Journal of Global Optimization **25:** 345–362, 2003.

New Interval Analysis Support Functions Using Gradient Information in a Global Minimization Algorithm

L.G. CASADO[1], J.A. MARTÍNEZ[1], I. GARCÍA[1] and YA.D. SERGEYEV[2]
[1]*Computer Architecture & Electronics Dpt., University of Almería, Cta. Sacramento SN, 04120 Almería. Spain. E-mail: leo@ace.ual.es* [2]*ISI-CNR c/o DEIS, Universitá della Calabria, 87036 Rende (CS) Italy, and University of Nizhni Novgorod, Nizhni Novgorod, Russia. E-mail: yaro@si.deis.unical.it*

Abstract. The performance of interval analysis branch-and-bound global optimization algorithms strongly depends on the efficiency of selection, bounding, elimination, division, and termination rules used in their implementation. All the information obtained during the search process has to be taken into account in order to increase algorithm efficiency, mainly when this information can be obtained and elaborated without additional cost (in comparison with traditional approaches). In this paper a new way to calculate interval analysis support functions for multiextremal univariate functions is presented. The new support functions are based on obtaining the same kind of information used in interval analysis global optimization algorithms. The new support functions enable us to develop more powerful bounding, selection, and rejection criteria and, as a consequence, to significantly accelerate the search. Numerical comparisons made on a wide set of multiextremal test functions have shown that on average the new algorithm works almost two times faster than a traditional interval analysis global optimization method.

Key words: Global optimization, Interval arithmetic, Branch-and-bound

1. Introduction

In this paper the problem of finding the global minimum f^* of a real valued one-dimensional continuously differentiable function $f : S \to \mathbb{R}$, $S \subset \mathbb{R}$, and the corresponding set S^* of global minimizers is considered, i.e.:

$$f^* = f(s^*) = \min\{f(s) : s \in S\}, \quad s^* \in S^*. \tag{1}$$

In contrast to one-dimensional local optimization problems which were very well studied in the past, univariate global optimization problems are in the area of interest of many researchers nowadays (see, for example, [1, 2, 3, 4, 6, 10, 9, 11, 14, 16, 15, 17, 21, 27, 30, 32]). Such an interest is explained by the existence of a large number of applications where it is necessary to solve this kind of problem (see [2, 10, 12, 29, 30, 31]). On the other hand, numerous approaches (see, for example, [5, 11, 13, 18, 19, 22, 23, 28, 31]) enable us to generalize to the multidimensional case methods developed to solve univariate problems.

In those cases where the objective function $f(x)$ is given by a formula, it is possible to use an interval analysis branch-and-bound approach to solve problem (1) (see [13, 19, 20, 23]). A general global optimization algorithm based on this approach is shown in Algorithm 1.

ALGORITHM 1. A general interval branch-and-bound global optimization algorithm.

Funct IGO(S, f)
1. Set the working list $L := \{S\}$ and the final list $Q := \{\}$
2. **while** ($L \neq \{\}$)
3. Select an interval X from L
4. **if** X cannot be eliminated
5. Divide X generating $X_i,\ i = 1 \ldots n$
6. **if** X_i satisfies the termination criterion
7. Store X_i in Q
8. **else**
9. Store X_i in L
10. **return** Q

This algorithm selects the next interval to be processed (selection rule, line 3), which can be totally or partially rejected when it is guaranteed that it does not contain any global minimizer, (elimination rule, line 4). This elimination process is carried out using information obtained from the inclusion functions which return an enclosure of the real range of $f(x)$ (and in some cases of $f'(x)$ and $f''(x)$) on X (bounding rule). If the interval cannot be rejected, it is subdivided (division rule, line 5). When the generated subintervals are informative enough, they are stored in the final list (termination rule, line 7). Otherwise, they are stored in the working list for further processing (line 9). The algorithm finishes when there are no intervals to be processed (line 2) and returns the set of intervals with valuable information (line 10). An overview on theory and history of these rules can be found, for example, in [13].

Of course, every concrete realization of Algorithm 1 depends on the available information about the objective function $f(x)$. In this paper it is supposed that inclusion functions can be evaluated for $f(x)$ and its first derivative $f'(x)$ on X. Thus, the information about the objective function which can be obtained during the search is:

$$F(x),\ F(X),\ F'(X), \tag{2}$$

where

- $X = [\underline{x}, \overline{x}]$: Interval defined by its lower and upper bound.
- $F(X) = [\underline{F}(X), \overline{F}(X)]$: Interval inclusion function of $f(x)$ at X obtained by interval arithmetic. For the real range of $f(x)$ over X the following inclusion $f(X) \subseteq F(X)$ holds.

- $F(x)$: Interval inclusion of the function $f(x)$ at a point $x \in X$.
- $F'(X) = [\underline{F'}(X), \overline{F'}(X)]$: Interval inclusion function of $f'(x)$ over X.

When the information stated in (2) is available, the rules of a traditional realization of Algorithm 1 can be written more precisely. Below we describe a Traditional Interval analysis global minimization Algorithm with Monotonicity test (TIAM) which is frequently used for solving the problem (1) using the information stated in (2) (see [13]).

Selection rule: Among all the intervals X_i stored in the working list L select an interval X such that $\underline{F}(X) = \min\{\underline{F}(X_i) : X_i \in L\}$.

Bounding rule: The fundamental theorem of interval arithmetic provides a natural and rigorous way to compute an *inclusion function*. In the present study the inclusion function F of the objective function f is available by the extended interval arithmetic (see [7, 13]).

Elimination rule: Common elimination rules are the following:

Midpoint test: An interval X is rejected when $\underline{F}(X) > \overline{f^{\sim}}$, where $\overline{f^{\sim}}$ is the best known upper bound of f^*. The value of $f^{\sim} = [\underline{f^{\sim}}, \overline{f^{\sim}}]$ is updated by evaluating $F(m(X))$, where $m(X) = (\underline{x} + \overline{x})/2$ is the midpoint of the interval X.

Cut-off test: When $\overline{f^{\sim}}$ is improved, all intervals X stored in the working and final lists satisfying the condition $\underline{F}(X) > \overline{f^{\sim}}$ are rejected.

Monotonicity test: If for an interval X the condition $0 \notin F'(X)$ is fulfilled, then this means that the interval X does not contain any minimum and, therefore, can be rejected.

Division rule: Usually two subintervals are generated using $m(X)$ as the subdivision point (bisection).

Termination rule: A parameter ϵ determines the desired accuracy of the problem solution. Therefore, intervals X that have a width $w(X)$ less than ϵ, i.e., $w(X) \leqslant \epsilon$, $w(X) = \overline{x} - \underline{x}$, are moved to the final list Q. Other termination criteria can be found in [23].

As can be seen from the above description, the algorithm evaluates lower bounds for $f(x)$ in each interval separately, without considering some valuable information which can be obtained from other intervals. The value of $F'(X)$ is only used by the Monotonicity test and is not connected with the information obtained from $F(m(X))$ and $F(X)$. Only the value of $F(X)$ is used in order to obtain a lower bound for $f(x)$ over X, all the rest of the search information is not used for this goal. The only exchange of information between the intervals is done through $f^{\sim}$.

