

# A deterministic global optimization using smooth diagonal auxiliary functions<sup>☆</sup>

Yaroslav D. Sergeyev<sup>a,b</sup>, Dmitri E. Kvasov<sup>a,b,\*</sup>

<sup>a</sup>*DIMES – University of Calabria, Via P. Bucci 42C, 87036 – Rende (CS), Italy*

<sup>b</sup>*Software Department, N.I. Lobachevsky State University, Nizhni Novgorod, Russia*

---

## Abstract

In many practical decision-making problems it happens that functions involved in optimization process are black-box with unknown analytical representations and hard to evaluate. In this paper, a global optimization problem is considered where both the goal function  $f(x)$  and its gradient  $f'(x)$  are black-box functions. It is supposed that  $f'(x)$  satisfies the Lipschitz condition over the search hyperinterval with an unknown Lipschitz constant  $K$ . A new deterministic ‘Divide-the-Best’ algorithm based on efficient diagonal partitions and smooth auxiliary functions is proposed in its basic version, its convergence conditions are studied and numerical experiments executed on eight hundred test functions are presented.

*Keywords:* Global optimization, deterministic methods, Lipschitz gradients, unknown Lipschitz constant

*2000 MSC:* 65K05, 90C26, 90C56

---

## 1. Introduction

In many important applied problems, some decisions should be made by finding the global optimum of a multiextremal objective function subject to a set of constraints (see, e. g., [9, 17, 18, 31, 32, 34, 36, 45, 46, 50, 51, 54] and the

---

<sup>☆</sup>This work was supported by the INdAM–GNCS 2014 Research Project of the Italian National Group for Scientific Computation of the National Institute for Advanced Mathematics “F. Severi”.

\*Corresponding author. Fax: +39 0984 494713.

*Email addresses:* [yaro@si.dimes.unical.it](mailto:yaro@si.dimes.unical.it) (Yaroslav D. Sergeyev),  
[kvadim@si.dimes.unical.it](mailto:kvadim@si.dimes.unical.it) (Dmitri E. Kvasov)

references given therein). Frequently, especially in engineering applications, the functions involved in optimization process are black-box with unknown analytical representations and hard to evaluate. Such computationally challenging decision-making problems often cannot be solved by traditional optimization techniques based on strong suppositions (such as, e. g., convexity) about the problem.

Because of significant computational costs involved, normally a limited number of functions evaluations are available for a decision-maker (physicist, biologist, economist, engineer, etc.) when he/she optimizes this kind of functions. Hence, the main goal is to construct fast global optimization algorithms that generate acceptable solutions with a relatively small number of functions evaluations.

Usually, the methods applied in this context are subdivided in stochastic and deterministic. Stochastic approaches (see, e. g., [2, 9, 18, 31, 32, 53, 54]) can often work in a simpler manner than the deterministic algorithms. They can be also suitable for the problems where the functions evaluations are corrupted by noise. However, solutions found by some stochastic algorithms (e. g., by popular heuristic methods like evolutionary algorithms, simulated annealing, etc.; see, e. g., [9, 11, 32, 38, 52]) can be only local solutions that can be located far away from the global one. This fact can preclude these methods from their usage in practice if an accurate estimate of the global solution is required. Therefore, our attention in the paper is focused on deterministic approaches.

Deterministic global optimization is an important applied field (see, e. g., [9, 18, 36, 45, 51, 55]). As a rule, deterministic methods exhibit (under certain conditions) rigorous global convergence properties and allow one to obtain guaranteed estimations of global solutions. However, deterministic models can still require too many function evaluations to achieve adequately good solutions for the considered global optimization problems.

One of the simplest techniques in this framework is represented by the so-called derivative-free (or direct) approach (see, e. g., [4, 5, 22, 30, 37]), often used for solving important applied problems (see, e. g., pattern search methods [22], the DIRECT method [20], the response surface, or surrogate model methods [21], etc.). Unfortunately (see, e. g., [6, 26, 35, 44]), these methods either are designed to detect only stationary points or can demand too high computational effort for their work. As observed, e. g., in [18, 49], if no particular assumptions are made on the objective function, any finite number of function evaluations cannot guarantee getting close to the global

minimum, since the function may have very deep and narrow peaks.

The Lipschitz-continuity assumption is one of the natural suppositions on the objective function (in fact, in technical systems the energy of change is always limited). The problem involving Lipschitz functions is said to be the Lipschitz global optimization problem (see, e. g., [7, 18, 27, 36, 45, 46, 51, 54]).

The global optimization problem with a differentiable objective function having the Lipschitz gradient (with an unknown Lipschitz constant) is an important class of Lipschitz global optimization problems. Formally, this class of problems can be stated as follows:

$$f^* = f(x^*) = \min_{x \in D} f(x), \quad (1)$$

$$\|f'(x') - f'(x'')\| \leq K \|x' - x''\|, \quad x', x'' \in D, \quad 0 < K < \infty, \quad (2)$$

where

$$D = [a, b] = \{x \in R^N : a(j) \leq x(j) \leq b(j)\}. \quad (3)$$

It is assumed here that the objective function  $f(x)$  can be black-box, multiextremal, its gradient  $f'(x) = \left( \frac{\partial f(x)}{\partial x(1)}, \frac{\partial f(x)}{\partial x(2)}, \dots, \frac{\partial f(x)}{\partial x(N)} \right)^T$  (which can be itself an expensive black-box vector-function) can be calculated during the search, and  $f'(x)$  is Lipschitz-continuous with some unknown constant  $K$ ,  $0 < K < \infty$ , over  $D$ . Problem (1)–(3) is frequently met in engineering (see, e. g., [36, 45, 51]), for instance, in electrical engineering design (see, e. g., [42, 51]).

There are known several methods for solving this problem that can be distinguished with respect to the way the Lipschitz constant  $K$  from (3) is estimated in their computational schemes. There exist algorithms using an a priori given estimate of  $K$  (see, e. g., [1, 39]), its adaptive estimates (see, e. g., [13, 14, 39]), and adaptive estimates of local Lipschitz constants (see, e. g., [39, 45]). Recently, methods working with multiple estimates of  $K$  chosen from a set of possible values have been also proposed (see [25, 27]).

This paper is devoted to developing a new global optimization algorithm for solving problem (1)–(3). The new method adaptively estimates the unknown Lipschitz constant  $K$  from (2) during the search and is based on efficient diagonal partitions of the search domain proposed recently. Theoretical background for the introduced method is drawn in Section 2. The algorithm is presented and analyzed theoretically in Section 3. Section 4 presents results of numerical experiments executed on several hundreds of multiextremal test functions. Finally, Section 5 concludes the paper.

