + \delta)$.

By choosing ϵ sufficiently close to zero, one forces all boxes to become atomic or to be removed from the cover. As a result, we get the best lower and upper bounds for μ that are possible with the given expression for f and the given precision of the arithmetic.

The main limitation of the algorithms of this type is the large number of sets in the cover. Removing the sets of the cover whose lower bound exceeds U is an application of the branch-and-bound principle. This was added to the Moore-Skelboe algorithm by Ichida and Fujii [6].

One way to speed up these algorithms is to use the value of the objective function somewhere inside the boxes of the cover instead of its right bound. These are also upper bounds for the global minimum and are less than the upper bound obtained by interval arithmetic.

The above applies in the case of unconstrained optimization. However, in the presence of constraints, one has to prove the existence of a feasible point inside the box B_i.

4.2 The localization problem for unconstrained global optimization

The localization problem is to gain information about the δ-minimizer. This can take two forms: an *outer* approximation or an *inner* approximation. An outer approximation is a set of boxes whose union contains the δ-minimizer. This can always be achieved, though the union may be so large as not to be useful. Algorithm MS_3 in Figure 4 makes this union as small as possible. The distinctive property of this algorithm is the following.

Theorem 5 *Algorithm MS_3 terminates and gives the best outer approximation to the δ-minimizer.*

Proof. Suppose x is in the δ-minimizer. Then $f(x) \leq \mu + \delta$ and hence $f(x) \leq U + \delta$. This implies that there is a box in the cover containing x because only boxes B' with $lb(f(B')) > U + \delta$ have been removed from the cover.

Figure 5 illustrates the outer approximation of δ-minimizer. Note that boxes are ordered using the lower bound of f on each box. Let U be the upper bound of the box in the cover with least upper

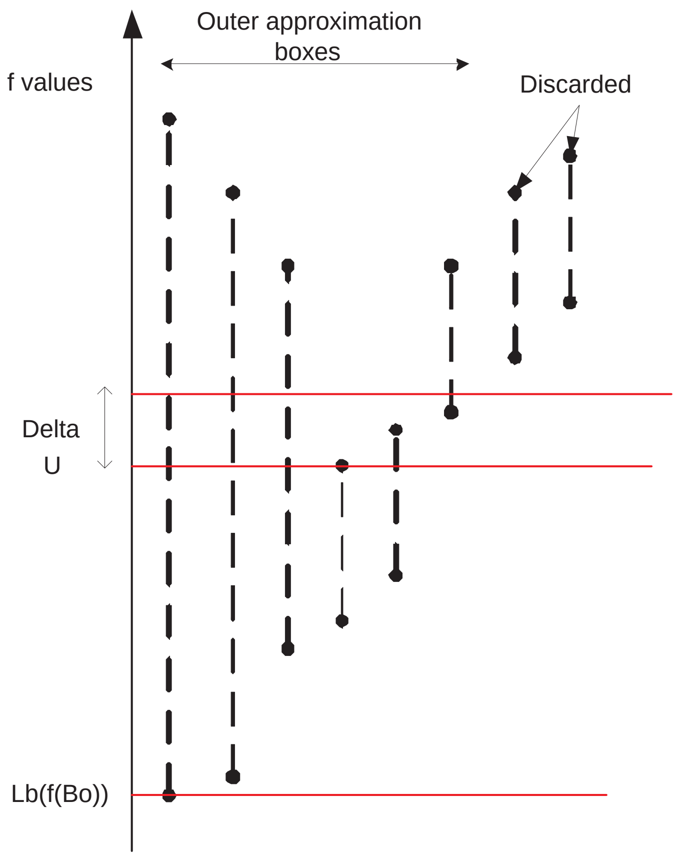

Figure 5: A terminal situation of the algorithm MS_3 for outer approximation of the δ-minimizer.

