Numerical Algorithms **37:** 61–69, 2004.

On an efficient use of gradient information for accelerating interval global optimization algorithms *

J.A. Martínez [a], L.G. Casado [a], I. García [a], Ya.D. Sergeyev [b] and B. Tóth [c]

[a] *Computer Architecture & Electronics Department, University of Almería Cta. Sacramento SN, 04120 Almería, Spain*
E-mail: leo@ace.ual.es

[b] *DEIS, Universita' della Calabria, Italy, and University of Nizhni Novgorod, Nizhni Novgorod, Russia*
E-mail: yaro@si.deis.unical.it

[c] *University of Szeged, Department of Applied Informatics, H-6701 Szeged, Hungary*
E-mail: boglarka@inf.u-szeged.hu

This paper analyzes and evaluates an efficient n-dimensional global optimization algorithm. It is a natural n-dimensional extension of the algorithm of Casado et al. [1]. This algorithm takes advantage of all available information to estimate better bounds of the function. Numerical comparison made on a wide set of multiextremal test functions has shown that on average the new algorithm works faster than a traditional interval analysis global optimization method.

Keywords: global optimization, interval arithmetic, branch-and-bound

AMS subject classification: 65G, 65K, 90C

1. Introduction and notation

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

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

The following notation is used. $\mathbb{I} = \{X = [a, b] \mid a \leqslant b; a, b \in \mathbb{R}\}$ is the set of the one-dimensional intervals. $X = [\underline{x}, \bar{x}] \in \mathbb{I}$ is a one-dimensional interval. $X = (X_1, \ldots, X_n) \subseteq S$, $X_i \in \mathbb{I}$, $i = 1, \ldots, n$ is an n-dimensional interval, also called box. $\mathbb{I}^n$ is the set of the n-dimensional intervals. $m(X) = (m(X_1), \ldots, m(X_n))$ is the midpoint of X, where $m(X_i) = (\bar{x}_i + \underline{x}_i)/2$, $i = 1, \ldots, n$; $w(X) = \max_{i=1,\ldots,n} w(X_i)$ is the width of X, where $w(X_i) = (\bar{x}_i - \underline{x}_i)$; $f(X) = \{f(x) \mid x \in X\}$ is the real range of f on $X \subseteq S$. F and $F' = (F'_1, \ldots, F'_n)$ are interval extensions of f and its

* This work was supported by the Ministry of Education of Spain (CICYT TIC2002-00228), by the Russian Fund of Basic Research through grant 01-01-00587 and by the grant OTKA T 034350.

derivative f', respectively. The inclusions $f(X) \subseteq F(X)$ and $f'(X) \subseteq F'(X)$ hold. $F_c(X, c) = F(c) + F'(X)(X - c)$, with $c \in X$ and $f(X) \subseteq F_c(X, c)$, is the centered form.

We define the projected(slice) interval, narrowing the ith dimension into a given point $X(i : p) = (X_1, \ldots, X_{i-1}, [p, p], X_{i+1}, \ldots, X_n)$, $p \in X_i$. We are going to use it mostly in three cases, projecting the interval to the left border of the ith dimension: $X_i^l = X(i : \underline{x}_i)$, to the right: $X_i^r = X(i : \bar{x}_i)$, and to the midpoint: $X_i^m = X(i : m(X_i))$. The lower bound of $f(X)$ is $lbf(X) \in \mathbb{R}$ satisfying $lbf(X) \leqslant f(x)$, $\forall x \in X$, and the support function of $f(X)$ at the border of the interval X is $sp(X) = (sp(X_1), \ldots, sp(X_n))$, where $sp(X_i) = \{sp(X_i^l), sp(X_i^r)\} = \{lbf(X_i^l), lbf(X_i^r)\}$, $i = 1, \ldots, n$.

In those cases where the objective function $f(x)$ is given by a formula, it is possible to use an interval analysis B&B approach to solve problem (1) (see [7–10]). A general interval GO (IGO) algorithm based on this approach is shown in algorithm 1.