## 2. Theoretical background

In this Section, some important theoretical results, necessary for introducing the new algorithm, are briefly described.

### 2.1. ‘Divide-the-Best’ algorithms

As known, many global optimization methods (of both stochastic and deterministic types) have a similar structure. Therefore, several approaches to the development of a general framework for describing global optimization algorithms and providing their convergence conditions in a unified manner have been proposed (see, e. g., [9, 16, 18, 36]). The ‘Divide-the-Best’ approach (see [40, 45]) is one of such unifying schemes. It generalizes both the schemes of adaptive partition [36] and characteristic [16, 45, 51] algorithms, which are widely used for constructing and studying global optimization methods.

In a ‘Divide-the-Best’ algorithm (its generic iteration is represented by the flow chart in Figure 1), given a vector  $q$  of the method parameters, an adaptive partition of the search domain  $D$  from (3) into a finite set  $\{D_i^k\}$  of robust subsets  $D_i^k$  is considered at every iteration  $k$ . In Step 1, the Lipschitz constant ( $K$  from (2), for the objective function gradient  $f'(x)$ ; or  $L$  in the case of the Lipschitz objective function  $f(x)$ ) is estimated in some way. Basing on the previously obtained information  $X^k, Z^k$  about the objective function, the ‘merit’ (called *characteristic*)  $R_i$  of each subset (see Step 2 in Figure 1) is estimated for performing a further, more detailed, investigation (see Steps 3 and 4 in Figure 1). The best (in a predefined sense) characteristic achieved over a hyperinterval  $D_i^k$  corresponds to a higher possibility to determine the global minimum point within  $D_i^k$  (see Step 3). The hyperinterval  $D_i^k$  is then partitioned at the next iteration of the algorithm. More than one ‘promising’ hyperintervals can be subdivided at each iteration.

Various techniques (for instance, in the context of the *geometric* approach, see, e. g., [18, 27, 46, 51]) for selection of hyperintervals to be partitioned (see Step 3 in Figure 1) and for executing this partitioning (by means of an operator  $P$ , see Step 4 in Figure 1 and the following subsection 2.2) can be used within this scheme.

As the stopping criteria, one can check, e. g., the volume of a hyperinterval with the best characteristic or depletion of computational resources such as the maximal number of *trials* (i. e., evaluations of the objective function and its gradient).

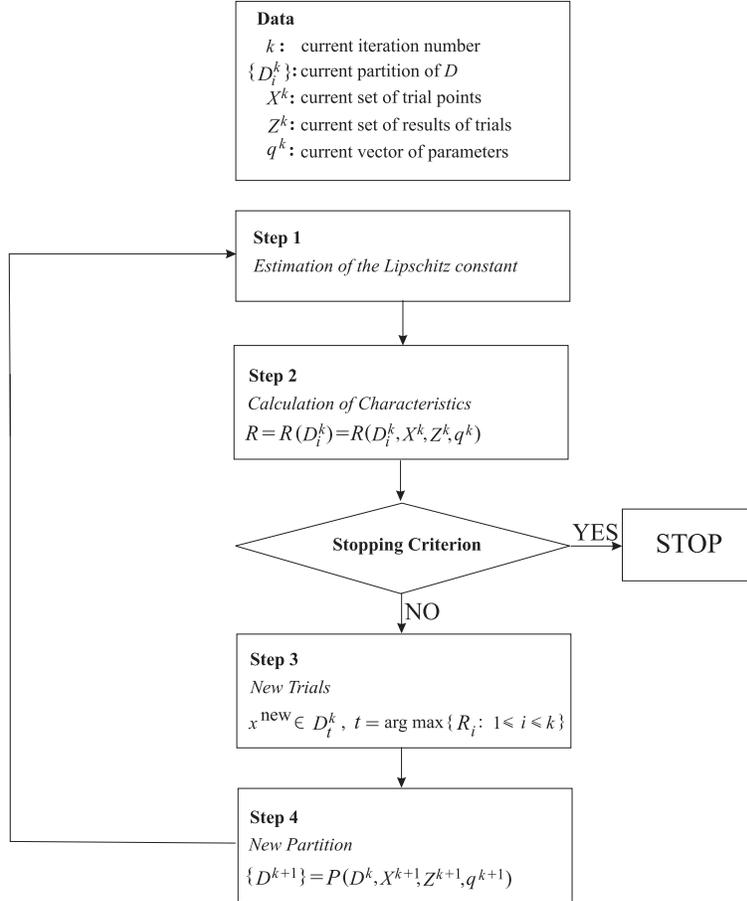


Figure 1: Flow chart of ‘Divide-the-Best’ algorithms.

Theoretical analysis of the ‘Divide-the-Best’ algorithms with different types of characteristics and partition operators is performed in [40, 45]. A particular attention is given there to situations (important in applied problems) when conditions of global (local) convergence are satisfied not in the whole domain  $D$ , but only in its small subregion(s). This can happen, e. g., in many Lipschitz global optimization algorithms that either underestimate the Lipschitz constant during their work or use local information in various subregions of  $D$  (see, e. g., [23, 40, 45, 51]). It should be also noticed that the ‘Divide-the-Best’ scheme can be successfully applied to develop parallel multidimensional global optimization methods (see, e. g., [16, 51]).

## 2.2. Efficient partition strategy

Regarding the partition strategies (see partitioning operator  $P$  on Step 4 in Figure 1), the *diagonal* partition strategies (see the references in [36, 44, 45, 46]) are taken into account in this paper, although other types of subdivisions are worth to be considered (see, e. g., simplicial partitions in [3, 33, 34]). In the diagonal approach, the hyperinterval  $D$  from (3) is subdivided into a set of smaller hyperintervals, the objective function (and its gradient) is (are) evaluated only at two vertices corresponding to the main diagonal of the generated hyperintervals (see, e. g., vertices  $a_i$  and  $b_i$  of a hyperinterval  $D_i$  in Figure 2), and the results of these trials are used to choose a hyperinterval for subsequent subdivisions. This approach has a number of interesting theoretical properties and has proved to be efficient in solving practical problems.

First, efficient one-dimensional global optimization algorithms can be easily extended to the multidimensional case by means of the diagonal approach (see, e. g., [44, 45, 46]). For instance, in order to obtain the characteristic  $R_i$  of a multidimensional hyperinterval  $D_i$ , some univariate characteristics can be used as prototypes. After an appropriate transformation they can be considered along the one-dimensional segment being the main diagonal  $[a_i, b_i]$  of the hyperinterval  $D_i$  (see smooth auxiliary function  $\phi_i(\lambda)$  in Figure 2).