In Lipschitz global optimization there exist algorithms for solving the problem stated in (1). They evaluate lower bounds by constructing support functions (in fact,

$\underline{F}(X)$ can be viewed as a special support function – constant – for $f(x)$ over X) for the objective function (see, for example, [6, 10, 11, 17, 18, 21, 22, 26, 27, 30, 31]). They work in a similar way to the TIAM approach: support functions are built and successively improved in order to obtain a better lower bound for the global minimum. Of course, these support functions are completely different and are built on the basis of different ideas. An interesting aspect of the support function concept in the context of this paper is found in the use of the search information. When a support function is built for an interval $[a, b]$, the information regarding neighbours $[c, a]$ and $[b, d]$ is also used in order to construct a better support function and to obtain a better lower bound for f^*.

In this paper, a new Interval analysis global minimization Algorithm using Gradient information (IAG) is proposed for solving problem (1). It uses the information stated in (2) as TIAM does but, due to a more efficient usage of the search information, it constructs support functions which are closer to the objective function and enables us to obtain better lower bounds. Hereinafter it will be shown that this new method (IAG) has quite a promising performance in comparison with the traditional TIAM method.

The rest of the paper is structured as follows: In Section 2 some theoretical results explaining construction of the support functions and lower bounds are presented. The algorithm IAG is described in Section 3. Numerical experiments comparing performance of TIAM and IAG are presented in Section 4. Finally, in Section 5 some conclusions and future work are presented.

2. New support functions based on gradient information

In order to proceed with the description of the new algorithm, theoretical results are presented to explain how the new support functions and the corresponding lower bounds are constructed in IAG. The description starts with the following lemma illustrated in Figure 1, in a similar way as was done for non-differentiable functions in both, Lipschitz optimization (see [9, 11, 18, 21, 22, 27, 30, 31]) and approaches based on slope evaluation [24].

LEMMA 1. *Given a continuously differentiable function $f : S \to \mathbb{R}$, where S is a closed interval in $\mathbb{R}$, an interval $X \subseteq S$, an enclosure $F(c)$ of $f(c)$, $c \in X$, and an enclosure $F'(X)$ of $f'(x)$, $x \in X$ then the following bounds hold for $f(x)$, $x \in X$:*

$$\underline{F}(c) + \min \left\{ \begin{array}{c} \overline{F'}(X) \cdot (x - c) \\ \underline{F'}(X) \cdot (x - c) \end{array} \right\} \leqslant f(x) \leqslant \overline{F}(c) + \max \left\{ \begin{array}{c} \overline{F'}(X) \cdot (x - c) \\ \underline{F'}(X) \cdot (x - c) \end{array} \right\}$$

Proof. It follows from the mean-value theorem that there exists a point $\xi \in [x, c]$ such that

$$f(x) = f(c) + f'(\xi) \cdot (x - c). \tag{3}$$

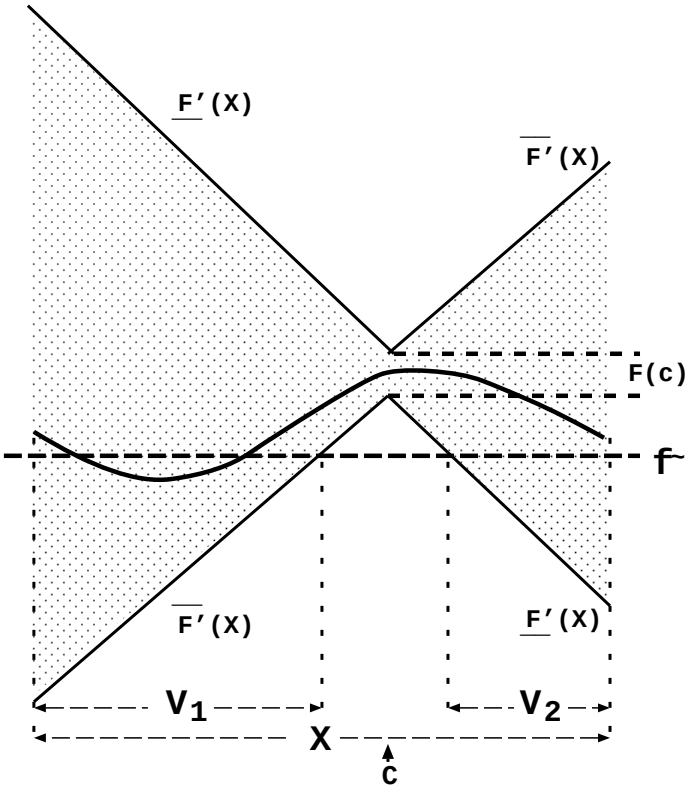

Figure 1. Graphical example of Lemma 1 and Theorem 2.

Extending equation (3) to intervals the following inclusion

$$f(x) \subseteq F(c) + F'(X) \cdot (x - c), \qquad x, c \in X, \tag{4}$$

is obtained. Let us take a generic point $x \in X$. Three results can be deduced from (4) depending on the mutual disposition of the points c and x in X:

(a) $x = c$:

$$f(x) \subseteq F(c).$$

(b) $x > c$:

$$f(x) \leqslant \overline{F}(c) + \overline{F'}(X) \cdot (x - c),$$

$$f(x) \geqslant \underline{F}(c) + \underline{F'}(X) \cdot (x - c).$$

(c) $x < c$:

$$f(x) \leqslant \overline{F}(c) - \underline{F'}(X) \cdot (c - x) = \overline{F}(c) + \underline{F'}(X) \cdot (x - c),$$

$$f(x) \geqslant \underline{F}(c) - \overline{F'}(X) \cdot (c - x) = \underline{F}(c) + \overline{F'}(X) \cdot (x - c).$$

Lemma is proved.

This Lemma allows us to construct new interval analysis support functions for $f(x)$. It can be seen from Figure 1 that it is similar to the ones built in Lipschitz global optimization (see, for example, [10, 21, 22, 26, 31]). The Lipschitz support functions are piece-wise linear. The slope of each linear piece is L or $-L$, where L is the Lipschitz constant. In our approach, for every interval X the slopes of support functions are equal to $\overline{F'}(X)$ for all $x \leqslant c$ and to $\underline{F'}(X)$ for all $x \geqslant c$ (see Figure 1).

The following results are the basis for the new support functions and explain how the new lower bounds for f^* are evaluated.

