Optimization Letters manuscript No.
(will be inserted by the editor)

A univariate global search working with a set of Lipschitz constants for the first derivative

Dmitri E. Kvasov · **Yaroslav D. Sergeyev**

Abstract In the paper, a global optimization problem is considered where the objective function $f(x)$ is univariate, black-box, and its first derivative $f'(x)$ satisfies the Lipschitz condition with an unknown Lipschitz constant K. In the literature, there exist methods solving this problem by using an a priori given estimate of K, its adaptive estimates, and adaptive estimates of local Lipschitz constants. Algorithms working with a number of Lipschitz constants for $f'(x)$ chosen from a set of possible values are not known in spite of the fact that a method working in this way with Lipschitz objective functions, DIRECT, has been proposed in 1993. A new geometric method evolving its ideas to the case of the objective function having a Lipschitz derivative is introduced and studied in this paper. Numerical experiments executed on a number of test functions show that the usage of derivatives allows one to obtain, as it is expected, an acceleration in comparison with the DIRECT algorithm.

Keywords Global optimization, Lipschitz derivatives, a set of Lipschitz constants, geometric algorithms

Mathematics Subject Classification (2000) 65K05, 90C26, 90C56

1 Introduction

Let us consider the following problem

$$f^* = f(x^*) = \min\ f(x), \quad x \in D = [a,b], \tag{1}$$

$$|f'(x_1) - f'(x_2)| \le K|x_1 - x_2|, \qquad x_1, x_2 \in D. \tag{2}$$

It is supposed that the objective function $f(x)$ can be black-box, multiextremal, its first derivative $f'(x)$, $x \in [a,b]$, can be calculated during the search, and $f'(x)$ is Lipschitz-continuous with some fixed, but unknown, constant K, $0 < K < \infty$, over $[a,b]$. This

This research was supported by the RFBR grant 07-01-00467-a and the grant 4694.2008.9 for supporting the leading research groups awarded by the President of the Russian Federation.

Dmitri E. Kvasov and Yaroslav D. Sergeyev *(corresponding author)*
DEIS – University of Calabria, Via P. Bucci 42C, 87036 – Rende (CS), Italy
Software Department, N.I. Lobatchevsky State University, Nizhni Novgorod, Russia
E-mail: kvadim@si.deis.unical.it (D.E. Kvasov); yaro@si.deis.unical.it (Ya.D. Sergeyev)

kind of problems is often encountered in engineering applications (see, e.g., [25, 30, 33]), particularly, in electrical engineering optimization problems (see, e.g., [6, 7, 28, 33]).

In the literature, there exist several methods for solving this problem that can be distinguished, for instance, with respect to the way the Lipschitz constant K is estimated in their computational schemes. There exist algorithms using an a priori given estimate of K (see, e.g., [3, 31]), its adaptive estimates (see, e.g., [14, 31]), and adaptive estimates of local Lipschitz constants (see, e.g., [30,31]). Algorithms working with a number of Lipschitz constants for $f'(x)$ chosen from a set of possible values varying from zero to infinity were not known so far in spite of the fact that a method working in this way with Lipschitz objective functions, DIRECT, has been proposed in [20] in 1993.

The present paper meets this lack and solves this more than a decade standing problem. A new geometric method for solving problem (1), (2) and working with a number of Lipschitz constants for $f'(x)$ is introduced and studied in this paper. At each iteration k of this algorithm, a new partition D^k of the search interval $D = [a, b]$ into subintervals is performed and a new set of trial points (i.e., points at which both the objective function $f(x)$ and its first derivative $f'(x)$ are evaluated) is generated. To perform this operation, a set of possible estimates of the Lipschitz constant from (2) is used instead of only one estimate K. The procedure of the selection of subintervals for producing new trials is based on estimates of the lower bounds of the objective function over the subintervals. These estimates are calculated taking into account all possible values of the Lipschitz constant K from zero to infinity. In terms of the geometric algorithms (i.e., algorithms that use in their work auxiliary functions to estimate the behavior of $f(x)$ over the search region, see, e.g., [18, 19, 30, 33]), it is possible to say that all admissible minorant functions are examined during the current iteration of the algorithm without constructing a specific one.

In the algorithm, a special attention is also paid to the improvement of the current minimal function value (the so-called record value) and a basic version of this improvement is incorporated in order to provide a faster convergence to a global minimizer. Numerical experiments executed on a number of test functions taken from the literature (see [18, 26]) show that the usage of derivatives allows one to obtain, as it is expected, an acceleration in comparison with the DIRECT method.

To conclude the introduction, we would like to emphasize the practical importance of global optimization methods using all possible Lipschitz constants in their work (see, e.g., [1, 2, 4, 5, 15, 17, 22–24, 35, 36]). In spite of the fact that they have the so-called everywhere dense convergence (i.e., convergence of an infinite sequence of trial points to any point of the search domain, see, e.g., [30, 32, 33]), the truncated (by some termination criterion) finite sequences of trial points generated by these methods are rapidly concentrated around the global minimizers (see, e.g., [8,9,17,20,21,29]). Therefore, these algorithms are widely used, for example, in situations where only a certain, a priori fixed, number of trials can be executed (see, e.g., [25, 30, 33]). Note that if the Lipschitz constant (or its valid estimate) of the objective function $f(x)$ or the first derivative $f'(x)$ can be used by a method, a more economic convergence of its trial sequence to the only points of global minimum of $f(x)$ can be guaranteed (see, e.g., various methods from [19, 25, 30, 33]).

2 Theoretical background

In this section, the main theoretical results, necessary for introducing the new algorithm, are obtained. First, a procedure for estimation of lower bounds of the objective function over subintervals is described. The second part is dedicated to the introduction of a procedure for determining nondominated subintervals, i.e., subintervals having the smallest lower bound for some particular estimate of the Lipschitz constant for $f'(x)$. These subintervals are candidates for partitioning at each iteration of the new method. Finally, a selection and partition strategy adopted by the algorithm for dividing subintervals adaptively is presented.