Second, the diagonal approach is computationally close to one of the simplest strategies—center-sampling technique (see, e. g., [4, 7, 8, 10, 20])—but at the same time, the function trials are executed at two points of each hyperinterval, providing so more information about the function over the subregion under investigation than center-sampling algorithms that evaluate  $f(x)$  (and  $f'(x)$ ) only at one central point.

An important issue in the framework of diagonal algorithms is the way the hyperintervals are partitioned during the search. Various exploration techniques based on different diagonal partition strategies are studied, e. g., in [41, 45, 46]. It is shown that partition strategies traditionally used in diagonal methods can perform many redundant trials and, hence, do not fulfil the requirements of computational efficiency (this redundancy significantly slows down the search in the case of expensive functions).

A diagonal partition strategy introduced in [41, 45] allows one to avoid the computational redundancy of diagonal methods. In contrast to traditional diagonal schemes, the proposed strategy (called *non-redundant*, or *efficient* diagonal partition strategy) produces regular trial meshes in such a way that one vertex where  $f(x)$  and  $f'(x)$  are evaluated can belong to several hyperintervals (up to  $2^N$ ,  $N$  is the problem dimension from (3); see Figure 3

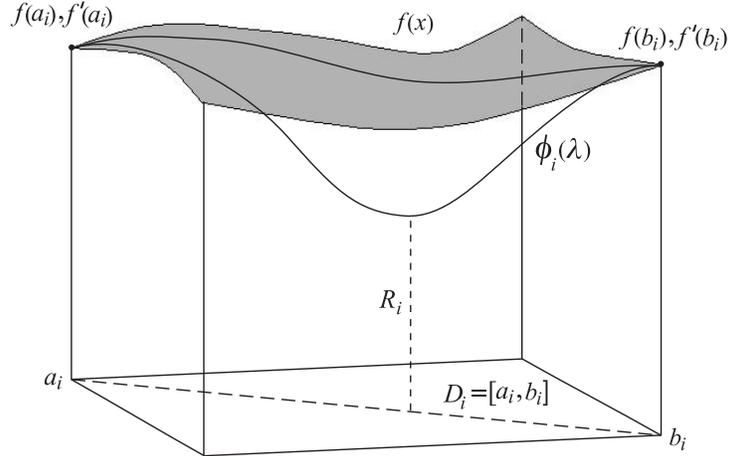


Figure 2: Obtaining the lower bound  $R_i$  for the objective function  $f(x)$  with the Lipschitz gradient over  $D_i = [a_i, b_i]$  by using smooth auxiliary function  $\phi_i(\lambda)$ ,  $\lambda \in [a_i, b_i]$ , along the main diagonal  $[a_i, b_i]$  of  $D_i$ .

that shows how partitions of gray-colored hyperintervals are executed in the course of iterations 1–6, where trial points are represented by black dots). So, the time-consuming operation of evaluating both  $f(x)$  and  $f'(x)$  is replaced by a significantly faster procedure of reading (up to  $2^N$  times) the previously obtained functions values, saved in a special database (see, e.g., [45]). In Figure 3, during the 6-th iteration the partition has been performed *for free* since the required values of  $f(x)$  and  $f'(x)$  have been already obtained at the 4-th and 5-th iterations (trial points of the 6-th iteration are in brackets, see Figure 3d). Hence, the non-redundant partition strategy speeds up the search and also leads to saving computer memory. It is important that this feature becomes more pronounced when the problem dimension  $N$  increases (see, e.g., [24, 44, 45]). A formal procedure of subdivisions by means of the efficient diagonal partition strategy can be viewed in the next Section (see formulae (12)–(18)).

A novel scheme for developing fast Lipschitz global optimization methods can be, thus, considered. It can rely on the non-redundant diagonal partition strategy allowing an efficient extension of well-studied one-dimensional Lipschitz global optimization algorithms to the multidimensional case. Interesting multidimensional diagonal methods, based on different ways for achieving the Lipschitz information and constructed in the framework of the efficient diagonal approach, have been proposed and their convergence prop-

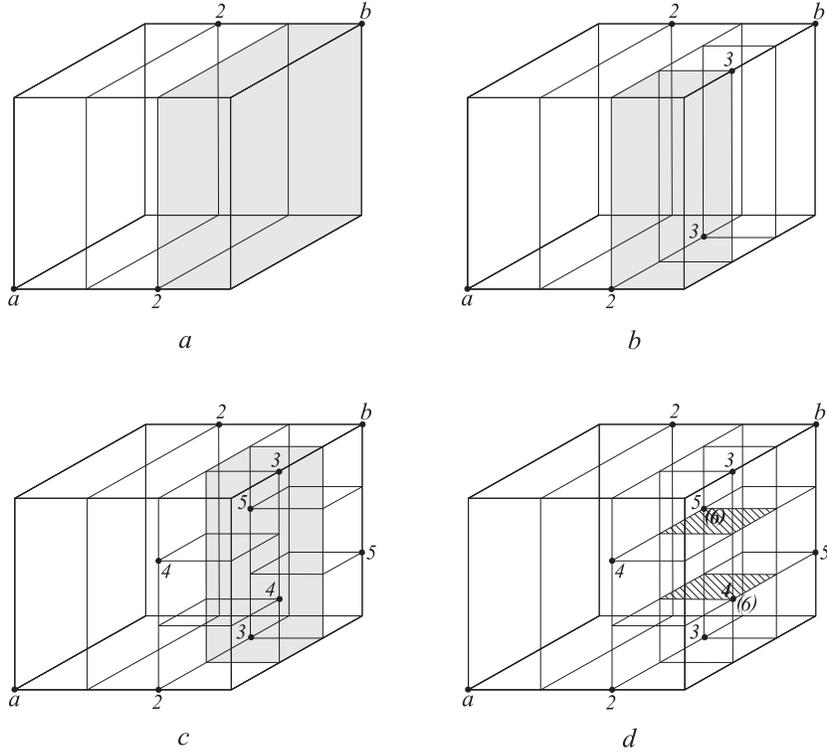


Figure 3: Example of partitioning a three-dimensional hyperinterval by the efficient diagonal partition strategy.

erties have been studied, e. g., in [24, 44, 45]. In the following Section, this scheme is used (see [45]) to develop a new global optimization algorithm for solving problem (1)–(3). This algorithm uses smooth diagonal auxiliary functions along the main diagonals of hyperintervals and shows how the unknown constant  $K$  from (2) can be adaptively estimated during the search.