Algorithm 1. A general interval B&B GO algorithm.
Funct IGO(S, f)

```
Set the working list L := {S} and the final list Q := {}
while ( L ≠ {} )
   Select an interval X from L                          Selection rule
   Compute lbf(X)                                       Bounding rule
   if X cannot be eliminated                            Elimination rule
      Divide X into subintervals X^j, j = 1, ..., r     Division rule
      if X^j satisfies the termination criterion        Termination rule
         Store X^j in Q
      else
         Store X^j in L
return Q
```

An overview on theory and history of the rules of this algorithm can be found, for example, in [7]. 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:

$$F(x), \quad F(X) \quad \text{and} \quad F'(X). \tag{2}$$

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 Multidimensional Traditional Interval analysis global minimization Algorithm with Monotonicity test (MTIAM) which is frequently used to solve the problem (1), using the information stated in (2) (see [7]).

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

- **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 [5,7] ($lbf(X) = \underline{F}(X)$).
- **Elimination rule**. Common elimination rules are the following:
  - **Midpoint test**. An interval X is rejected when $\underline{F}(X) > \overline{\tilde{f}}$, where $\overline{\tilde{f}}$ is the best known upper bound of f^*. The value of $\tilde{f} = [\underline{\tilde{f}}, \overline{\tilde{f}}]$ is usually updated by evaluating $F(m(X))$.
  - **Cutoff test**. When $\overline{\tilde{f}}$ is improved, all intervals X stored in the working and final lists satisfying the condition $\underline{F}(X) > \overline{\tilde{f}}$ 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 minima (the box is rejected) or the minima is on the border of the search region (the box is reduced).
- **Division rule**. Usually two subintervals are generated using $m(X)$ as the subdivision point (bisection) on direction k, where k is the coordinate such that $w(X_k) = \max_{i=1,\dots,n} w(X_i)$.
- **Termination rule**. A parameter ε determines the desired accuracy of the problem solution. Therefore, intervals X with $w(X) \leqslant \varepsilon$, are moved to the final list Q. Other termination criteria can be found in [10].

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 given information is not used for this goal. The only exchange of information between the intervals is done through $\tilde{f}$.

In this paper, the general framework of a feasible extension for multidimensional functions of the algorithm proposed by Casado et al. [1] is analyzed and evaluated. This extension follows the ideas of Ratz [11].

This new Multidimensional Interval analysis global minimization Algorithm using Gradient information (MIAG) is proposed to solve problem (1). It uses the information stated in (2) as MTIAM does but, due to a more efficient usage of the search information, it constructs support functions, which are closer to the objective function, that enable us to obtain better lower bounds and to diminish the width of the current interval. Hereinafter it will be shown that this new method (MIAG) has quite a promising performance in comparison with the traditional MTIAM method.

The rest of the paper is structured as follows. In section 2 some theoretical results explaining the construction of the support functions and lower bounds are presented and the algorithm MIAG is described. Numerical experiments comparing performance of MTIAM and MIAG are presented in section 3, where some conclusions are also described.

2. The multidimensional algorithm based on new support functions

In order to proceed with the description of the new algorithm, theoretical results are presented which are the foundations of the new support functions and explain how the new lower bounds of $f(X)$ are evaluated. As Ratz has proposed in [11], we will analyze a feasible approach which computes enclosures based on values of particular components of the gradient vector $F'(X)$; i.e. this corresponds to a componentwise derivative computation of the support functions.

Theorem 1. Let X and S be closed intervals such that $X \subseteq S \subset \mathbb{R}^n$ and let $f : S \to \mathbb{R}$ be a continuously differentiable function. Let us suppose that, given a point $c = (c_1, \ldots, c_n) \in X$, for an interval $X(i : c_i) \in X$ a lower bound $lbf(X(i : c_i))$ of $f(X(i : c_i))$ is determined and an enclosure $F'(X)$ of $f'(X)$ is obtained. For a given current upper bound $\overline{\tilde{f}}$ of f^*, there exists a set

$$X^a_{go} = \bigl\{x \in X\colon\ lbf\bigl(X(i : c_i)\bigr) + \min\bigl\{F'_i(X) \cdot (x_i - c_i)\bigr\} \leqslant \overline{\tilde{f}}\bigr\} \subseteq X,$$

where all the global minimizer points of $f(X)$, if any, are included.