2.1 Lower bounding

Let us consider an iteration $k \geq 1$ of the new algorithm and a partition $\{D^k\}$ of the search interval $D = [a, b]$ into subintervals $[a_i, b_i]$, $1 \leq i \leq M(k)$, over which both the function and its first derivative are evaluated at a trial point $x^{j(k)} = c_i$, $1 \leq j \leq T(k)$ (hereafter $M(k)$ is the current number of subintervals and $T(k)$ is the current number of trial points). The point c_i can be one of the end-points of the subinterval $[a_i, b_i]$ from $\{D^k\}$, $1 \leq i \leq M(k)$, i.e., either $c_i = a_i$ or $c_i = b_i$. Let a positive value $\tilde{K}$ be chosen as an estimate of the Lipschitz constant K from (2), $\tilde{K} \geq K$. Given the estimate $\tilde{K}$, a lower bound R_i of the function values over the subinterval $[a_i, b_i]$ can be calculated as follows

$$R_i = f(c_i) \pm f'(c_i)(b_i - a_i) - 0.5\tilde{K}(b_i - a_i)^2, \tag{3}$$

where the sign '$-$' is used in the case of the right-end function evaluation, i.e., $c_i = b_i$ (see Fig. 1(a)), and the sign '$+$' is used in the case of the left-end function evaluation, i.e., $c_i = a_i$ (see Fig. 1(b)).

2.2 Nondominated subintervals

By using the obtained lower bounds of $f(x)$, the relation of domination can be established between every two subintervals of a current partition $\{D^k\}$ of D and a set of nondominated subintervals can be identified for a possible subdivision at the next iteration of the algorithm.

Definition 1 Given an estimate $\tilde{K} > 0$ of the Lipschitz constant K from (2), a subinterval $D_i = [a_i, b_i]$ *dominates* a subinterval $D_j = [a_j, b_j]$ with respect to $\tilde{K}$ if

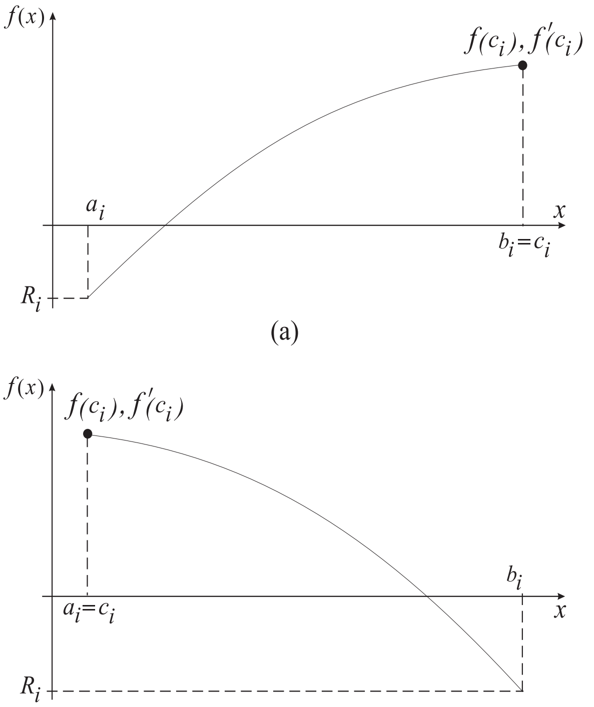
$$R_i(\tilde{K}) < R_j(\tilde{K}).$$

A subinterval $D_t = [a_t, b_t]$ is said to be *nondominated with respect to* $\tilde{K} > 0$ if for the chosen value $\tilde{K}$ there is no other subinterval in $\{D^k\}$ which dominates D_t.

Let us present the key result showing that each subinterval $D_i = [a_i, b_i] \in \{D^k\}$ and the respective lower bound R_i can be represented in a two-dimensional diagram (see Fig. 2, cf. [20, 21, 29]). The horizontal coordinate d_i and the vertical coordinate F_i of a dot, corresponding to D_i, are defined as follows

$$\begin{aligned} d_i &= 0.5(b_i - a_i)^2, \\ F_i &= f(c_i) \pm f'(c_i)(b_i - a_i), \end{aligned} \tag{4}$$

where in the last formula the sign '$-$' is used in the case of the right-end function evaluation, i.e., $c_i = b_i$ (see Fig. 1(a)), and the sign '$+$' is used in the case of the left-end function evaluation, i.e., $c_i = a_i$ (see Fig. 1(b)). Note that a point (d_i, F_i) in the diagram can correspond to several subintervals with the same length and the same value F_i.



(a)

(b)

Fig. 1 Finding the lower bound R_i from (3) over an interval $[a_i, b_i]$ in the following two cases: (a) right-end function evaluation, $c_i = b_i$; (b) left-end function evaluation, $c_i = a_i$

For example, let us consider a subinterval $D_A = [a_A, b_A]$. It is represented by the dot A in Fig. 2. Assuming an estimate of the Lipschitz constant K for the first derivative equal to $\tilde{K}$, $\tilde{K} \geq K$, a lower bound of $f(x)$ over the subinterval D_A is given by the value $R_A(\tilde{K})$ from (3). This value is the vertical coordinate of the intersection point of the line passed through the point A with the slope $\tilde{K}$ and the vertical coordinate axis (see Fig. 2). In fact, as can be seen from (3), intersection of the line with the slope $\tilde{K}$ passed through any dot representing a subinterval in the diagram of Fig. 2 and the vertical coordinate axis gives us the lower bound (3) of $f(x)$ over the corresponding subinterval.

Note that the points on the vertical axis ($d_i = 0$) do not represent any subinterval. This axis is used to express such values as lower bounds, the current minimal value f_{min} of the function, etc. (see also the related discussion in [29]).