### 3. New algorithm SMOOTHD

In this Section, a new algorithm SMOOTHD for solving problem (1)–(3) is described. It generalizes the one-dimensional method adaptively estimating the constant  $K$  (see [39]) to the multidimensional case by using the diagonal approach and the efficient partition strategy from [41, 45]. First, the new method is presented and its computational scheme is given, then its convergence properties are analyzed.

To describe the algorithm SMOOTHD formally, we need the following designations:

$k \geq 1$  – the current iteration number of the algorithm;

$p(k)$  – the total number of function trials executed during the previous  $k - 1$  iterations of the method;

$\{x^{p(k)}\}$  – the sequence of trial points generated by the algorithm in  $k$  iterations;

$\{z^{p(k)}\} = \{f(x^{p(k)})\}$  – the corresponding sequence of the objective function values;

$\{\frac{\partial z^{p(k)}}{\partial x(j)}\} = \{\frac{\partial f(x^{p(k)})}{\partial x(j)}\}$ ,  $1 \leq j \leq N$ , –  $N$  sequences of the corresponding partial derivatives  $f'_j(x)$ ,  $1 \leq j \leq N$ , evaluated at  $x^{p(k)}$ ;

$M = M(k)$  – the total number of hyperintervals of the current partition of the search domain at the beginning of the  $k$ -th iteration;

$\{D^k\} = \{D_i^k\} = \{D_i\}$ ,  $1 \leq i \leq M$ , – the current partition of the initial hyperinterval  $D$  into hyperintervals  $D_i = [a_i, b_i]$ .

For each hyperinterval  $D_i$  the coordinates of the vertices  $a_i$  and  $b_i$  of its main diagonal  $[a_i, b_i]$ , the corresponding values of  $f(a_i)$  and  $f(b_i)$ , and the corresponding values of partial derivatives  $f'_j(a_i)$  and  $f'_j(b_i)$ ,  $1 \leq j \leq N$ , are stored.

Directional derivatives along main diagonals are then evaluated as follows:

$$f'(a_i) = \left( \sum_{j=1}^N f'_j(a_i)(b_i(j) - a_i(j)) \right) / \Delta_i, \quad (4)$$

$$f'(b_i) = \left( \sum_{j=1}^N f'_j(b_i)(b_i(j) - a_i(j)) \right) / \Delta_i, \quad (5)$$

where

$$\Delta_i = \|a_i - b_i\| = \sqrt{\sum_{j=1}^N (a_i(j) - b_i(j))^2} \quad (6)$$

is the length of the main diagonal of hyperinterval  $D_i$ ,  $1 \leq i \leq M(k)$ .

**New Algorithm SMOOTHD.** Set the initial value of the iteration counter  $k := 1$ . Evaluate the objective function  $f(x)$  and its partial derivatives  $f'_j(x)$ ,  $1 \leq j \leq N$ , at the vertices  $a$  and  $b$  of the search domain  $D$ :  $x^1 = a$ ,  $x^2 = b$ . Set both the current number of trials  $p(1) := 2$  and the current number

of hyperintervals  $M(1) := 1$ . The initial partition of the search domain is  $\{D^1\} := \{D\}$ .

Suppose that  $k \geq 1$  iteration of the algorithm have been already executed. Iteration  $k + 1$  consists of the following Steps.

**Step 1** (*Estimation of the Lipschitz constant for the gradient*). Obtain the current estimate  $m$  of the Lipschitz constant  $K$  from (2) for the objective function gradient as

$$m = r \max\{\xi, \hat{m}\}, \quad (7)$$

where  $r > 1$  is the reliability parameter of the method,  $\xi$  is a small positive value (it ensures the correct algorithm execution when the value  $\hat{m}$  is too small; as a rule,  $\xi \leq 10^{-6}$ ), and  $\hat{m}$  is found as

$$\hat{m} = \max_{1 \leq i \leq M(k)} \{w_i\},$$

with

$$w_i = \frac{|2(f(a_i) - f(b_i)) + (f'(a_i) + f'(b_i))\Delta_i| + d_i}{\Delta_i^2}$$

and

$$d_i = \{[2(f(a_i) - f(b_i)) + (f'(a_i) + f'(b_i))\Delta_i]^2 + (f'(b_i) - f'(a_i))^2 \Delta_i^2\}^{\frac{1}{2}}.$$

**Step 2** (*Characteristics calculations*). For each hyperinterval  $D_i \in \{D^k\}$ ,  $1 \leq i \leq M(k)$ , calculate its characteristic  $R_i$ , being the minimum of the auxiliary function  $\phi_i(\lambda)$  on the diagonal  $[a_i, b_i]$  (see Fig. 2), as follows.

**Step 2.1.** If inequality

$$\pi'_i(y_i)\pi'_i(y'_i) < 0,$$

is verified, where  $\pi'_i(y_i)$ ,  $\pi'_i(y'_i)$  are found by formulae

$$\pi'_i(y_i) = my_i + B_i, \quad \pi'_i(y'_i) = my'_i + B_i,$$

with

$$y_i = \frac{\Delta_i}{4} + \frac{f'(b_i) - f'(a_i)}{4m} + \frac{f(a_i) - f(b_i) + f'(b_i)\Delta_i + 0.5m\Delta_i^2}{m\Delta_i + f'(b_i) - f'(a_i)},$$

$$y'_i = -\frac{\Delta_i}{4} - \frac{f'(b_i) - f'(a_i)}{4m} + \frac{f(a_i) - f(b_i) + f'(b_i)\Delta_i + 0.5m\Delta_i^2}{m\Delta_i + f'(b_i) - f'(a_i)},$$

$$B_i = f'(b_i) - 2my_i + m\Delta_i,$$

then go to Step 2.2. Otherwise, go to Step 2.3.

**Step 2.2.** Calculate characteristic  $R_i$  as

$$R_i = \min\{f(a_i), \phi_i(\hat{x}_i), f(b_i)\}, \quad (8)$$

where

$$\phi_i(\hat{x}_i) = f(b_i) - f'(b_i)\Delta_i - 0.5m\Delta_i^2 + my_i^2 - 0.5m\hat{x}_i^2$$

and

$$\hat{x}_i = 2y_i - m^{-1}f'(b_i) - \Delta_i.$$

Go to Step 3.

**Step 2.3.** Calculate characteristic  $R_i$  as

$$R_i = \min\{f(a_i), f(b_i)\}. \quad (9)$$

**Step 3** (*Hyperinterval selection*). Select a hyperinterval  $D_t \in \{D^k\}$  for the further partitioning such that condition

$$t = \arg \min_{1 \leq i \leq M(k)} R_i \quad (10)$$

is verified. If there are several hyperintervals satisfying condition (10), then choose among them the hyperinterval with the minimal index  $t$ .