THEOREM 2. *Let X and S be closed intervals such that $X \subseteq S \subset \mathbb{R}$ and let $f : S \to \mathbb{R}$ be a continuously differentiable function. Let's suppose that for a point $c \in X$ a lower bound $\underline{lb}(c)$ of $f(c)$ is determined and an enclosure $F'(X)$ of $f'(X)$ is obtained. For a given current upper bound $\overline{f^*}$ of f^*, there exists a set $V \subseteq X$ where all the global minimizers of X, if any, are included.*

Proof. For a minimizer point $x^* \in S^*$ it applies that $f(x^*) \leqslant \overline{f^*}$. Combining with Lemma 1 a minimizer $x^* \in X \cap S^*$ has to fulfill:

$$\underline{lb}(c) + \min \left\{ \begin{array}{l} \overline{F'}(X) \cdot (x^* - c) \\ \underline{F'}(X) \cdot (x^* - c) \end{array} \right\} \leqslant f(x^*) \leqslant \overline{f^*}$$

and therefore it can only be located in the following set:

$$V = \left\{ x \in X : \underline{lb}(c) + \min \left\{ \begin{array}{l} \overline{F'}(X) \cdot (x - c) \\ \underline{F'}(X) \cdot (x - c) \end{array} \right\} \leqslant \overline{f^*} \right\}$$

From Theorem 2 it can be derived that if $\overline{f^*} < \underline{lb}(c)$ then $c \notin S^*$ and $V = V_1 \cup V_2$ can be constructed as:

$$V_1 = \left\{ \begin{array}{ll} \{x \in X : x \leqslant c - \frac{\underline{lb}(c) - \overline{f^*}}{\overline{F'}(X)}\}, & when \ \overline{F'}(X) > 0 \\ \emptyset & otherwise \end{array} \right. \tag{5}$$

$$V_2 = \left\{ \begin{array}{ll} \{x \in X : x \leqslant c - \frac{\underline{lb}(c) - \overline{f^*}}{\underline{F'}(X)}\}, & when \ \underline{F'}(X) < 0 \\ \emptyset & otherwise \end{array} \right. \tag{6}$$

As an example, V_1, V_2 have been depicted in Figure 1 for the case $\overline{F'}(X) > 0$ and $\underline{F'}(X) < 0$.

Notice that V can be obtained in a similar way by applying the interval Newton operator to find the roots of the function $f(x) - \overline{f^*}$ on X, under the Theorem 2 conditions [8, 19, 20].

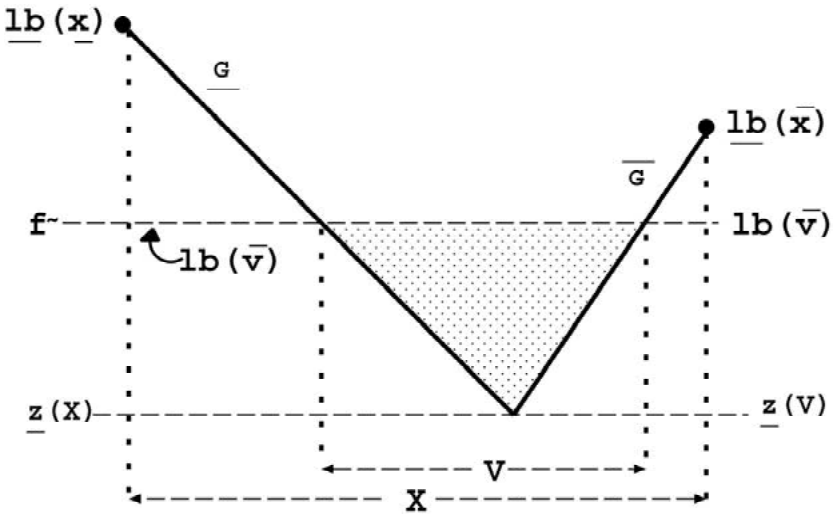

Figure 2. Interval V is the region which can contain global minimizers. The set $X \backslash V$ does not contain any global minimizer.

THEOREM 3. *Let us consider a continuously differentiable function $f : S \to \mathbb{R}$, where S is a closed interval in $\mathbb{R}$ and intervals X, Y such that $X \subseteq Y \subseteq S$. If:*

1. *lower bounds $\underline{lb}(\underline{x})$ and $\underline{lb}(\overline{x})$ of, respectively, $f(\underline{x})$ and $f(\overline{x})$, have been evaluated;*
2. *a current upper bound $\overline{f^{\sim}}$ of f^* is such that*

$$\overline{f^{\sim}} \leqslant \min\{\underline{lb}(\underline{x}), \underline{lb}(\overline{x})\};$$

3. *bounds $\underline{G} = \underline{F'}(Y) < 0$ and $\overline{G} = \overline{F'}(Y) > 0$ have been obtained.*

Then, only the interval

$$V = [\underline{x} - \frac{\underline{lb}(\underline{x}) - \overline{f^{\sim}}}{\underline{G}}, \overline{x} - \frac{\underline{lb}(\overline{x}) - \overline{f^{\sim}}}{\overline{G}}], \qquad V \subseteq X, \tag{7}$$

can contain global minimizers and a lower bound $\underline{z}(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), G)$ of $f(x)$ over the interval X can be calculated as follows:

$$z(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), G) = \frac{\underline{lb}(\underline{x}) \cdot \overline{G} - \underline{lb}(\overline{x}) \cdot \underline{G}}{w(G)} + w(X) \cdot \frac{\underline{G} \cdot \overline{G}}{w(G)}, \tag{8}$$

where $w(G) = \overline{G} - \underline{G}$ (see Figure 2).

Proof. Applying Theorem 2 with $c = \overline{x}$, interval V_2 from (6) is obtained. The same operation with the point $c = \underline{x}$ gives us interval V_1 from (5). Then, interval V from (7) is obtained as $V = V_1 \cap V_2$.

Let us now prove the formula (8). Since $X \subseteq Y$, $F'(X) \subseteq F'(Y)$ and, by applying the mean-value theorem, we have

$$f(x) \geqslant \underline{lb}(\underline{x}) + \underline{F'}(X) \cdot (x - \underline{x}) \geqslant \underline{lb}(\underline{x}) + \underline{G} \cdot (x - \underline{x}), \qquad x \in X,$$

and

$$f(x) \geqslant \underline{lb}(\overline{x}) + \overline{F'}(X) \cdot (x - \overline{x}) \geqslant \underline{lb}(\overline{x}) + \overline{G} \cdot (x - \overline{x}), \quad x \in X.$$

From these two inequalities follows

$$f(x) \geqslant d(x) = \max \left\{ \begin{array}{l} \underline{lb}(\underline{x}) + \underline{G} \cdot (x - \underline{x}) \\ \underline{lb}(\overline{x}) + \overline{G} \cdot (x - \overline{x}) \end{array} \right\}, \quad x \in X. \tag{9}$$

so

$$z(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), G) = \min d(x), \ x \in X$$

and

$$\underline{f}(X) \geqslant z(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), G)$$

which proves the theorem.

COROLLARY 4. *If for an interval X the inequality* $\underline{z}(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), G) > \overline{f^{\sim}}$ *is fulfilled then it can be derived that X does not contain any global minimizer.*

Proof. Proof is evident and so it is omitted.