Proof. For a minimizer point $x^* \in S^*$ it applies that $f(x^*) \leqslant \overline{\tilde{f}}$. From [1, lemma 1] a minimizer point $x^* \in X \cap S^*$ has to fulfill:

$$lbf\bigl(X(i : c_i)\bigr) + \min\bigl\{F'_i(X) \cdot (x^*_i - c_i)\bigr\} \leqslant f(x^*) \leqslant \overline{\tilde{f}},$$

and therefore it can only be located in the set X^a_{go}. □

It can be derived from theorem 1 that if $\overline{\tilde{f}} < lbf(X(i : c_i))$ then $X(i : c_i) \nsubseteq S^*$ and $X^a_{go} = X \backslash V$, where V is given by

$$V = \left\{x \in X\colon\ x_i \in \left[c_i - \frac{lbf(X(i : c_i)) - \overline{\tilde{f}}}{\overline{F}'_i(X)}, c_i - \frac{lbf(X(i : c_i)) - \overline{\tilde{f}}}{\underline{F}'_i(X)}\right]\right\}$$

when $0 \in F'(X)$. As an example, V have been depicted in figure 1 for the case $c_1 = X^m_1$ and $0 \in F'(X)$. If $0 \notin F'(X)$ then $x^* \notin X$.

Theorem 2. Let us consider a continuously differentiable function $f : S \to \mathbb{R}$, where S is a closed interval in $\mathbb{R}^n$ and intervals X, Y such that $X \subseteq Y \subseteq S$. If for some $i \in \{1, \ldots, n\}$:

1. lower bounds $lbf(X^l_i)$ and $lbf(X^r_i)$ of $f(X^l_i)$ and $f(X^r_i)$, respectively, have been evaluated;
2. a current upper bound $\overline{\tilde{f}}$ of f^* is such that
$$\overline{\tilde{f}} \leqslant \min\bigl\{lbf\bigl(X^l_i\bigr), lbf\bigl(X^r_i\bigr)\bigr\};$$
3. bounds $G_i = F'_i(Y)$ of $f'_i(Y)$ such that $0 \in F'(Y)$, so bounds of $f'_i(X)$, have been obtained.

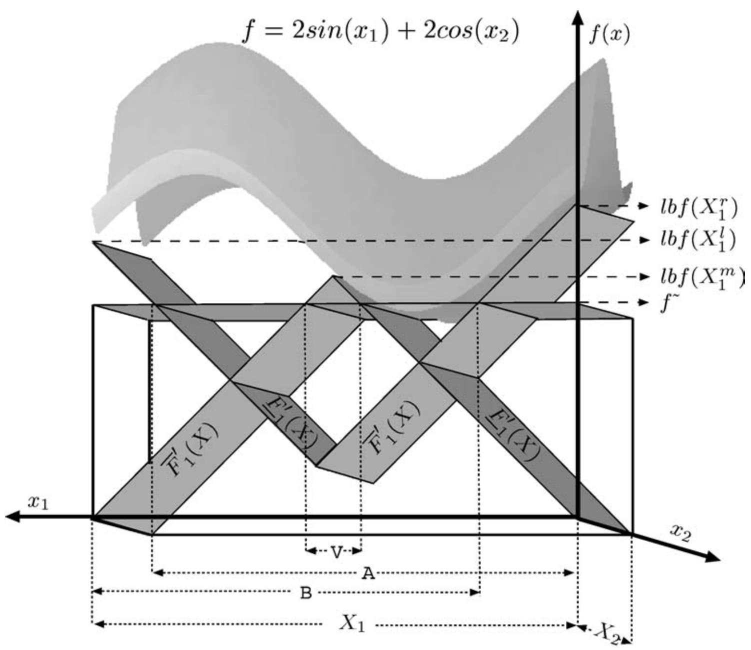

Figure 1. Support function using $F_1'(X)$, $lbf(X_1^l)$, $lbf(X_1^m)$ and $lbf(X_1^r)$. F' denotes the slope of the planes.

Then, only the interval

$$X_{go}^b = \left\{ x \in X\colon\ x_i \in \left[\underline{x}_i - \frac{lbf(X_i^l) - \overline{\tilde{f}}}{\underline{G}_i}, \bar{x}_i - \frac{lbf(X_i^r) - \overline{\tilde{f}}}{\overline{G}_i} \right] \right\} \tag{3}$$

can contain global minimizers and a lower bound $z(X)$ of $f(X)$ can be calculated as follows:

$$z(X) = \max_{j \in \{1,\dots,n\}} \left\{ \frac{lbf(X_j^l) \cdot \overline{G}_j - lbf(X_j^r) \cdot \underline{G}_j}{w(G_j)} + w(X) \cdot \frac{\underline{G}_j \cdot \overline{G}_j}{w(G_j)} \right\}. \tag{4}$$

Proof. Applying theorem 1 for $X(i : c_i) = X_i^l$ and for $X(i : c_i) = X_i^r$, subintervals $A = X_{go}^a(X_i^l)$ and $B = X_{go}^a(X_i^r)$ of X are obtained and the interval $X_{go}^b = A \cap B$ (see figure 1).