**Step 4** (*Stopping criterion*). If

$$\|a_t - b_t\| \leq \varepsilon \|a - b\|, \quad (11)$$

where  $\varepsilon > 0$  is a given accuracy and  $\|\cdot\|$  is chosen in accordance with (2) and (6), then stop the algorithm.

Take as an estimate of the global minimum  $f^*$  value  $z_k^*$

$$z_k^* = \min\{z \mid z \in z^{p(k)}\},$$

attained at point

$$x_k^* = \arg \min_{x^i \in x^{p(k)}} f(x^i).$$

Otherwise, go to Step 5.

**Step 5** (*Generation of points  $u$  and  $v$* ). Calculate coordinates of points  $u$  and  $v$  for partitioning the hyperinterval  $D_t = [a_t^k, b_t^k]$ :

$$u = (a(1), \dots, a(i-1), a(i) + \frac{2}{3}(b(i) - a(i)), a(i+1), \dots, a(N)), \quad (12)$$

$$v = (b(1), \dots, b(i-1), b(i) + \frac{2}{3}(a(i) - b(i)), b(i+1), \dots, b(N)), \quad (13)$$

where  $a(j) = a_t^k(j)$ ,  $b(j) = b_t^k(j)$ ,  $1 \leq j \leq N$ , and  $i$  is found as

$$i = \min \arg \max_{1 \leq j \leq N} \{|b(j) - a(j)|\}. \quad (14)$$

Check whether the function  $f(x)$  and its partial derivatives  $f'_j(x)$ ,  $1 \leq j \leq N$ , have been already evaluated at the points  $u$  and  $v$ .

**Step 5.1.** If  $f(x)$  and  $f'_j(x)$ ,  $1 \leq j \leq N$ , have been previously evaluated at both the points  $u$  and  $v$ , then set  $p(k+1) := p(k)$  and go to Step 6.

**Step 5.2.** If  $f(x)$  and  $f'_j(x)$ ,  $1 \leq j \leq N$ , have been previously evaluated at only one point, e. g., at point  $u$  (similar operation is performed in the case when the functions evaluations have been performed only at point  $v$ ), then perform a new trial of  $f(x)$  and  $f'_j(x)$ ,  $1 \leq j \leq N$ , at point

$$x^{p(k)+1} = v,$$

set  $p(k+1) := p(k) + 1$  and go to Step 6.

**Step 5.3.** If neither  $f(x)$  nor  $f'_j(x)$ ,  $1 \leq j \leq N$ , have been previously evaluated at points  $u, v$ , then evaluate  $f(x)$  and  $f'_j(x)$ ,  $1 \leq j \leq N$ , at points

$$x^{p(k)+1} = u, \quad x^{p(k)+2} = v,$$

set  $p(k+1) := p(k) + 2$  and go to Step 6.

**Step 6** (*Efficient diagonal partition*). Obtain a new partition  $\{D^{k+1}\}$  of the search domain  $D$ :

$$\{D^{k+1}\} = \{D^k\} \setminus \{D_t\} \cup \{D_{M(k)+1}^{k+1}\} \cup \{D_{M(k)+2}^{k+1}\}, \quad (15)$$

where vertexes of the main diagonals of new hyperintervals  $D_t^{k+1}$ ,  $D_{M(k)+1}^{k+1}$  and  $D_{M(k)+2}^{k+1}$  are given by formulae

$$a_t^{k+1} = a_{M(k)+2} = u, \quad b_t^{k+1} = b_{M(k)+1} = v. \quad (16)$$

$$a_{M(k)+1} = a_t^k, \quad b_{M(k)+1} = v, \quad (17)$$

$$a_{M(k)+2} = u, \quad b_{M(k)+2} = b_t^k, \quad (18)$$

respectively.

Set  $M(k+1) := M(k) + 2$ , increase the iteration counter  $k := k + 1$  and go to Step 1.

The introduced algorithm SMOOTHD belongs to the class of diagonally extended geometric algorithms and, thus, to a more general class of ‘Divide-the-Best’ algorithms (see [40]). Its reliability parameter  $r$  from (7) controls the estimate  $m$  of the Lipschitz constant  $K$  of  $f'(x)$  and affects the algorithm’s convergence, as it will be shown in the next Section. Namely, by increasing  $r$  the reliability of the method also increases. As this parameter decreases, the search rate increases, but the probability of convergence to a point other than the global minimizer of  $f(x)$  grows as well.

The next Theorem establishes convergence properties of the proposed algorithm.

**Theorem 1.** *For any function  $f(x)$  with the gradient satisfying the Lipschitz condition (2) with the constant  $K$ ,  $0 < K < \infty$ , there exists a value  $r^*$  of the reliability parameter  $r$  of the SMOOTHD algorithm such that for any  $r \geq r^*$  the infinite ( $\varepsilon = 0$  in (11)) sequence of trial points  $\{x^{p(k)}\}$ , generated by the SMOOTHD algorithm during minimization of  $f(x)$ , will converge only to the global minimizers of  $f(x)$ .*

**Proof.** This result can be obtained as a particular case of the general convergence study of ‘Divide-the-Best’ algorithms from [40] and its proof is so omitted.  $\square$

It should be noticed in this context that if a value of  $r$  smaller than  $r^*$  is used, the algorithm can converge (see the general analysis executed for ‘Divide-the-Best’ methods in [40]) to a local minimizer of  $f(x)$  or to the boundary of a subregion of  $D$  corresponding to the best characteristics (8)–(10). This situation indicates the necessity to increase the value of the reliability parameter, i. e., it is a practical hint for the choice of  $r$ .

## 4. Numerical experiments

In this Section, results of numerical experiments aimed at evaluating the SMOOTHD method performance and comparing it with some known global optimization techniques are given. Particularly, two popular global optimization algorithms (downloadable from <http://www4.ncsu.edu/~ctk/SOFTWARE/DIRECTv204.tar.gz>) have been taken for numerical comparisons: the DIRECT method from [20] and its locally-biased version DIRECT $l$  from [10]. Both of them partition the search region  $D$  into small hyperintervals and are widely used for solving applied black-box global optimization problems, being therefore important competitors for the SMOOTHD method. They also belong to the class of ‘Divide-the-Best’ algorithms but do not use the information about the objective function gradient during their work.

All experiments were performed on a 3.40 GHz Intel(R) Core(TM) i7-2600 CPU PC with 12 Gb memory using 64-bit Windows system.