Let us now return to the problem (1). We can use the information stated in (2) during the global search. Thus, by using $\underline{F}(X)$ together with the function $d(x)$ from (9) we can build a new support function $D(x)$ for $f(x)$ over each interval X:

$$D(x) = \max\{\underline{F}(X), d(x)\}, \quad x \in X. \tag{10}$$

The new lower bound $\underline{Fz}(X)$ for $f(x)$ over the interval X is calculated in the following way

$$\underline{Fz}(X) = \max\{\underline{F}(X), \ \underline{z}(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), G)\} = \min D(x), \ x \in X. \tag{11}$$

The essence of the algorithm is that for $V = [\underline{v}, \overline{v}]$ obtained from interval X according to (7), the current value of $f^{\sim}$ is a lower bound of f at $\underline{v}$ and $\overline{v}$; i.e. $\underline{f^{\sim}} \leqslant f(\underline{v})$ and $\underline{f^{\sim}} \leqslant f(\overline{v})$, so $\underline{lb}(\underline{v}) = \underline{lb}(\overline{v}) = \underline{f^{\sim}}$ are easily available bounds.

3. Description of the new algorithm

On the basis of theoretical results presented in the previous section we can determine new rules for Algorithm 1 in order to introduce the new Interval analysis global minimization Algorithm using Gradient information (IAG) (described in Algorithm 2):

Selection rule : Select an interval X such that

$$\underline{Fz}(X) = \min\{\underline{Fz}(X_i) : X_i \in L\},$$

where L is the working list. Elements of L are ordered by non-decreasing values of $\underline{Fz}(X_i)$ as the first ordering criterion and non-increasing order with respect to the age of the intervals as the second ordering criterion. Therefore, the selected interval will always be at the head of the working list.

Bounding rule : The lower bound $\underline{Fz}(X)$ from (11) is used.

Elimination rule : Four elimination rules are used:

Monotonicity test : As previously described in Section 1.

RangeUp test: An interval X is rejected if $\overline{f^{\sim}} < \underline{Fz}(X)$.

Gradient test: The subregion $\{X \backslash V\}$, where V is defined by (7), is rejected. Of course, when $V = \emptyset$ the whole interval X can be eliminated.

Cut-off test : When $\overline{f^{\sim}}$ is improved, all intervals X stored in the working and final lists for which the condition $\underline{Fz}(X) > \overline{f^{\sim}}$ is fulfilled are rejected. Note, that this Cut-off test is different from the Cut-off test of TIAM algorithm where the condition $\underline{F}(X) > \overline{f^{\sim}}$ was used.

Division rule : Let's suppose an interval X that has been obtained as a result of applying RangeUp and Gradient tests to an interval Y and then also suppose that X is stored in the working list L. If the interval X is chosen for subdivision, $m(X)$ is used as the subdivision point. Note, that in general $m(X) \neq m(Y)$ and therefore this division rule does not coincide with the division rule of the TIAM algorithm.

ALGORITHM 2. Interval analysis global minimization Algorithm using Gradient information (IAG)

Funct IAG(S, F, ϵ)

1. $L = \{\}$; $Q = \{\}$
2. **if** ($\overline{F}(\underline{s}) < \overline{F}(\overline{s})$)
3. $\quad x^{\sim} := \underline{s}$; $f^{\sim} := F(\underline{s})$
4. **else**
5. $\quad x^{\sim} := \overline{s}$; $f^{\sim} := F(\overline{s})$
6. **if** ($0 \notin F'(S)$) *Monotonicity Test*
7. $\quad$ **return** ($f^* := f^{\sim}$; $s^* := x^{\sim}$)
8. $lb(\underline{s}) = F(\underline{s})$; $lb(\overline{s}) = F(\overline{s})$
9. X := GradTest ($S, lb(\underline{s}), lb(\overline{s}), f^{\sim}, F'(S)$)
10. $lb(\underline{x}) := lb(\overline{x}) := f\hat{}(X) := f^{\sim}$
11. $\underline{Fz}(X) := \max\{\underline{F}(X), \underline{z}(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), F'(S))\}$ *Lower bound of* $f(X)$
12. **if** ($w(X) \leqslant \epsilon$)
13. $\quad$ Save $\{X, f\hat{}(X), \underline{Fz}(X)\}$ in Q
14. **else**

15. Save $\{X, f^\wedge(X), \underline{Fz}(X)\}$ in L
16. **while** ($L \neq \{\}$)
17. $\{X, f^\wedge(X), \underline{Fz}(X)\}$:= Head(L)
18. **comment:** $X := X_j : \underline{Fz}(X_j) = \min\{\underline{Fz}(X_i)\},\ \forall X_i \in L$
19. **if** ($0 \in F'(X)$) *Monotonicity Test*
20. **if** ($\overline{F}(m(X)) < \overline{f^\sim}$)
21. $f^\sim := F(m(X))$
22. CutOffTest ($\overline{f^\sim}$, L, Q) *Remove $X_i : \underline{Fz}(X_i) > \overline{f^\sim}$ from {L,Q}*
23. $X_1 = [\underline{x}, m(X)]$; $X_2 = [m(X), \overline{x}]$ *Interval Subdivision*
24. $lb(\underline{x_1}) := lb(\overline{x_2}) := f^\wedge(X)$
25. $lb(\overline{x_1}) := lb(\underline{x_2}) := F(m(X))$
26. **for** i := 1,2
27. X_i := GradTest ($X_i, lb(\underline{x_i}), lb(\overline{x_i}), f^\sim, F'(X)$)
28. $lb(\underline{x_i}) := lb(\overline{x_i}) := f^\wedge(X_i) := f^\sim$
29. **if** ($w(X_i) > 0$) *X_i is not fully rejected*
30. $\underline{Fz}(X_i) := \max\{\underline{F}(X_i), \underline{z}(X_i, \underline{lb}(\underline{x_i}), \underline{lb}(\overline{x_i}), F'(X))\}$
31. **if** ($\underline{Fz}(X_i) \leqslant \overline{f^\sim}$) *RangeUp test*
32. **if** ($w(X_i) \leqslant \epsilon$)
33. Save $\{X_i, f^\wedge(X_i), \underline{Fz}(X_i)\}$ in Q
34. **else**
35. Save $\{X_i, f^\wedge(X_i), \underline{Fz}(X_i)\}$ in L
36. **return** Q, $\tilde{f}$

Every element X_i in the working and final lists, L and Q, respectively, is a structure with the following data:

- Bounds $\underline{x_i}$ and $\overline{x_i}$ of the interval X_i.
- $f^\wedge(X_i) = [\underline{f^\wedge}(X_i), \overline{f^\wedge}(X_i)]$; the value of $f^\wedge(X_i)$ is only updated with the value $f^\sim$ had when the interval X_i was created.
- $\underline{Fz}(X_i)$, a lower bound of $f(X_i)$.