By using the given graphical representation, it is possible to determine whether a subinterval dominates (with respect to a given estimate of K) some other subinterval from a partition $\{D^k\}$ (e.g., in Fig. 2 the subinterval D_A dominates both subintervals D_B and D_C with respect to the estimate $\tilde{K}$, the subinterval D_C dominates D_B with respect to $\tilde{K}$).

If a higher estimate $\hat{K} > \tilde{K}$ of the Lipschitz constant K is considered (see Fig. 2), the subinterval D_A still dominates D_B with respect to $\hat{K}$, since $R_A(\hat{K}) < R_B(\hat{K})$. But D_A in its turn is dominated by the subinterval D_C with respect to $\hat{K}$, because $R_A(\hat{K}) > R_C(\hat{K})$ (see Fig. 2).

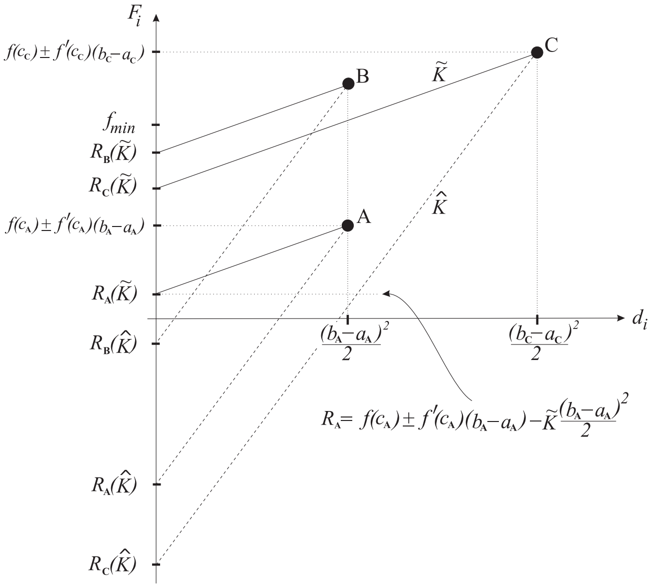

F_i
$f(c_C) \pm f'(c_C)(b_C - a_C)$
f_{min}
$R_B(\tilde{K})$
$R_C(\tilde{K})$
$f(c_A) \pm f'(c_A)(b_A - a_A)$
$R_A(\tilde{K})$
$R_B(\hat{K})$
$R_A(\hat{K})$
$R_C(\hat{K})$
C
B
A
$\tilde{K}$
$\hat{K}$
d_i
$\frac{(b_A - a_A)^2}{2}$
$\frac{(b_C - a_C)^2}{2}$
$R_A = f(c_A) \pm f'(c_A)(b_A - a_A) - \tilde{K}\frac{(b_A - a_A)^2}{2}$

Fig. 2 Graphical interpretation of lower bounds of $f(x)$ over subintervals

Thus, some subintervals (as D_B in Fig. 2) are always dominated by another ones, independently on the chosen estimate of the Lipschitz constant K. But there also exist subintervals (as D_A and D_C in Fig. 2) that are nondominated with respect to one estimate of the Lipschitz constant K and dominated with respect to another estimate of K. Since the exact Lipschitz constant K for $f'(x)$ (or its valid overestimate) is unknown, in the new algorithm it is proposed (following [20] and [29]) to use various estimates of the Lipschitz constant K from zero to infinity for lower bounding $f(x)$ over subintervals. The following definition should be given in this connection.

Definition 2 A subinterval $D_t \in \{D^k\}$ is called *nondominated* if there exists an estimate $0 < \tilde{K} < \infty$ of the Lipschitz constant K such that D_t is nondominated with respect to $\tilde{K}$.

In other words, nondominated subintervals are subintervals over which $f(x)$ has the smallest lower bound from (3) for some particular estimate of the Lipschitz constant for the first derivative $f'(x)$. For example, in Fig. 2 the subintervals D_A and D_C are nondominated.

It can be shown (see [29]) that nondominated subintervals (in the sense of Def. 2) are located on the lower-right convex hull of the set of dots representing the subintervals of the current partition of D. In Fig. 3 the subintervals represented by the dots F (the smallest subinterval), E, C, and A (the largest subinterval) are nondominated subintervals. The nondominated subintervals can be found by applying some algorithm for identifying the convex hull of the dots (e.g., Jarvis march, see [27] and also [16, 17]).

2.3 Selection and partition strategy

Let us finally present a selection and partition strategy adopted by the new algorithm for dividing subintervals in order to produce new trial points. Two types of subintervals are

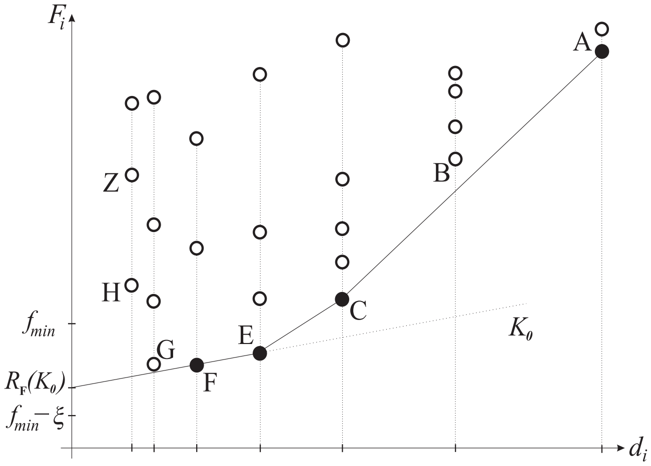


Fig. 3 Dominated subintervals are represented by white dots and nondominated subintervals are represented by black dots

selected for partitioning at each iteration k of the algorithm: a number of nondominated subintervals and a subinterval containing the record point $x_{min}(k)$, i.e., the point at which the current minimal value $f_{min}(k)$ is obtained.