This Section is structured as follows. First, methodology of executing numerical experimentation is described presenting test functions and comparison criteria used. Then, numerical issues on the reliability parameter of the new method are discussed and sensitivity analysis of this parameter is performed. Finally, overall numerical results are summarized and commented on.

### 4.1. Classes of test functions and comparison criteria

Eight classes of differentiable test problems generated by the GKLS-generator from [12] were used to perform numerical experiments. This generator constructs three types (non-differentiable, continuously differentiable,

and twice continuously differentiable) of classes of multidimensional and multiextremal test functions with known local and global minima. The generation procedure consists of defining a convex quadratic function systematically distorted by polynomials.

Each test class provided by the generator consists of 100 functions and is defined by the following parameters: (i) problem dimension  $N$ , (ii) number of local minima  $m$ , (iii) global minimum value  $f^*$ , (iv) radius of the attraction region of the global minimizer  $\rho$ , (v) distance from the global minimizer to the quadratic function vertex  $d$ . The other necessary parameters are chosen randomly by the generator for each test function of the class. A special notebook with a complete description of all functions is supplied to the user. The GKLS-generator always produces the same test classes for a given set of the user-defined parameters, allowing one to perform repeatable numerical experiments. By changing the user-defined parameters, classes with different properties can be created. The generator is available on the ACM Collected Algorithms (CALGO) database (the CALGO is part of a family of publications produced by the Association for Computing Machinery) and it is also downloadable for free from <http://www.info.dimes.unical.it/~yaro/GKLS.html>.

In order to obtain comparable results, the same eight GKLS test classes of continuously differentiable functions of dimensions  $N = 2, 3, 4$ , and 5 defined by the same five parameters as in [44] were used (see Table 1). The global minimum value  $f^*$  was equal to  $-1.0$  and the number of local minima  $m$  was equal to 10 for all classes (these values are default settings of the GKLS-generator). Two test classes were considered for each particular dimension  $N$ : the ‘simple’ class and the ‘hard’ one. The difficulty of a class was increased either by decreasing the radius  $\rho$  of the attraction region of the global minimizer  $x^*$  (as for two- and five-dimensional classes) or by increasing the distance  $d$  from  $x^*$  to the quadratic function vertex (three- and four-dimensional classes).

In all numerical experiments, the maximal allowed number of function trials was set equal to  $p_{max} = 1\,000\,000$ . A problem was considered to be solved by a method under examination if the method generated a trial point  $x'$  in an  $\varepsilon$ -neighborhood of the global minimizer  $x^* \in D = [a, b]$ , i. e.,

$$|x'(j) - x^*(j)| \leq \sqrt[\nu]{\varepsilon}(a(j) - b(j)), \quad 1 \leq j \leq N, \quad (19)$$

where  $0 < \varepsilon < 1$  is an accuracy coefficient (its values are given in the third column of Table 1). Such a type of stopping criterion is acceptable only when

Table 1: Description of the GKLS test classes used in numerical experiments

Class	Difficulty	$\varepsilon$	$N$	$m$	$f^*$	$d$	$\rho$
1.	Simple	$10^{-4}$	2	10	-1.0	0.90	0.20
2.	Hard	$10^{-4}$	2	10	-1.0	0.90	0.10
3.	Simple	$10^{-6}$	3	10	-1.0	0.66	0.20
4.	Hard	$10^{-6}$	3	10	-1.0	0.90	0.20
5.	Simple	$10^{-6}$	4	10	-1.0	0.66	0.20
6.	Hard	$10^{-6}$	4	10	-1.0	0.90	0.20
7.	Simple	$10^{-7}$	5	10	-1.0	0.66	0.30
8.	Hard	$10^{-7}$	5	10	-1.0	0.66	0.20

the global minimizer  $x^*$  is known, i. e., in the case of test functions. When a real black-box objective function is minimized and global minimization methods have an internal stopping criterion (as that of the proposed algorithm, see formula (11)), they execute a number of iterations (that can be high) after a ‘good’ estimate of  $f^*$  has been achieved in order to demonstrate the ‘goodness’ of the solution found (see, e. g., [18, 36, 51]).

Since each evaluation of a real-life black-box objective function is usually a time-consuming operation (see, e. g., [19, 34, 36, 45, 48, 51, 54]), the maximal number of function trials executed by the methods until the satisfaction of the stopping criteria (19) was chosen as the main criterion of the comparison (other possible comparison indicators as the number of generated hyperintervals, the average number of trials, and so on can be also used, see, e. g., [33, 44, 45, 48]).

#### 4.2. The reliability parameter and its sensitivity analysis

In order to evaluate the influence of the reliability parameter  $r$  from (7) on the convergence properties of the SMOOTHD method, the two-dimensional class 1 from Table 1 was taken. A problem from this class was considered to be solved by the new method if it generated a trial satisfying condition (19) (if the method stopped due to its internal stopping criterion (11) and condition (19) was not verified, the problem was considered to be unsolved by SMOOTHD with a given value of the reliability parameter).

The following numbers were taken as indicators of the SMOOTHD method performance in this analysis subject to a fixed value  $\bar{r}$  of the reliability parameter  $r$ , the same for all 100 functions of the class:

Table 2: Influence of the reliability parameter on the SMOOTHD method performance in the case of the two-dimensional GKLS test class 1

Indicator	$\bar{r} = 1.2$	$\bar{r} = 1.8$	$\bar{r} = 2.8$	$\bar{r} = 3.8$	$\bar{r} = 4.8$	$\bar{r} = 5.8$
$p^*(\bar{r})$	199	272	374	410	424	451
$p_{\text{avg}}(\bar{r})$	105.14	169.63	247.11	293.52	323.42	341.60
$S(\bar{r})$	51	81	95	98	99	100

$S(\bar{r})$  – number of solved problems among 100 of the test class;

$p^*(\bar{r})$  – maximal number of trials required for the method to satisfy condition (19) for *all*  $S(\bar{r})$  functions of the test class;

$p_{\text{avg}}(\bar{r})$  – average number of trials required for the method to satisfy condition (19) for all  $S(\bar{r})$  functions of the test class.

Numerical results obtained by the proposed method on the considered two-dimensional test class when the reliability parameter varied from  $\bar{r} = 1.2$  up to  $\bar{r} = 5.8$  are reported in Table 2.

It can be seen from Table 2 that with increasing the value of  $r$  the reliability (the number of solved problems  $S(\bar{r})$ ) of the method increases. However, the numbers of trials (both maximal and average) increase too. This is due to the fact (well known in Lipschitz global optimization; see, e. g., [48]) that every function optimized by the proposed method has its own value  $r^*$  from Theorem 1. Therefore, when one executes tests with a class of 100 different functions it becomes difficult to use specific values of  $r$  for each function.