ALGORITHM 3. Gradient test

Funct GradTest($X, lb(\underline{x}), lb(\overline{x}), f^\sim, G$)

1. **if** ($\underline{lb}(\underline{x}) > \overline{f^\sim}$)
2. $V_1 = [\underline{x} - \frac{\underline{lb}(\underline{x}) - \overline{f^\sim}}{\underline{G}},\ \infty)$
3. $X := V_1 \cap X$
4. **if** ($w(X) > 0$ **and** $\underline{lb}(\overline{x}) > \overline{f^\sim}$)
5. $V_2 = (-\infty,\ \overline{x} - \frac{\underline{lb}(\overline{x}) - \overline{f^\sim}}{\overline{G}}]$
6. $X := V_2 \cap X$
7. **return** X

Let us comment Algorithm 2. IAG algorithm starts evaluating $F(\underline{s})$ and $F(\overline{s})$ (line 2) and initializing $x^\sim$ and $f^\sim = F(x^\sim)$ (lines 3 and 5). If the monotonicity

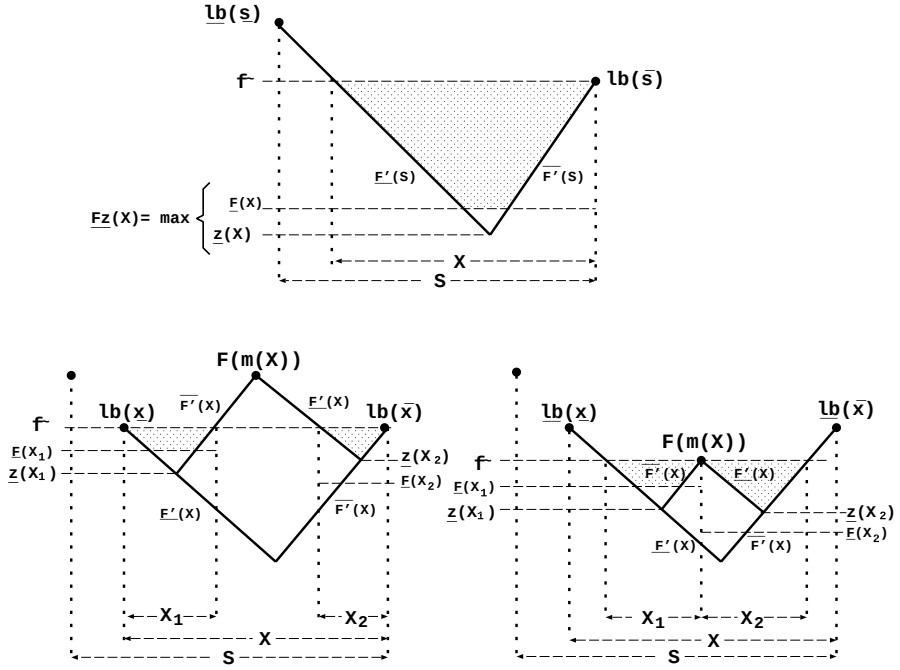

Figure 3. The upper graph is an example of the initial phase of IAG. Bottom graphs show an example of how the intervals X_1 and X_2 are built from X. The bottom left hand graph shows the case $\underline{F}(m(X)) > \overline{f^{\sim}}$ and the bottom right hand $\overline{f^{\sim}} < \min\{\underline{lb}(\underline{x}), \underline{lb}(\overline{x})\}$. Only shaded areas can contain f^*.

test is satisfied (line 6), the algorithm finishes and the solution is given by $\{f^{\sim}, x^{\sim}\}$. Otherwise, $lb(\underline{s})$ and $lb(\overline{s})$ are initialized (line 8) before applying the Gradient test (Algorithm 3). In line 9, the Gradient test is applied to the starting interval S. The GradTest procedure applies Theorems 2 and 3 to S, using $c = \underline{s}$ and $c = \overline{s}$, and returns an interval $X \subseteq S$, such that the set of global minimizers of S are also in X. Lower bounds of $f(\underline{x})$ and $f(\overline{x})$ ($lb(\underline{x})$ and $lb(\overline{x})$, respectively) and $f\hat{}(X)$ are set to $f^{\sim}$, and $\underline{Fz}(X)$ is computed (lines 10 and 11). The interval X is stored in the final list Q (line 13) or in the working list L (line 15), depending on the value of $w(X)$. This initialization stage is shown in the top graph of Figure 3.

After this initialization stage and while the list L is not empty (line 16), IAG will select the interval at the head of L (line 17) for further processing. If $F'(X)$ does not satisfy the Monotonicity test (line 19), $F(m(X))$ is evaluated (line 20). If $F(m(X))$ provides an upper bound of f^* better than $f^{\sim}$, $f^{\sim}$ is updated to $F(m(X))$ (line 21) and the Cut-off test is applied to the intervals in the working and final lists (line 22). The interval X is then divided into two subintervals X_1 and X_2 (line 23). These subintervals inherit from X the lower bound of $f(x)$ in one of their bounds (line 24) and the other shared bound is set to $F(m(X))$ (line 25). For each subinterval X_i, $i = \{1, 2\}$, the Gradient test is carried out using the derivative

information of F at X ($F'(X)$) (line 27) instead of the value of $F'(X_i)$ which has not been evaluated. Using the derivative information of F at X we avoid the need for additional computations of $F'(X_i)$, $(i = \{1, 2\})$ which would be useless if these intervals were never chosen for subdivision. If an interval X_i, $(i = \{1, 2\})$ is not rejected (line 29), $\underline{Fz}(X_i)$ is evaluated using also the value of $F'(X)$ (line 30). Only when the RangeUp test is not satisfied (line 31), the interval X_i will be saved in the final (line 33) or working lists (line 35).

Bottom graphs of Figure 3 show how the intervals X_1 and X_2 are built from X. In case of $\underline{F}(m(X)) > \overline{f^{\sim}}$ (see lower left hand graph), interval X_1 and X_2 can be shortened by applying the GradTest procedure. In addition, if $\overline{f^{\sim}} < \underline{lb}(\underline{x})$ and/or $\overline{f^{\sim}} < \underline{lb}(\overline{x})$ (see lower right hand graph of Figure 3), X_1 and/or X_2 can be shortened again by the GradTest procedure. Notice that in the IAG algorithm, if $\overline{f^{\sim}} < \underline{lb}(\underline{x})$ then $\overline{f^{\sim}} < \underline{lb}(\overline{x})$, too, and vice versa.

4. Numerical results

The new algorithm IAG has been numerically compared with the method TIAM on a set of forty test functions. This set of test functions is described in Table 1⋆ and has been taken from [3, 4, 24]. The search region, the number of local and global minimizers and the numerical value of the global optimum (rounded to six decimal digits) are shown for all the functions. For both algorithms the stopping criterion was: $w(X) \leqslant \epsilon = 10^{-6}$.

Table 2 shows numerical comparison between TIAM and IAG. Column NFE presents the number of interval function evaluations, i.e., the number of $F(X)$ evaluations plus the number of interval point evaluations $F(x)$, column NDE shows the number of interval function evaluations of the derivative $F'(X)$, and column $w(f^*)$ shows the approximated width of the interval containing the global optimum (given by $[\underline{F}(X), \overline{\tilde{f}}]$ and $[\underline{Fz}(X), \overline{\tilde{f}}]$ for TIAM and IGO algorithms, respectively, with X=Head(Q)). If TM and TG represent $NFE + NDE$ for algorithms TIAM and IAG, respectively, column TIAM/IAG shows the values TM/TG, providing information on the relative speedup of the IAG algorithm compared to the TIAM algorithm.