Particularly, the first group of subintervals to be partitioned includes nondominated subintervals over which it is expected a significant improvement on the function values with respect to the current minimal value $f_{min}(k)$. Once a subinterval $[a_t, b_t]$ of the current partition D^k becomes nondominated, it can be subdivided only if the following additional condition is satisfied

$$R_t(\tilde{K}) \leq f_{min}(k) - \xi, \tag{5}$$

where the lower bound $R_t(\tilde{K})$ is calculated by formula (3). This condition prevents the algorithm from subdividing already well-explored small subintervals. For example, the subinterval D_F (dot F in Fig. 3) is nondominated but does not satisfy condition (5). In fact, this subinterval is nondominated with respect to estimates of the Lipschitz constant K smaller than or equal to K_0 (see Fig. 3) and, therefore, the lower bound of $f(x)$ over D_F is greater than the threshold ($f_{min} - \xi$).

A subinterval $D_{min}(k)$ containing the record point (called hereafter the record subinterval) forms by itself the second group of subintervals to be partitioned. Note that the record value $f_{min}(k)$ can be obtained at the end-points of different subintervals. Among them the record subinterval is that with the smallest lower bound of $f(x)$. It should be also highlighted that the record value is always greater than or equal to the vertical coordinate of the lowest dot (dots F and G in Fig. 3). The record subinterval itself can be represented in the two-dimensional diagram (d_i, F_i) by a dot which is not the lowest one (e.g., in Fig. 3 the dot Z represents the record subinterval D_{min}).

Subdivision of the record subinterval reflects the well-known fact in global optimization asserting the necessity of the local improvement of the record value (see, e.g., the references given in [18, 30]). The DIRECT algorithm [20] performs this operation automatically since the vertical coordinates of the points (as in formula (4)), representing subintervals on a diagram similar to that of Fig. 2, are given directly by the objective function values. In the new algorithm, generally, this is not the case and, thus, the record subinterval has to be

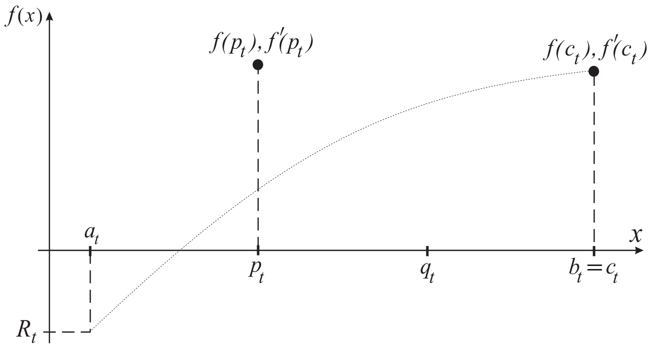


Fig. 4 Subdivision of a subinterval $[a_t, b_t]$ in the situation where $f(x)$ and $f'(x)$ are evaluated at the point p_t since they have been previously evaluated at the point b_t

specially subdivided (obviously, if it does not already belong to the group of nondominated subintervals satisfying condition (5)).

Note that the usage of the first derivative $f'(x)$ allows us to stop automatically the local record improvement. In fact, it is advisable to subdivide the record subinterval when the absolute value of the record first derivative is sufficiently high, i.e., when the following condition is satisfied

$$|f'(x_{min})| > \delta, \quad \delta \geq 0, \tag{6}$$

where the parameter δ is set by the user either in an absolute way or in a way related to the behavior of the objective function and/or its first derivative.

Let us now suppose that at the current iteration of the new algorithm a subinterval $D_t = [a_t, b_t]$, represented in the two-dimensional diagram of Fig. 2 by the dot (d_t, F_t) from (4), has been chosen for partitioning as described above. The subdivision of this subinterval is performed in such a way that three new equal subintervals of the length $(b_t - a_t)/3$ are created, i.e.,

$$\begin{gathered} [a_t, b_t] = [a_t, p_t] \cup [p_t, q_t] \cup [q_t, b_t], \\ p_t = a_t + (b_t - a_t)/3, \quad q_t = b_t - (b_t - a_t)/3. \end{gathered} \tag{7}$$

Obviously, the subinterval D_t is removed from the two-dimensional diagram (d_i, F_i), representing the current partition of the search interval, and the three subintervals generated are introduced into the diagram.

A new trial is performed either at the point p_t (if both the objective function $f(x)$ and its first derivative $f'(x)$ have been evaluated over subinterval $[a_t, b_t]$ at point b_t, see Fig. 4), or at the point q_t (if $f(x)$ and $f'(x)$ have been evaluated over subinterval $[a_t, b_t]$ at point a_t).

3 New Algorithm

In this section, the new algorithm for solving problem (1), (2) is described. First, the new method is presented and then its convergence properties are analyzed.

A formal scheme of the new algorithm can be given as follows.

Step 0 (*Initialization*). Set the iteration counter $k := 1$. Let the first evaluation of $f(x)$ and $f'(x)$ be performed at the central point of the initial interval $D = [a, b]$, i.e., $x^1 := c$, where $c = (a + b)/2$. Set the current partition of the search interval as

$D^1 := \{[a_1, b_1], [a_2, b_2]\}$, where $a_1 = a$, $b_2 = b$, $b_1 = a_2 = c$. Set the current number of subintervals $M(1) := 2$ and the current number of trials $T(1) := 1$. Set $f_{min}(1) := f(x^1)$, $x_{min}(1) := c$, and $D_{min}(1) := [a_1, b_1]$ ($D_{min}(1) := [a_2, b_2]$) if $F_1 \le F_2$ ($F_1 > F_2$), where F_1, F_2 are from (4).

Suppose now that $k \ge 1$ iterations of the algorithm have already been executed. The iteration $k + 1$ consists of the following steps.