In fact (see an illustration given in Figures 4 and 5), for one particular function (see Figure 4, left part) a relatively small value of  $r$  can be already sufficient to find its global minimizer in sense of condition (19). In this case, a higher value of the reliability parameter (see Figure 4, right part) can cause the method to perform an additional examination of the search domain in the hope of capturing an eventual better minimum. Obviously, this exploration increases the number of trials performed by the method on this particular function (see Figure 4, right part) but allows the method to solve some other problem (as that in Figure 5) for which a greater value of  $r$  is required. An insufficient (i.e., too small) value of  $r$  makes the method converge to a local minimizer, as it is shown in Figure 5, left part. It should be noticed in this connection that in some practical applications the discovering of a local minimizer can be also very useful. Figure 5, left part, shows that when local solutions are of an interest, the algorithm SMOOTHD can provide a fast

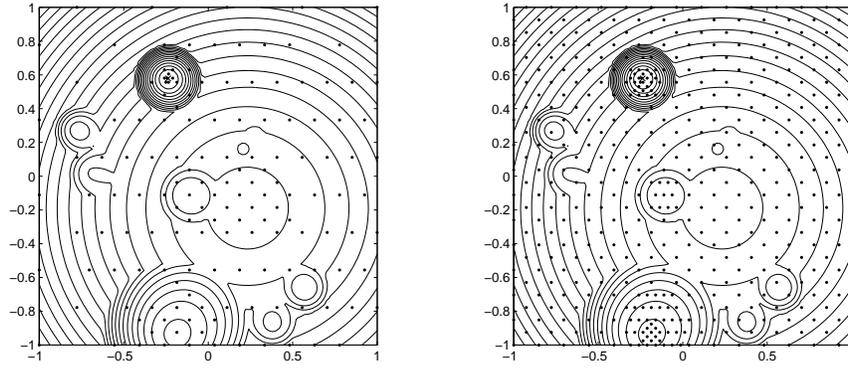


Figure 4: Trial points (black dots) generated by the SMOOTHD method when minimizing function number 58 from the GKLS test class 1,  $x^* \approx (-0.2371, 0.5791)$  (left:  $\bar{r} = 1.2$ , 152 trials; right:  $\bar{r} = 5.8$ , 452 trials).

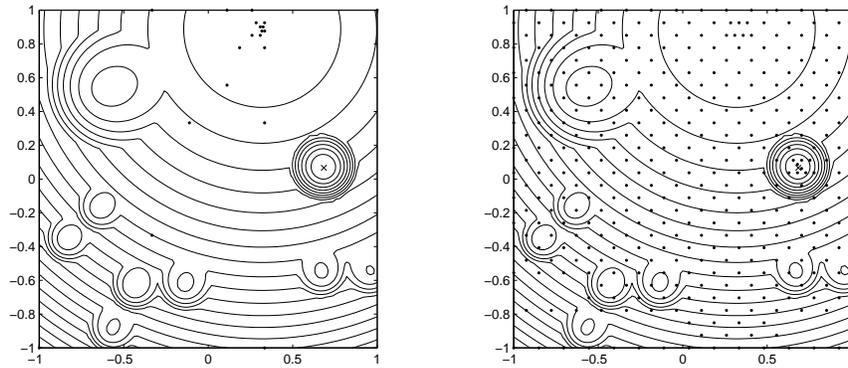


Figure 5: Trial points (black dots) generated by the SMOOTHD method when minimizing function number 54 from the GKLS test class 1,  $x^* \approx (0.6841, 0.0664)$  (left:  $\bar{r} = 1.2$ , 20 trials; right:  $\bar{r} = 5.8$ , 375 trials).

Table 3: Analysis of the adaptive reliability parameter of the SMOOTH method in the case of the two-dimensional GKLS test class 1

Indicator	$\bar{r} = 1.2$	$\bar{r} = 1.8$	$\bar{r} = 2.8$	$\bar{r} = 3.8$	$\bar{r} = 4.8$	$\bar{r} = 5.8$
$C = 10.0$						
$p^*(\bar{r})$	201	277	377	410	424	453
$p_{\text{avg}}(\bar{r})$	113.16	176.06	250.45	294.64	324.10	342.01
$S(\bar{r})$	62	86	96	98	99	100
$C = 20.0$						
$p^*(\bar{r})$	179	283	379	411	424	453
$p_{\text{avg}}(\bar{r})$	122.01	181.26	252.86	295.51	324.73	342.47
$S(\bar{r})$	69	86	96	99	99	100
$C = 50.0$						
$p^*(\bar{r})$	214	293	387	414	425	454
$p_{\text{avg}}(\bar{r})$	146.42	194.99	257.56	299.23	327.24	343.83
$S(\bar{r})$	83	92	100	100	100	100
$C = 100.0$						
$p^*(\bar{r})$	251	314	392	416	428	456
$p_{\text{avg}}(\bar{r})$	170.40	212.98	267.48	305.07	330.47	345.85
$S(\bar{r})$	92	97	100	100	100	100

convergence to local minimizers with small values of  $r$ .

In order to avoid an excessive exploration of the search region, several approaches can be used. For example, an adaptive adjustment of the reliability parameter can be performed using the following formula

$$r = \bar{r} + \frac{C}{k}, \quad (20)$$

where  $k$  is the iteration number of the method and  $C$  is a positive constant. In this way, at the initial iterations of the method (when  $k$  is small) a greater attention is paid to the exploration of the whole search domain  $D$  while for high values of  $k$  the influence of  $C$  diminishes and the reliability parameter becomes closer to its fixed value  $\bar{r}$ . As it can be seen from results in Table 3 (where the same values  $\bar{r}$  as in Table 2 were used but with different values  $C > 0$  in (20)), this adaptive technique improves the method performance.

As a practical recommendation for solving real-life global optimization

problems, the following procedure can be adopted. An initial number of trials that are to be performed is fixed and the method is started with a small value of  $r$ . Once the method stopped, it is restarted on the same pool of trial points (without re-evaluating  $f(x)$  and  $f'(x)$ ) but with a greater value of  $r$ . If after repeating these restarts several times the method converges to the same point, this point is accepted as an approximation of the global minimizer. For more information on the reliability parameter and other parameters of this kind in Lipschitz global optimization see, e. g., [48, 51].