let the cover $\{B_0\}$
while ($\exists i$ such that $(U+\delta) \in f(B_i)$ and $\neg$*atomic*(B_i)) {
    // the union of the boxes B in the cover with *rb*$(f(B)) \leq (U+\delta)$
    //is an inner approximation to the $(U - lb(f(B_0)) + \delta)$-minimizer.
    remove B_i from the cover
    split B_i and insert the results into the cover in non-decreasing order of
    rb$(f(B_i))$, for $i = 0, \ldots, N-1$
    $U = rb(f(B_0))$
    remove all boxes from the cover with lower bound greater than $U + \delta$
}
// $\forall i$ *atomic*(B_i) or $(U+\delta) \notin f(B_i)$
remove every box B_j from the cover with $(U+\delta) \in f(B_j)$
output all boxes in the cover

Figure 6: The algorithm MS_4. After termination, the union of boxes with upper bounds below $U+\delta$ is the best inner approximation to the $(U - lb(f(B_0)) + \delta)$-minimizer.

bound. For an inner approximation, we let $\epsilon = U - lb(f(B_0))$, which is the width of the best interval for μ. Let B_i be a box such that $U = rb(f(B_i))$. Such a B_i will now be contained in the $(\epsilon+\delta)$-minimizer for any positive δ. In fact, all boxes of the cover that have upper bounds less than $U+\delta$ have a union that is contained in the $(\epsilon+\delta)$-minimizer. This inner approximation can be improved (that is, made larger) by splitting certain boxes. This improvement is carried out by the algorithm in Figure 6. Its distinctive characteristic is the following.

Theorem 6 *Algorithm MS_4 terminates and gives the best inner approximation to the $(\epsilon+\delta)$-minimizer.*

Proof. Assume $x \in B$ for one of the boxes of the cover. Then $f(x) \in f(B)$, so that $f(x) \leq rb(f(B))$. For all boxes B' in the cover $rb(f(B')) \leq U+\delta$. As $U \geq \mu+\epsilon$, we have $f(x) \leq \mu+\delta+\epsilon$. So it is in the $(\epsilon+\delta)$-minimizer.

Figure 7 illustrates the inner approximation of $(\epsilon+\delta)$-minimizer. In the case of inner approximation, note that boxes are ordered using the upper bound of f on each box.

5 Suggestions for further work

To be able to concentrate on the main principle, we have restricted ourselves to the framework of interval arithmetic. This has the advantage of simplicity in exposition, but it is also more restricted and less effective than the more advanced technique of interval *constraints* [4, 5]. With interval constraints, the algorithms in this paper can be extended to constrained nonconvex global optimization. Whether constrained or not, interval constraints allow a lower bound y to be obtained by transforming the objective function to a constraint system and showing the inconsistency of adding $f(x) \leq y$, as was first shown in [2]. Such lower bounds are stricter than the ones obtained by the Moore-Skelboe algorithm with the same level of splitting.

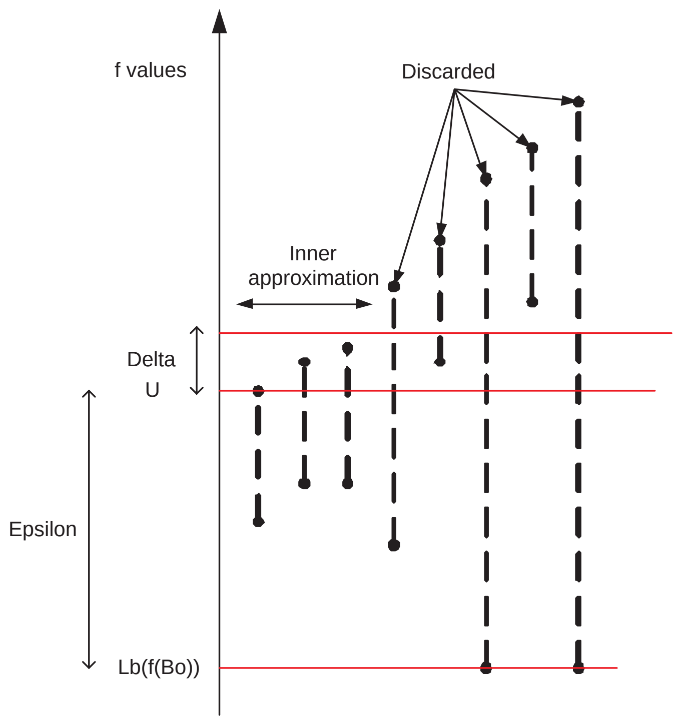

Figure 7: A terminal situation of the algorithm MS_4 for inner approximation of the $(\delta + \epsilon)$-minimizer.