Step 1 (*Nondominated Subintervals*). Identify both the set $\mathcal{S}$ of nondominated subintervals (accordingly to Def. 2) that satisfy condition (5), and the corresponding set $\mathcal{I}$ of their indices.

Step 2 (*Record Improvement*). If the record subinterval is such that $D_{min}(k) \notin \mathcal{S}$ and condition (6) is satisfied for the record point $x_{min}(k)$ then subdivide the record subinterval and produce a new trial point accordingly to subsection 2.3.
Increase both the current number of trial points $T := T + 1$ and the current number of subintervals $M := M + 2$. Update the current record f_{min} and the current record point x_{min}, if necessary.
Otherwise, go to Step 3.

Step 3 (*Subdivision of Nondominated Subintervals*). Perform the following Steps 3.1–3.3:

Step 3.1 (*Subinterval Selection*). Select a new subinterval $D_t = [a_t, b_t]$ from $\mathcal{S}$ such that

$$t = \arg \max_{j \in \mathcal{I}} \{b_j - a_j\}.$$

Step 3.2 (*Subdivision and Sampling*). Subdivide subinterval D_t and produce a new trial point accordingly to subsection 2.3.
Increase both the current number of trial points $T := T + 1$ and the current number of subintervals $M := M + 2$. Update the current record f_{min} and the current record point x_{min}, if necessary.

Step 3.3 (*Next Subinterval*). Eliminate the interval D_t from $\mathcal{S}$, i.e., set $\mathcal{S} := \mathcal{S} \backslash \{D_t\}$ and $\mathcal{I} := \mathcal{I} \backslash \{t\}$. If $\mathcal{S} \neq \varnothing$, then go to Step 3.1. Otherwise go to Step 4.

Step 4 (*End of the Current Iteration*). Increase the iteration counter $k := k + 1$. Go to Step 1 and start the next iteration.

The algorithm stops, for example, when the number of generated trial points reaches the maximal allowed number T_{max}. During the course of the algorithm, the satisfaction of this termination criterion is verified after every subdivision of a subinterval. The approximation of the global minimum value f^* and the global minimizer x^* is given by the current record value f_{min} and the current record point x_{min}.

Note that in the scheme of the method introduced, a basic and sufficiently simple technique for the record improvement is proposed (see Step 2). A more sophisticated approach, as for example the two-phase approach from [29], can also be adopted for this scope.

Moreover, either one of the points a or b of the search interval $[a, b]$, or both of them (producing in this way two initial trial points, i.e., $T(1) := 2$ in Step 0 of the scheme) can also be chosen as the starting points of the method. In this case, the algorithm determines exactly the global minimizer x^* if it belongs to one of the end-points of the search interval $[a, b]$.

Let us study convergence properties of the new algorithm during minimization of the function $f(x)$ from (1), (2) when the maximal allowed number of generated trial points T_{max} is infinitely high. In this case the algorithm does not stop (the number of iterations k goes to infinity) and an infinite sequence of trial points $\{x^{j(k)}\}$ is generated. The following theorem establishes the so-called everywhere dense convergence of the method, i.e., convergence of an infinite sequence of trial points to any point of the search domain. This result is in accordance with [32, 34], where it is shown that all deterministic algorithms which do not use the global information during their work can converge to the global optimum of a continuous function if and only if they produce an everywhere dense mesh of trial points.

Theorem 1 *For any point $x \in D$ and any $\sigma > 0$ there exist an iteration number $k(\sigma) \geq 1$ and a point $x' \in \{x^{j(k)}\}$, $k > k(\sigma)$, such that $|x - x'| < \sigma$.*

Proof Due to the subinterval partition scheme (see subsection 2.3), every subdivision of a subinterval generates three new subintervals with the length equal to a third of the length of the subdivided subinterval. Thus, fixed a positive value of σ, it is sufficient to prove that after a finite number of iterations $k(\sigma)$, the largest subinterval of the current partition of the search interval D will have the length smaller than σ. In such a case, in σ-neighborhood of any point of D there will exist at least one trial point generated by the algorithm.

To see this, recall that at each fixed iteration k' of the method, the two-dimensional diagram (d_i, F_i) is constantly updated and represents all subintervals of a partition $\{D^{k'}\}$ of the search interval D, including the record subinterval $D_{min}(k')$. This diagram groups all subintervals having the same length into a column of dots with the same horizontal coordinate d_i and orders them within the column d_i by the corresponding values $F_{i'}$ (see (4)).

Let us now consider the group d_{max} of the largest subintervals of the partition $\{D^{k'}\}$ which is always taken into account when nondominated subintervals are looked for. Particularly, a subinterval $D_t \in \{D^{k'}\}$ from this group, having the smallest value F_t, must be partitioned at the current iteration of the algorithm. In fact, there always exists a sufficiently large estimate K_∞ of the Lipschitz constant K for $f'(x)$ such that the subinterval D_t is a nondominated subinterval with respect to K_∞ and condition (5) is satisfied for the lower bound $R_t(K_\infty)$ (see Fig. 3). Three new subintervals generated during the subdivision of D_t (and having the length equal to a third of the length of D_t) are inserted into the group with a smaller horizontal coordinate $d_j < d_{max}$.

Since each group contains only finite number of subintervals, after a sufficiently large number of iterations $k > k'$ all subintervals of the group d_{max} will be subdivided. The group d_{max} will become empty and all largest subintervals will be positioned into the group d_j, $d_j < d_{max}$. The same procedure will be repeated with a new group of the largest subintervals.

To conclude the proof, it should be noted that the record subinterval D_{min} is itself represented by a dot in the diagram (d_i, F_i). It is subdivided either separately during the step of the record improvement (see Step 2), or as a nondominated subinterval (see Step 3). The latter will eventually happen when the record subinterval belongs to the group of largest subintervals.