Proof of equation (4) can be obtained considering that $X \subseteq Y$, $F'(X) \subseteq F'(Y) = G$ and applying the mean-value theorem (see [1]). $\square$

Corollary 1. If for an interval X the inequality $z(X) > \overline{\tilde{f}}$ is fulfilled then it can be derived that X does not contain any global minimizer.

Let us now return to the problem (1). We can use the information stated in (2) during the global search. Thus, using $\underline{F}(X)$ together with the value of $z(X)$ from (4), and the centered form $F_c(X, m(X))$, we can build a new lower bound $lbzf(X)$ for $f(X)$ in the following way

$$lbzf(X) = \max\{\underline{F}(X), z(X), \underline{F_c}(X, m(X))\}. \tag{5}$$

The essence of the algorithm is that for any interval W obtained from interval X according to (3) ($W = X^b_{go}$), the current value of $\underline{\tilde{f}}$ is a lower bound of f at W_i^l and W_i^r; i.e. $\underline{\tilde{f}} \leqslant f(W_i^l)$ and $\underline{\tilde{f}} \leqslant f(W_i^r)$, so $lbf(W_i^l) = lbf(W_i^r) = \underline{\tilde{f}}$ are easily available bounds.

Additionally, a lower bound of $f(X(i : c_i))$, with $c = (c_1, c_2, \ldots, c_n) \in X$ can be obtained applying the centered form, i.e.

$$lbf(X(i : c_i)) = \underline{F_c}(X(i : c_i), c). \tag{6}$$

Moreover, if we have previously evaluated $F(c)$ and $F'(X)$, the value of $F_c(X(i : c_i), c)$ can be obtained without additional inclusion function evaluations.

The MIAG algorithm follows the theoretical results presented above and improves MTIAM algorithm in the following way: by using $lbzf(X)$ instead of $\underline{F}(X)$, which improves the selection rule, bounding rule, midpoint test and cutoff test; by adding the GradientTest elimination rule, based on theorem 2 and described in algorithm 3.

To obtain a better performance of GradientTest, the kth-coordinate of X with $\max_i\{\underline{F_c}(X_i^m, m(X))\}$ value (line 7, algorithm 2) is bisected, generating the subboxes W^1 and W^2 (lines 8 and 9, algorithm 2). The values of $sp(W^1)$ and $sp(W^2)$ are established in lines 10 to 13 of algorithm 2 to use the GradientTest, applied to the kth-coordinate of the generated subboxes.

The MIAG algorithm, as the IGO algorithm, uses a work list (L) and a final list (Q). In the MIAG algorithm L and Q store the following elements: $ListNode(X) = (X, lbzf(X), sp(X), F(m(X)), F'(X))$. The first element of a list can be extracted by the function PopHead. The MonotonicityTest and CutOffTest evaluate the eliminating rules described in section 1.

3. Numerical results and conclusions

The new algorithm MIAG has been numerically compared with the method MTIAM on a set of forty n-dimensional test functions. This set of test functions is listed in table 1 and has been taken from [2,6,12–14]. The reference where every function is described (Ref) and the dimension of the function (n) are shown for all the functions.

Table 1 also shows numerical comparison between MTIAM and MIAG. If FE presents the number of interval function evaluations, i.e., the number of $F(X)$ evaluations plus the number of interval point evaluations $F(x)$, and GE shows the number of interval function evaluations of the gradient $F'(X)$, then columns Eff_1 and Eff_2 represent $FE + n \cdot GE$ for algorithms MTIAM and MIAG, respectively. Column $SpUp$ shows the values of $SpUp = Eff_1/Eff_2$, providing information on the relative speedup of the MIAG algorithm compared to the MTIAM algorithm.