It can be seen from Table 2 that the ratio $TIAM/IAG$ is always greater than one, so IAG outperforms TIAM for all the functions. For this set of functions the speedup $TIAM/IAG$ ranges between [1.22, 6.98] and in average is 1.78. It can also be seen from Table 2 that for those functions where TIAM needs a lot of func-

⋆ For functions 31 and 35 the values of a_i, k_i and c_i are:
a = (3.040, 1.098, 0.674, 3.537, 6.173, 8.679, 4.503, 3.328, 6.937, 0.700),
k = (2.983, 2.378, 2.439, 1.168, 2.406, 1.236, 2.868, 1.378, 2.348, 2.268),
c = (0.192, 0.140, 0.127, 0.132, 0.125, 0.189, 0.187, 0.171, 0.188, 0.176) and
a = (4.696, 4.885, 0.800, 4.986, 3.901, 2.395, 0.945, 8.371, 6.181, 5.713),
k = (2.871, 2.328, 1.111, 1.263, 2.399, 2.629, 2.853, 2.344, 2.592, 2.929),
c = (0.149, 0.166, 0.175, 0.183, 0.128, 0.117, 0.115, 0.148, 0.188, 0.198), respectively.

Table 1. Description of the test functions. N: function's ID , S: the search interval, LM: the number of local minimizers, GM: the number of global minimizers, and f^*: the value of global minimum (rounded to six decimal digits)

N	Function $f(x)$	S	LM	GM	f^*
1	$e^{-3x} - \sin^3 x$	$[0, 20]$	4	1	$e^{\frac{-27\pi}{2}} - 1$
2	$\sum_{k=1}^{5} -\cos[(k+1)x] + 4$	$[0.2, 7.0]$	7	1	-1.0
3	$(x - x^2)^2 + (x-1)^2$	$[-10, 10]$	1	1	0.0
4	$(3x - 1.4)\sin(18x) + 1.7$	$[0.2, 7.0]$	21	1	-17.582872
5	$2x^2 - 3/100e^{-(200(x-0.0675))^2}$	$[-10, 10]$	1	1	-0.020903
6	$\cos(x) - \sin(5x) + 1$	$[0.2, 7.0]$	6	1	-0.952897
7	$-x - \sin(3x) + 1.6$	$[0.2, 7.0]$	4	1	-6.262872
8	$x + \sin(5x)$	$[0.2, 7.0]$	7	1	-0.077590
9	$-e^{-x}\sin(2\pi x) + 1$	$[0.2, 7.0]$	7	1	0.211315
10	$e^{-x}\sin(2\pi x)$	$[0.2, 7.0]$	7	1	-0.478362
11	$-x + \sin(3x) + 1$	$[0.2, 7.0]$	5	1	-5.815675
12	$x\sin(x) + \sin(10x/3) + \ln(x) - 0.84x + 1.3$	$[0.2, 7.0]$	4	1	-7.047444
13	$\sin(x) + \sin(10x/3) + \ln(x) - 0.84x$	$[2.7, 7.5]$	3	1	-4.601308
14	$\ln(3x)\ln(2x) - 0.1$	$[0.2, 7.0]$	1	1	-0.141100
15	$\sum_{k=0}^{5} k\cos[(k+1)x + k] + 12$	$[0.2, 7.0]$	8	1	-0.870885
16	$-\sum_{k=1}^{5} k\sin[(k+1)x + k] + 3$	$[0.2, 7.0]$	7	1	-9.031249
17	$\sin^2(1 + (x-1)/4) + ((x-1)/4)^2$	$[-10, 10]$	1	1	0.475689
18	$\sqrt{x}\sin^2(x)$	$[0.2, 7.0]$	3	2	0.0
19	$x^2 - \cos(18x)$	$[-5, 5]$	29	1	-1.0
20	e^{x^2}	$[-10, 10]$	1	1	1.0
21	$(x^2/20) - \cos(x) + 2$	$[-20, 20]$	7	1	1.0
22	$\cos(x) + 2\cos(2x)e^{-x}$	$[0.2, 7.0]$	2	1	-0.918397
23	$(x + \sin(x))e^{-x^2}$	$[-10, 10]$	1	1	-0.824239
24	$2\sin(x)e^{-x}$	$[0.2, 7.0]$	2	1	-0.027864
25	$2\cos(x) + \cos(2x) + 5$	$[0.2, 7.0]$	3	2	3.5
26	$e^{\sin(3x)}$	$[0.2, 7.0]$	5	3	0.367879
27	$\sin(x)\cos(x) - 1.5\sin^2(x) + 1.2$	$[0.2, 7.0]$	3	2	-0.451388
28	$\sin(x)$	$[0, 20]$	4	3	-1.0
29	$2(x-3)^2 - e^{x/2} + 5$	$[0.2, 7.0]$	1	1	-0.410315
30	$-e^{\sin(3x)} + 2$	$[0.2, 7.0]$	4	4	-0.718282
31	$-\sum_{i=1}^{10} 1/((k_i(x - a_i))^2 + c_i)$	$[0, 10]$	8	1	-14.592652
32	$\sin(1/x)$	$[0.02, 1]$	6	6	-1.0
33	$-\sum_{k=1}^{5} k\sin((k+1)x + k)$	$[-10, 10]$	20	3	-12.031249
34	$(x^2 - 5x + 6)/(x^2 + 1) - 0.5$	$[0.2, 7.0]$	1	1	-0.535534
35	$-\sum_{i=1}^{10} 1/((k_i(x - a_i))^2 + c_i)$	$[0, 10]$	7	1	-13.922345
36	$(x+1)^3/x^2 - 7.1$	$[0.2, 7.0]$	1	1	-0.35
37	$x^4 - 12x^3 + 47x^2 - 60x - 20e^{-x}$	$[-1, 7]$	1	1	-32.781261
38	$x^6 - 15x^4 + 27x^2 + 250$	$[-4, 4]$	2	2	7.0
39	$x^4 - 10x^3 + 35x^2 - 50x + 24$	$[-10, 20]$	2	2	-1.0
40	$24x^4 - 142x^3 + 303x^2 - 276x + 3$	$[0, 3]$	1	1	1.0