Thus, there exists a finite number $k(\sigma)$ such that after performing $k(\sigma)$ iterations of the algorithm the largest subinterval of the current partition $\{D^{k(\sigma)}\}$ will have the main diagonal smaller than σ. ■

4 Numerical experiments

In this section, we present results of numerical experiments performed to compare the new algorithm with the DIRECT method [20], which has been developed for minimizing Lipschitz-continuous functions $f(x)$. The DIRECT method does not use the information about the first derivative $f'(x)$ but estimates the Lipschitz constant of the objective function $f(x)$ from a set of possible values from zero to infinity. The implementation of the

DIRECT method described in [10, 12] and downloadable from [11] has been used in the experiments performed.

Numerical comparison has been made with the DIRECT method because of the following three reasons. First, both the DIRECT method and the new one work with several Lipschitz constants simultaneously. Second, their comparison can show how the usage of derivatives can accelerate the search in the methods with this kind of estimating. Finally, methods using other kinds of estimates have their practical internal stopping rules related to the chosen estimate (or to another similar parameter) of the Lipschitz constant. The new algorithm and the DIRECT method do not have such a practical stopping rule (in fact, these methods usually are forced to stop when the maximal allowed number of trials T_{max} is reached), and, therefore, a comparison with other methods becomes less obvious.

To make the comparison, two series of experiments have been conducted. In the first series, the widely used set of 20 one-dimensional test functions from [18] has been considered (global minimizers of these functions have been defined more precisely with respect to [18], namely, with a tolerance at least 10^{-7}). In the second series, a class of 100 one-dimensional randomized test functions from [26] has been taken. Each function $f_s(x)$, $1 \le s \le 100$, of this class is defined over interval $[-5, 5]$ and has the following form (see [26])

$$f_s(x) = 0.025(x - x_s^*)^2 + \sin^2[(x - x_s^*) + (x - x_s^*)^2] + \sin^2(x - x_s^*), \tag{8}$$

where the global minimizer x_s^*, $1 \le s \le 100$, is chosen randomly (and differently for 100 function of the class) from interval $[-5, 5]$ by means of the random number generator used in the GKLS-generator of multidimensional test functions (see [13]; this generator is also downloadable for free from `http://wwwinfo.deis.unical.it/~yaro/GKLS.html`).

In order to compare the two methods, we stopped them (given a sufficiently high value $T_{max} = 5000$) when the global minimizer $x^* \in D$ was considered to be found, namely, when a trial point x' was generated in a small neighborhood of the global minimizer x^*

$$|x' - x^*| \le \varDelta(b - a), \tag{9}$$

where $[a, b]$ is the search interval from (1). Naturally, such a type of stopping criterion is acceptable only when the global minimizer x^* is known, i.e., for the case of test functions. In order to demonstrate the influence of changing the search accuracy $\varDelta$ from (9) on the convergence speed of the methods, three different values of $\varDelta$, namely, $\varDelta = 10^{-4}$, $\varDelta = 10^{-5}$, and $\varDelta = 10^{-6}$ were used in both series of experiments. Recall that for the new method each trial is more expensive with respect to a trial of the DIRECT method since it corresponds to the evaluations of both the objective function and its first derivative.

To execute numerical experiments, we need to define the parameter ξ from (5) of the algorithm, which can be set in different ways (see the related discussion in [29]). In order to make easier the numerical comparison with the DIRECT algorithm [20], a value ξ related to the current minimal function value $f_{min}(k)$ is fixed as it is done in the DIRECT method, i.e.,

$$\xi = \epsilon |f_{min}(k)|, \quad \epsilon \ge 0. \tag{10}$$

The recommended value of $\epsilon = 10^{-4}$ (see [20] and the references in [29]) was used in (10). The value of δ from (6) was set to 10^{-10}.

The obtained numbers of trial points generated by the algorithms while minimizing 20 functions from [18] are reported in Table 1, where in the last row the average values are given. It should be noted that the minimization of two functions (number 4 and 13) by the DIRECT method with the accuracy $\varDelta = 10^{-6}$ in (9) has required more than $T_{max} = 5000$

Table 1 Number of trial points generated by the new algorithm and the DIRECT algorithm for test functions from [18] with different values of the accuracy $\varDelta$ from (9)

$\mathcal{N}$	$\varDelta = 10^{-4}$		$\varDelta = 10^{-5}$		$\varDelta = 10^{-6}$	
	DIRECT	New	DIRECT	New	DIRECT	New
1	77	20	195	24	4977	36
2	26	12	347	26	1003	29
3	23	12	59	23	104	28
4	86	18	2096	28	5000*	35
5	34	17	201	26	1639	31
6	56	23	56	23	916	39
7	58	22	118	25	2978	37
8	76	30	105	37	418	40
9	53	21	281	28	804	31
10	66	21	142	24	1177	30
11	62	20	277	27	2195	31
12	95	23	269	27	269	27
13	20	10	2238	28	5000*	36
14	54	25	90	29	1271	37
15	29	17	29	17	545	43
16	38	19	496	31	496	31
17	51	40	87	66	414	84
18	72	45	116	67	178	89
19	76	21	556	29	1674	32
20	45	30	45	30	931	40
Average	54.85	22.30	390.15	30.75	> 1545.40	39.30

Table 2 Average number of trial points generated by the new algorithm and the DIRECT algorithm for 100 test functions (8) from [26] with different values of the accuracy $\varDelta$ from (9)

Method	$\varDelta = 10^{-4}$	$\varDelta = 10^{-5}$	$\varDelta = 10^{-6}$
DIRECT	40.11	68.02	98.39
New	22.34	29.37	37.22

trials (in Table 1 this fact is indicated by asterisks after the numbers of trial points). The impossibility of the DIRECT method to determine efficiently global minimizers of these functions is due to the flatness and largeness of their global minimizers basins (similar observations are also reported in [29]).