Algorithm 2. MIAG algorithm.
Funct MIAG(S, F, ε)

$sp(S) = (\{\underline{F}(S_1^l), \underline{F}(S_1^r)\}, \ldots, \{\underline{F}(S_n^l), \underline{F}(S_n^r)\})$
Eval $F(S)$, $\tilde{f} = F(m(S))$, $F'(S)$, and $lbzf(S)$
$\mathrm{L} = \{ListNode(S)\}$
$\mathrm{Q} = \{\emptyset\}$ *Initiation of the final list*
while ($L \neq \emptyset$)
  $ListNode(X) = \mathrm{PopHead(L)}$ *Extract the first element of L*
  $\underline{F_c}(X_k^m, m(X)) = \max_i\{\underline{F_c}(X_i^m, m(X))\}$
  $W^1 = X$; $W_k^1 = [\underline{x}_k, m(X_k)]$
  $W^2 = X$; $W_k^2 = [m(X_k), \bar{x}_k]$
  $sp(W^1) = sp(X)$
  $sp(W_k^1) = \{sp(X_k^l), \underline{F_c}(X_k^m, m(X))\}$
  $sp(W^2) = sp(X)$
  $sp(W_k^2) = \{\underline{F_c}(X_k^m, m(X)), sp(X_k^r)\}$
  for ($i = 1, 2$)
    GradientTest (W_k^i, $F_k'(X)$, $sp(W_k^i)$, $\tilde{f}$)
    if ($w(W^i) > 0$)
      Eval $F(W^i)$, $F'(W^i)$
      if (MonotonocityTest($F'(W^i)$) **continue**
      if ($\overline{F}(m(W^i)) < \overline{\tilde{f}}$) $\tilde{f} = F(m(W^i))$; CutoffTest(L,Q)
      if ($lbzf(W^i) \leqslant \overline{\tilde{f}}$)
        if ($w(W^i) \leqslant \varepsilon$) Save $ListNode(W^i)$ in Q
        else Save $ListNode(W^i)$ in L
return Q

Algorithm 3. GradientTest algorithm.
Funct GradientTest (X, $F'(X)$, $sp(X)$, $\tilde{f}$)

if ($sp(X^l) > \overline{\tilde{f}}$)
  $Y = [\underline{x} - (sp(X^l) - \overline{\tilde{f}})/\underline{F}'(X),\ \infty)$
  $sp(X^l) = \underline{\tilde{f}}$; $X = Y \cap X$
if ($w(X) > 0$ **and** $sp(X^r) > \overline{\tilde{f}}$)
  $Y = (-\infty,\ \bar{x} - (sp(X^r) - \overline{\tilde{f}})/\overline{F}'(X)]$
  $sp(X^r) = \underline{\tilde{f}}$; $X = Y \cap X$
return X, $sp(X)$

We have tested both algorithms with stopping criterion $w(X) \leqslant \varepsilon = 10^{-8}$ and a limit for the run-time equal to one hour. For most of the functions both algorithms ended the execution, but for a set of ten functions MTIAM was not able to finish. For these functions we show the values of ε for which both algorithms have finished execution, however MIAG was able to provide a solution with higher precision.

Table 1
Results of numerical comparison between MTIAM and MIAG.