Table 2. Results of numerical comparison between TIAM and IAG

	TIAM			IAG			
N	NFE	NDE	$w(f^*)$	NFE	NDE	$w(f^*)$	TIAM/IAG
1	104	27	6.2e-14	75	20	2.9e-14	1.38
2	104	27	2.9e-13	72	21	4.7e-12	1.41
3	107	27	1.1e-13	88	22	3.0e-13	1.22
4	105	34	2.4e-06	79	25	1.3e-09	1.34
5	112	35	1.6e-07	88	26	9.0e-12	1.29
6	110	39	2.4e-07	75	25	1.4e-11	1.49
7	112	41	8.1e-07	69	25	8.9e-14	1.63
8	112	41	8.1e-07	66	22	1.1e-11	1.74
9	115	41	6.4e-07	76	25	4.0e-13	1.54
10	116	42	3.9e-07	83	28	5.3e-12	1.42
11	118	44	8.1e-07	79	28	1.4e-12	1.51
12	118	44	1.9e-06	74	26	3.5e-12	1.62
13	132	49	4.8e-07	81	30	5.0e-12	1.63
14	139	50	4.0e-07	91	31	7.4e-12	1.55
15	144	49	9.7e-06	102	34	2.3e-11	1.42
16	152	51	1.1e-05	113	35	3.0e-11	1.37
17	167	63	1.3e-07	114	39	3.7e-14	1.50
18	184	47	5.5e-15	140	38	1.4e-13	1.30
19	191	48	4.4e-16	77	23	5.5e-16	2.39
20	199	50	1.1e-15	116	38	1.1e-15	1.62
21	207	52	4.4e-16	114	36	6.6e-16	1.73
22	209	75	8.9e-08	132	46	4.0e-13	1.60
23	218	81	6.7e-07	146	50	9.6e-13	1.53
24	220	83	2.3e-08	152	50	4.7e-14	1.50
25	231	88	1.4e-06	167	61	5.3e-13	1.40
26	268	68	2.6e-14	214	57	4.3e-14	1.24
27	247	92	7.4e-07	191	67	1.3e-12	1.31
28	292	74	7.2e-15	206	55	2.8e-15	1.40
29	301	113	2.8e-06	174	60	1.6e-12	1.77
30	352	89	4.9e-15	270	71	9.1e-14	1.29
31	139	47	5.1e-06	113	35	2.8e-10	1.26
32	476	120	2.6e-11	384	101	2.2e-12	1.23
33	459	154	8.2e-06	333	108	2.8e-10	1.39
34	460	176	5.9e-07	259	91	7.4e-13	1.82
35	599	204	1.2e-05	293	96	5.3e-10	1.50
36	711	271	2.7e-06	331	114	2.4e-12	2.21
37	824	276	7.5e-05	288	101	5.9e-11	2.83
38	807	310	1.5e-03	498	176	1.2e-08	1.66
39	6041	2265	4.0e-04	1364	456	2.3e-09	4.56
40	6952	2534	1.4e-03	1020	340	2.9e-09	6.98

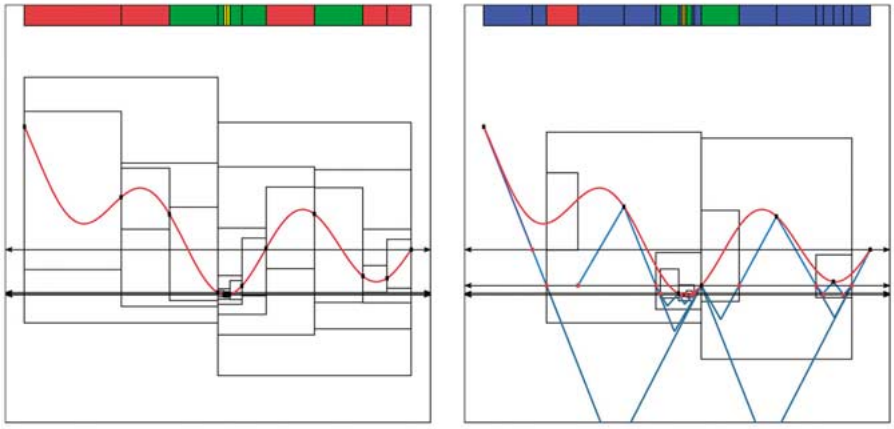

Figure 4. Graphical representation for the execution of TIAM (left hand graph) and IAG (right hand graph) algorithms for function $N = 13$.

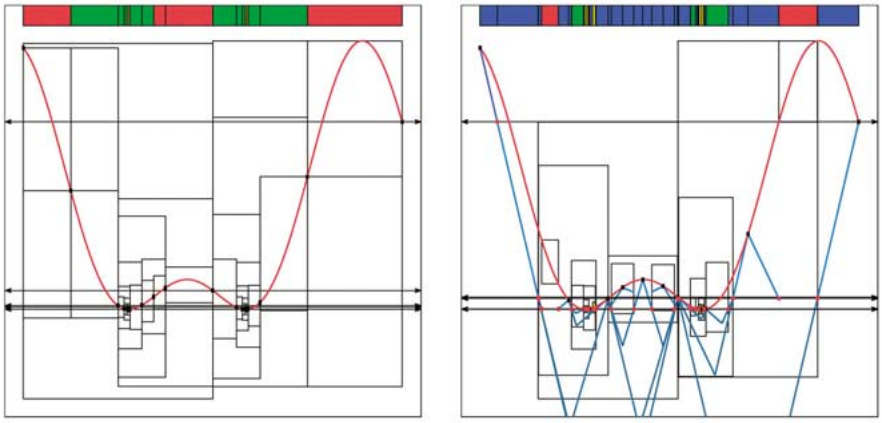

Figure 5. Graphical representation for the execution of TIAM (left hand graph) and IAG (right hand graph) algorithms for function $N = 25$.

tion evaluations the largest values of $TIAM/IAG$ were obtained (see functions $N = 39$ and $N = 40$, which obtained speed up of 4.56 and 6.98, respectively). For most of the functions (33 out of 40) the values of $w(f^*)$ are narrower for IAG algorithm than for TIAM one.

Figures 4 and 5 graphically show how algorithms TIAM and IAG work. The function $N = 13$ presented in Figure 4 has only one global minimizer while the function $N = 25$ (Figure 5) has two global minimizers. In both figures the left hand graph refers to algorithm TIAM while right hand graphs depict the performance of algorithm IAG. For all the graphs the termination criterion was $w(X) \leqslant 0.05$. Horizontal arrows represent the values of $\overline{f^{\sim}}$ during the execution. The boxes represent the margins of all the evaluated intervals X and the lower and upper bounds

of $F(X)$. For the IAG algorithm, the new support function $d(x)$ from (9) is also shown.

At the top of the graphs, colored boxes represent the set of rejected intervals as well as intervals which contain a global minimizer. The color of a box specifies the criterion responsible for the rejection of that interval (Blue = GradTest procedure, Green = monotonicity test, Red = midpoint and cut-off tests for TIAM and RangeUp and cut-off tests for IAG, Yellow = Boxes in the final list Q). From these graphs it is easy to realize how efficient every rejection criterion is.

Figures 4 and 5 show that for these examples more than 50% of the initial interval S was rejected due to the GradTest procedure. It is also clearly shown that TIAM had to evaluate more intervals than IAG. Figure 5 shows some intervals where the best lower bound of $f(X)$ was the one obtained by the computation of $\underline{z}(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), F'(X))$ instead of $\underline{F}(X)$, i.e., that $\underline{Fz}(X) = \underline{z}(X, \underline{lb}(\underline{x}), \underline{lb}(\overline{x}), F'(X))$ took place. It should also be noticed that the TIAM algorithm is unable to take advantage of the information provided by the evaluation of $F(m(X))$ when $F(m(X)) > \overline{f^*}$. In contrast, IAG is able to reduce the interval even in this case (clearly shown in Figures 4 and 5).