The Moore-Skelboe algorithm does not address the "clustering problem", explained below. As we restrict ourselves to minimal elaborations of the Moore-Skelboe algorithm, our versions have the same defect.

Algorithms of this type need to decide whether to include a box B in the list of boxes to be returned. Often, the interval $f(B)$ is so wide that the decision cannot be taken. Accordingly, B is split into subboxes for which the decision still cannot be taken. It may happen that after a number splits all descendants of B are all included. In the algorithms discussed here, all descendants appear separately in the output list. One can think of these as a *cluster* in the output list and call this phenomenon the "clustering problem".

In the situation described here it is unavoidable to decide the many descendants of B in order to decide B itself. But the cluster problem is solved by maintaining the tree structure of the successive splits so that B can be returned on the output list to replace the long list of its descendants. Such an algorithm was described in [12]. The algorithms here need to be modified accordingly.

6 Conclusion

Ratschek and Rokne [11] state properties of the Moore-Skelboe algorithm in the limit for infinite running time, infinite memory, and infinite precision of the floating-point number system. In this paper we find properties that can be verified in actual executions of the Moore-Skelboe algorithm.

We isolate the global optimization problem in the strict sense of the fathoming problem. In addition we consider the localization problem, for which we present an algorithm that yields a set of boxes containing the δ-minimizer. If their union is small, we know that there is a well-defined global minimum. Another algorithm yields a set of boxes contained in a δ-minimizer. If their union is large, then we know that there are widely separated points with objective function values near the global minimum. Obtaining these inner and outer approximations is one way of doing a sensitivity analysis on the optimization problem.

Acknowledgments

We are grateful to Piotr Kaminski for his insightful remarks on an early draft of this paper. The anonymous referee made several valuable suggestions. We acknowledge generous support by the University of Victoria, the Natural Science and Engineering Research Council NSERC, the Centrum voor Wiskunde en Informatica CWI, and the Nederlandse Organisatie voor Wetenschappelijk Onderzoek NWO.

References

[1] L. G. Casado, I. García, and T. Csendes. A heuristic rejection criterion in interval global optimization algorithms. *BIT Numerical Mathematics*, 41(4):683–705, 2001. http://citeseer.nj.nec.com/302739.html.

[2] H.M. Chen and M.H. van Emden. Global optimization with Hypernarrowing. In *Proceedings 1997 SIAM Annual Meeting*, Stanford, California, 1997.

[3] Eldon Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekker, 1992.

[4] Pascal Van Hentenryck, Laurent Michel, and Yves Deville. *Numerica: A Modeling Language for Global Optimization*. MIT Press, 1997.

[5] J. Hooker. *Logic-Based Methods for Optimization - Combining Optimization and Constraint Satisfaction*. Wiley-Interscience series in discrete mathematics and optimization. John Wiley and Sons, 2000.

[6] K. Ichida and Y. Fujii. An interval arithmetic method for global optimization. *Computing*, 23(1):85–97, February 1979.

[7] R. Baker Kearfott. A review of techniques in the verified solution of constrained global optimization problems. In R. Baker Kearfott and Vladik Kreinovich, editors, *Applications of Interval Computations*, pages 23–59. Kluwer, Dordrecht, Netherlands, 1996.

[8] R. Baker Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers, 1996. Nonconvex Optimization and Its Applications.

[9] Vladik Kreinovich and Tibor Csendes. Theoretical justification of a heuristic subbox selection criterion for interval global optimization. http://citeseer.nj.nec.com/kreinovich01theoretical.html.

[10] János Pintér. *Global Optimization in Action*. Kluwer, 1996.

[11] H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization*. Ellis Horwood/John Wiley, 1988.

[12] M.H. van Emden. The logic programming paradigm in numerical computation. In Krzysztof R. Apt, Victor W. Marek, Miroslaw Truszczynski, and David S. Warren, editors, *The Logic Programming Paradigm*, pages 257–276. Springer-Verlag, 1999.