Name	Ref	n	Eff_1	Eff_2	$SpUp$	ε
Schwefel 3.1	[12]	3	874	994	0.9	10^{-8}
Price	[4]	2	5322	5951	0.9	10^{-8}
Levy 5	[12]	2	1587	1598	1.0	10^{-8}
Shekel 10	[12]	4	1365	1374	1.0	10^{-8}
Schwefel 3.7	[14]	2	1762	1772	1.0	10^{-8}
Levy 8	[12]	3	851	857	1.0	10^{-8}
Shekel 5	[12]	4	1339	1348	1.0	10^{-8}
Schwefel 2.1 (Beale)	[14]	2	5560	5523	1.0	10^{-8}
Shekel 7	[12]	4	1365	1350	1.0	10^{-8}
Levy 3	[12]	2	7116	6979	1.0	10^{-8}
Rastrigin	[13]	2	1564	1496	1.0	10^{-8}
Schwefel 2.5 (Booth)	[14]	2	488	466	1.0	10^{-8}
Henriksen-Madsen 3	[6]	2	12204	11575	1.1	10^{-8}
Henriksen-Madsen 4	[6]	3	63693	59870	1.1	10^{-8}
Treccani	[3]	2	2430	2227	1.1	10^{-8}
EX1	[2]	2	488	443	1.1	10^{-8}
Branin	[12]	2	4869	4367	1.1	10^{-8}
Chichinadze	[4]	2	653	576	1.1	10^{-8}
Griewank 2	[13]	2	1952	1642	1.2	10^{-8}
Schwefel 1.2	[14]	4	27975	22963	1.2	10^{-8}
Schwefel 3.2	[14]	3	3170	2484	1.3	10^{-8}
Rosenbrock 2	[3]	2	1279	887	1.4	10^{-8}
Ratz 4	[12]	2	7096	4772	1.5	10^{-8}
Hartman 6	[12]	6	20996	13020	1.6	10^{-8}
Three-Hump-Camel-Back	[3]	2	3990	2138	1.9	10^{-8}
Hartman 3	[12]	3	4463	2046	2.2	10^{-8}
Schwefel 2.18 (Matyas)	[14]	2	10944	4812	2.3	10^{-8}
Six-Hump-Camel-Back	[13]	2	6824	2638	2.6	10^{-8}
Simplified Rosenbrock	[3]	2	2386	831	2.9	10^{-8}
Goldstein-Price	[12]	2	320969	30128	10.7	10^{-8}
Schwefel 2.14 (Powell)	[12]	4	387176	595993	0.6	10^{-5}
Schwefel 3.1p	[12]	3	7854	4131	1.9	10^{-3}
Ratz 5	[12]	3	917495	331049	2.8	10^{-3}
Ratz 6	[12]	5	2162657	468513	4.6	10^{-3}
Griewank 10	[13]	10	3875828	3869704	1.0	10^{-2}
Schwefel 2.10 (Kowalik)	[14]	4	673544	496155	1.4	10^{-2}
Rosenbrock 10	[3]	10	2708380	2045727	1.3	10^{-2}
EX2	[2]	5	1016177	256975	4.0	10^{-2}
Ratz 7	[12]	7	245691	50229	4.9	10^{-2}
Ratz 8	[12]	9	388997	71367	5.5	10^{-2}
Av. val.					1.93	

It can be seen from table 1 that the value of *SpUp* is less than one only for three out of 40 functions, and that in average (see the last row of table 1) MIAG is 1.93 times faster than MTIAM (*SpUp* ranges at the interval [0.6, 10.7]).

As the numerical results show, the MIAG algorithm has improved the MTIAM method with the refined use of the gradient information. The improvement is reached by pruning the searching region exploiting the already given information. For problems where the monotonicity test does not bring improvements, the gradient cannot help, thus the pruneable region is negligible and it can affect the convergence of the algorithm due to the different shape of the generated boxes. On the other hand for problems where the gradient is shown to be informative enough (e.g., Goldstein-Price) the improvement is surprisingly good.

References

[1] L.G. Casado, I. García, J.A. Martínez and Ya.D. Sergeyev, New interval analysis support functions using gradient information in a global minimization algorithm, J. Global Optimization 25(4) (2003) 1–18.

[2] T. Csendes and D. Ratz, Subdivision direction selection in interval methods for global optimization, SIAM J. Numer. Anal. 34 (1997) 922–938.

[3] L.W.C. Dixon and G.P. Szego, eds., *Towards Global Optimization* (North-Holland, Amsterdam, 1975).

[4] L.W.C. Dixon and G.P. Szego, eds., *Towards Global Optimization 2* (North-Holland, Amsterdam, 1978).

[5] R. Hammer, M. Hocks, U. Kulisch and D. Ratz, *C++ Toolbox for Verified Computing I: Basic Numerical Problems: Theory, Algorithms, and Programs* (Springer, Berlin, 1995).

[6] T. Henriksen and K. Madsen: Use of a depth-first strategy in parallel Global Optimization, Technical Report 92-10, Institute for Numerical Analysis, Technical University of Denmark (1992).

[7] R.B. Kearfott, *Rigorous Global Search: Continuous Problems* (Kluwer Academic, Dordrecht, 1996).

[8] R. Moore, *Interval Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1966).

[9] A. Neumaier, *Interval Methods for Systems of Equations* (Cambridge Univ. Press, Cambridge, 1990).

[10] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization* (Ellis Horwood, Chichester, 1988).

[11] D. Ratz, *Automatic Slope Computation and its Application in Nonsmooth Global Optimization* (Shaker Verlag, Aachen, 1998).

[12] 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.

[13] A. Törn and A. Žilinskas, *Global Optimization*, Lecture Notes in Computer Science, Vol. 350 (Springer, Berlin, 1989).

[14] G. Walster, E. Hansen and S. Sengupta, Test results for global optimization algorithm, in: *SIAM Numerical Optimization 1984* (SIAM, Philadelphia, PA) pp. 272–287.