5. A brief conclusion

In this paper a new way to calculate support functions for multiextremal univariate functions has been presented. The new support functions are based on obtaining the same kind of information used in interval analysis global optimization algorithms: interval evaluations of the objective function at a point, on an interval, and interval evaluation of the first derivative of the objective function on an interval, i.e., $F(x)$, $F(X)$, and $F'(X)$.

Traditional interval analysis global optimization algorithms use this information separately: $F(x)$ is used to obtain an upper bound for the global minimum, $F(X)$ is used to determine a support function – being a constant – for the objective function $f(x)$ over X, and, finally, $F'(X)$ is used in the Monotonicity test for rejecting intervals which do not contain global minimizers. In contrast, the new method uses all the information simultaneously in order to construct a support function which is closer to the objective function. The new support function enables us to develop more powerful rejection and bounding criteria and to significantly accelerate the search. In fact, the new algorithm works almost two times faster in comparison with other traditional interval analysis methods on a wide set of multiextremal test functions.

The new approach has several possibilities for generalization. First, interval analysis bounds for $F'(X)$ can be substituted by other estimates (for example, slope tools developed in [25] for non-smooth problems) in order to obtain new support functions. Second, the new method can be generalized to the multi-dimensional case by the diagonal approach proposed in [22] or by using adaptively constructed space-filling curves proposed in [28].

Acknowledgement

The authors would like to thank E.M.T. Hendrix for his useful remarks and suggestions. This work was supported by the Ministry of Education of Spain (CICYT TIC99-0361) and by the Russian Fund of Basic Research through grant 01-01-00587.

References

1. Calvin, J. and Žilinskas, A. (1999), On the convergence of the P-algorithm for one-dimensional global optimization of smooth functions. *JOTA* 102(3), 479–495.
2. Casado, L. G., García, I. and Sergeyev, Ya. D. (2000), Interval branch and bound algorithm for finding the First-Zero-Crossing-Point in one-dimensional functions. *Reliable Computing* 2(6), 179–191.
3. Daponte, P., Grimaldi, D., Molinaro, A. and Sergeyev, Ya. D. (1995), An algorithm for finding the zero crossing of time signals with Lipschitzeans derivatives. *Measurements* 16, 37–49.
4. Daponte, P., Grimaldi, D., Molinaro, A. and Sergeyev, Ya. D. (1996), Fast detection of the first zero-crossing in a measurement signal set. *Measurements* 19(1), 29–39.
5. Floudas, C. and Pardalos, P. (eds.) (1996), *State of the Art in Global Optimization. Computational Methods and Applications*, Vol. 7 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht.
6. Gergel, V. P. (1999), A global search algorithm using derivatives. In: *Systems Dynamics and Optimization*. N. Novgorod University Press, Norgood, pp. 161–178.
7. Hammer, R., Hocks, M., Kulisch, U. and Ratz, D. (1995), *C++ Toolbox for Verified Computing I: Basic Numerical Problems: Theory, Algorithms, and Programs*. Springer, Berlin.
8. Hansen, E. (1992), *Global Optimization Using Interval Analysis*, Vol. 165 of *Pure and applied mathematics*. Marcel Dekker, New York.
9. Hansen, E., Jaumard, B., and Lu, S.-H. (1992a), Global optimization of univariate Lipschitz functions: 1. Survey and properties. *Math. Programming* 55, 252–272.
10. Hansen, E., Jaumard, B. and Lu, S.-H. (1992b), Global optimization of univariate Lipschitz functions: 2. New algorithms and computational comparison. *Math. Programming* 55, 273–293.
11. Horst, R. and Pardalos, P. (eds.) (1995), *Handbook of Global Optimization*, Vol. 2 of *Noncovex optimization and its applications*. Kluwer Academic Publishers, Dordrecht.
12. Kalra, D. and Barr, A. H. (1989), Guaranteed ray intersections with implicit surfaces. *Computer Graphics* 23(3), 297–306.
13. Kearfott, R. B. (1996), *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers, Dordrecht.
14. Lamar, B. (1999), A method for converting a class of univariate functions into d.c. functions'. *Journal of Global Optimization* 15, 55–71.
15. Liu, Y. and Teo, K. (1999), A bridging method for global optimization. *Journal of Australian Mathematical Society, Series B* 41, 41–57.
16. Locatelli, M. and Schoen, F. (1995), An adaptive stochastic global optimization algorithm for one-dimensional functions. *Annals of Operations Research* 58, 263–278.
17. MacLagan, D., Sturge T. and Baritompa, W. (1996), Equivalent methods for global optimization. In: Floudas, C. and Pardalos, P. (eds.) *State of the Art in Global Optimization. Computational Methods and Applications*, Kluwer Academic Publications, Dordrecht, pp. 201–212.
18. Mladineo, R. (1992), Convergence rates of a global optimization algorithm. *Math. Programming* 54, 223–232.

19. Moore, R. (1966), *Interval analysis*. Prentice-Hall, Englewood Cliffs, NJ.
20. Neumaier, A. (1990), *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge.
21. Pijavskii, S. A. (1972), An algorithm for finding the absolute extremum of a function'. *USSR Maths. Math. Physics* 12, 57–67.
22. Pintér, J. D. (1996), *Global Optimization in Action*, Vol. 6 of *Noncovex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht.
23. Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization*. Ellis Horwood Ltd., Chichester.
24. Ratz, D. (1998), *Automatic Slope Computation and its Application in Nonsmooth Global Optimization*. Shaker Verlag, Aachen.
25. Ratz, D. (1999), A nonsmooth global optimization technique using slopes: the one-dimensional case. *Journal of Global Optimization* 14(4), 365–393.
26. Sergeyev, Ya. D. (1995), A one-dimensional deterministic global minimization algorithm. *Comput. Maths. Math. Phys* 35(5), 705–717.
27. Sergeyev, Ya. D. (1998), Global one-dimensional optimization using smooth auxiliary functions'. *Mathematical Programming* 81(1), 127–146.
28. Sergeyev, Ya. D. (2000), Efficient strategy for adaptive partition of N-dimensional intervals in the framework of diagonal algorithms. *Journal of Optimization Theory and Applications* 107(1), 145–168.
29. Sergeyev, Ya. D., Daponte, P., Grimaldi, D. and Molinaro, A. (1999), Two methods for solving optimization problems arising in electronic measurement and electrical engineering. *SIAM Journal on Optimization* 10(1), 1–21.
30. Strongin, R. G. (1978), *Numerical Methods on Multiextremal Problems*. Nauka, Moscow.
31. Strongin, R. G. and Sergeyev, Ya. D. (2000), *Global optimization with non-convex constraints: Sequential and parallel algorithms*. Kluwer Academic Publishers, Dordrecht.
32. Wang, X. and Chang, T. (1996), An improved univariate global optimization algorithm with improved linear bounding functions. *Journal of Global Optimization* pp. 393–411.