Table 2 represents the average numbers of trial points generated by the two methods while minimizing 100 randomized functions (8) from [26]. Note that – similarly to the results of Table 1 – the advantage of the new method in terms of the executed trials becomes more pronounced when the required search accuracy increases.

For the sake of illustration, in Fig. 5 and Fig. 6 functions from both test sets used in numerical experiments are represented. Particularly, Fig. 5 shows the graph of the function number 9 from the set of 20 functions [18] and the trial points (281 for the DIRECT method

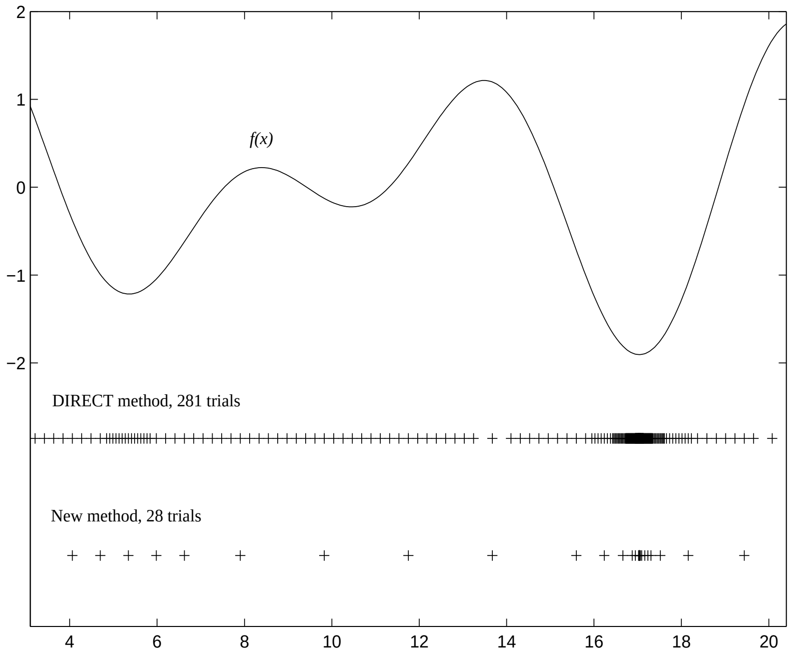


Fig. 5 Graph of the function number 9 from [18] and the trial points generated by the DIRECT method and the new one while minimizing this function with the accuracy $\Delta = 10^{-5}$ in (9)

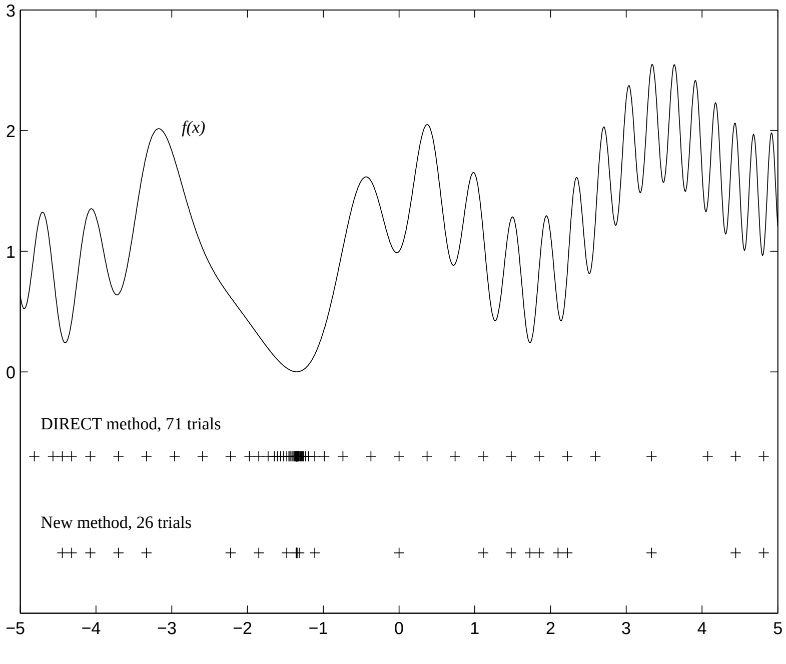


Fig. 6 Graph of a test function from [26] and the trial points generated by the DIRECT method and the new one while minimizing this function with the accuracy $\Delta = 10^{-5}$ in (9)

and 28 for the new method) generated by the two methods while minimizing this function with the accuracy $\Delta = 10^{-5}$ in (9). The single global minimum of the function, $f^* \approx -1.90596$, is attained at $x^* \approx 17.03919895$.

Fig. 6 shows the graph of the function number $s = 67$ from the set of randomly generated 100 functions (8) from [26] and the trial points (71 for the DIRECT method and 26 for the new method, which are close to the corresponding average numbers in Table 2) generated by the two methods while minimizing this function with the accuracy $\Delta = 10^{-5}$ in (9). The single global minimum of the function, $f^* = 0$, is attained at $x^* \approx -1.34952115$.

As demonstrated by the results of the performed numerical experiments, the usage of derivatives allows one to obtain a serious acceleration in comparison with the DIRECT method on the studied classes of test problems. Moreover, the more accurate global search is required, the more pronounced is the advantage of the new algorithm on the considered test functions.

References

1. C. A. Baker, L. T. Watson, B. Grossman, W. H. Mason, and R. T. Haftka. Parallel global aircraft configuration design space exploration. In M. Paprzycki, L. Tarricone, and L. T. Yang, editors, *Practical parallel computing*, volume 10 (4) of *Special Issue of the Inernational Journal of Computer Research*, pages 79–96. Nova Science Publishers, 2001.
2. M. C. Bartholomew-Biggs, S. C. Parkhurst, and S. P. Wilson. Using DIRECT to solve an aircraft routing problem. *Comput. Optim. Appl.*, 21(3):311–323, 2002.
3. L. Breiman and A. Cutler. A deterministic algorithm for global optimization. *Math. Program.*, 58(1–3):179–199, 1993.
4. R. G. Carter, J. M. Gablonsky, A. Patrick, C. T. Kelley, and O. J. Eslinger. Algorithms for noisy problems in gas transmission pipeline optimization. *Optim. Eng.*, 2(2):139–157, 2001.
5. S. E. Cox, R. T. Haftka, C. A. Baker, B. Grossman, W. H. Mason, and L. T. Watson. A comparison of global optimization methods for the design of a high-speed civil transport. *J. Global Optim.*, 21(4):415–433, 2001.
6. P. Daponte, D. Grimaldi, A. Molinaro, and Ya. D. Sergeyev. An algorithm for finding the zero-crossing of time signals with Lipschitzean derivatives. *Measurement*, 16(1):37–49, 1995.
7. P. Daponte, D. Grimaldi, A. Molinaro, and Ya. D. Sergeyev. Fast detection of the first zero-crossing in a measurement signal set. *Measurement*, 19(1):29–39, 1996.
8. D. E. Finkel and C. T. Kelley. Additive scaling and the DIRECT algorithm. *J. Global Optim.*, 36(4):597–608, 2006.
9. J. M. Gablonsky and C. T. Kelley. A locally-biased form of the DIRECT algorithm. *J. Global Optim.*, 21(1):27–37, 2001.
10. J. M. Gablonsky. An implemention of the DIRECT algorithm. Technical Report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University, Raleigh, N.C., USA, August 1998.
11. J. M. Gablonsky. DIRECT v2.04 FORTRAN code with documentation, 2001. Downloadable from: http://www4.ncsu.edu/∼ctk/SOFTWARE/DIRECTv204.tar.gz.
12. J. M. Gablonsky. *Modifications of the DIRECT algorithm*. PhD thesis, North Carolina State University, Raleigh, North Carolina, 2001.
13. M. Gaviano, D. Lera, D. E. Kvasov, and Ya. D. Sergeyev. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Trans. Math. Software*, 29(4):469–480, 2003.
14. V. P. Gergel. A global optimization algorithm for multivariate function with Lipschitzian first derivatives. *J. Global Optim.*, 10(3):257–281, 1997.
15. P. A. Graf, K. Kim, W. B. Jones, and L.-W. Wang. Surface passivation optimization using DIRECT. *J. Comput. Phys.*, 224(2):824–835, 2007.
16. J. He, A. Verstak, L. T. Watson, and M. Sosonkina. Design and implementation of a massively parallel version of DIRECT. *Comput. Optim. Appl.*, 40(2):217–245, 2008.
17. J. He, L. T. Watson, N. Ramakrishnan, C. A. Shaffer, A. Verstak, J. Jiang, K. Bae, and W. H. Tranter. Dynamic data structures for a direct search algorithm. *Comput. Optim. Appl.*, 23(1):5–25, 2002.

18. R. Horst and P. M. Pardalos, editors. *Handbook of Global Optimization*, volume 1. Kluwer Academic Publishers, Dordrecht, 1995.
19. R. Horst and H. Tuy. *Global Optimization – Deterministic Approaches*. Springer–Verlag, Berlin, 1996.
20. D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.*, 79(1):157–181, 1993.
21. D. R. Jones. The DIRECT global optimization algorithm. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, volume 1, pages 431–440. Kluwer Academic Publishers, Dordrecht, 2001.
22. K. Ljungberg, S. Holmgren, and Ö. Carlborg. Simultaneous search for multiple QTL using the global optimization algorithm DIRECT. *Bioinformatics*, 20(12):1887–1895, 2004.
23. C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Res.*, 13(11):2467–2474, 2003.
24. T. D. Panning, L. T. Watson, N. A. Allen, K. C. Chen, C. A. Shaffer, and J. J. Tyson. Deterministic parallel global parameter estimation for a model of the budding yeast cell cycle. *J. Global Optim.*, 40(4):719–738, 2008.
25. J. Pintér. *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)*. Kluwer Academic Publishers, Dordrecht, 1996.
26. J. Pintér. Global optimization: software, test problems, and applications. In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization*, volume 2, pages 515–569. Kluwer Academic Publishers, Dordrecht, 2002.
27. F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1993.
28. Ya. D. Sergeyev, P. Daponte, D. Grimaldi, and A. Molinaro. Two methods for solving optimization problems arising in electronic measurements and electrical engineering. *SIAM J. Optim.*, 10(1):1–21, 1999.
29. Ya. D. Sergeyev and D. E. Kvasov. Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM J. Optim.*, 16(3):910–937, 2006.
30. Ya. D. Sergeyev and D. E. Kvasov. *Diagonal Global Optimization Methods*. FizMatLit, Moscow, 2008. In Russian.
31. Ya. D. Sergeyev. Global one-dimensional optimization using smooth auxiliary functions. *Math. Program.*, 81(1):127–146, 1998.
32. C. P. Stephens and W. Baritompa. Global optimization requires global information. *J. Optim. Theory Appl.*, 96(3):575–588, 1998.
33. R. G. Strongin and Ya. D. Sergeyev. *Global optimization with non-convex constraints: Sequential and parallel algorithms*. Kluwer Academic Publishers, Dordrecht, 2000.
34. A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer–Verlag, Berlin, 1989.
35. L. T. Watson and C. Baker. A fully-distributed parallel global search algorithm. *Engineering Computations*, 18(1/2):155–169, 2001.
36. Hong Zhu and D. B. Bogy. Hard disc drive air bearing design: modified DIRECT algorithm and its application to slider air bearing surface optimization. *Tribology Internat.*, 37(2):193–201, 2004.