#### 4.3. Operating characteristics of the compared methods

The overall numerical results for three compared methods on the considered GKLS test classes are summarized in this subsection. Since the DIRECT-type methods under examination do not have any internal stopping criteria, they stopped either when condition (19) was satisfied or when  $p_{max}$  trials were performed. During their work, these methods evaluate only the objective function  $f(x)$  from (1) (and do not evaluate  $f'(x)$  as the SMOOTHD method does), therefore the operation of executing a trial is less expensive in the DIRECT-type algorithms with respect to that of the new method. The balancing parameter  $\epsilon$  of the DIRECT and DIRECT $l$  methods (see [10, 20]) was set equal to  $\epsilon = 10^{-4}$ , as recommended by many authors (see, e.g., the review of different techniques for setting this parameter in [33]).

Formula (20) was used in the SMOOTHD method to set the reliability parameter  $r$  from (7), with the value  $\xi = 10^{-6}$  in (7) and the values  $C$  from (20) set in relation to the problems dimension ( $C$  equal to 50, 100, 150, and 200 in the case of  $N$  equal to 2, 3, 4, and 5, respectively). Moreover, several values of  $\bar{r}$  can be fixed for the entire class starting from the initial value  $\bar{r} = 1.1$  (the maximal values of this coefficient were equal to 2.80, 5.80, 3.60, 4.30, 5.80, 6.60, 4.10, and 7.80 for the GKLS classes from 1 to 8, respectively).

The algorithms performance can be conveniently visualized by the operating characteristics (introduced in 1978 in [15], see, e. g., [51] for their English-language description). The operating characteristics are formed by the pairs  $(p, S(p))$  where  $p$  is the number of trials and  $S(p)$  ( $0 \leq S(p) \leq S = 100$ ) is the number of test problems (among a set of  $S$  tests) solved by a method with less than or equal to  $p$  function trials. Each pair (for increasing values of  $p$ ) corresponds to a point on the plane. Higher is the graph of the operating characteristics of a method with respect to another one, better is the method performance on the considered set of tests.

In Figures 6 and 7, operating characteristics of the compared methods are shown, respectively, on simple and hard GKLS classes from Table 1. The numbers  $p^*$  of trials required for the methods to satisfy condition (19) for all 100 functions of a test class are also indicated in these Figures (after performing  $p_{max} = 1\,000\,000$  trials, the DIRECT method was unable to solve 4, 4, 7, 1, and 16 problems of the GKLS classes 4–8, respectively, and the DIRECT $l$  method was unable to solve 4 problems of the GKLS class 8).

It can be seen from the graphs reported in Figures 6 and 7, that on relatively simple functions of each GKLS class (approximately, half of 100 problems of the class quickly solved by every method) the SMOOTHD method behaves similarly to the DIRECT and DIRECT $l$  methods, in terms of the function trials performed (see, e. g., the intersection of the vertical line corresponding to  $p \approx 200$  trials and the operating characteristics graphs on the simple two-dimensional GKLS class 1 in Figure 6). However, when more and more difficult problems of the particular GKLS class are to be solved, the advantage of the new method becomes more and more pronounced with respect to the competitors (see the ascent of the operating characteristics graphs of the SMOOTHD method on all the GKLS classes in Figures 6 and 7). The most significant advantage with respect to both the DIRECT and DIRECT $l$  methods is obtained by the new method on the hard five-dimensional GKLS class 8 (which is the most difficult among the classes from Table 1).

## 5. A brief conclusion

A new method SMOOTHD has been proposed to solve global optimization problems with the objective function having the Lipschitz gradient. It generalizes the one-dimensional geometric algorithm using smooth auxiliary functions and adaptive estimate of the Lipschitz constant from [39] to the multidimensional case by using the efficient diagonal scheme from [41, 45]. The proposed method belongs to the class of ‘Divide-the-Best’ algorithms for which strong convergence properties have been established in [40]. Results of numerical experiments performed with the SMOOTHD method on several hundreds of GKLS test functions from [12] show that already in its basic version the proposed algorithm manifests a very nice performance and opens new directions for developing powerful global optimization tools. In the future, it could be interesting to consider, in the connection to the SMOOTHD method, such approaches as the local tuning [39, 45] and local improvement techniques [28, 29, 48]. Another interesting direction of research is to de-

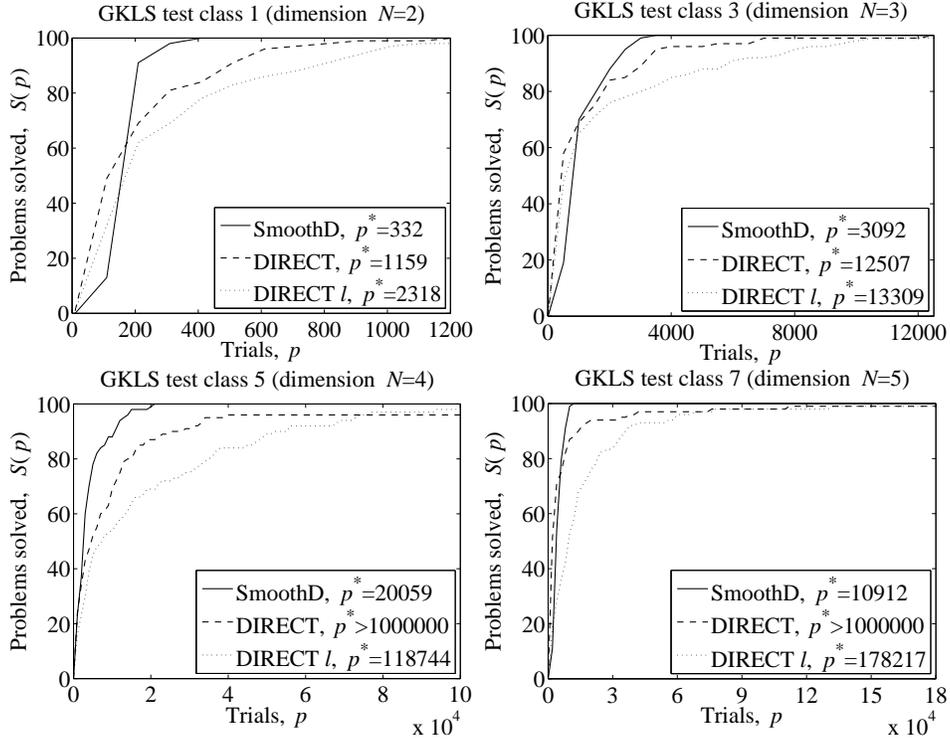


Figure 6: Operating characteristics for the compared methods SMOOTHD, DIRECT, and DIRECTl on ‘simple’ GKLS test classes 1, 3, 5, and 7 from Table 1.

velop parallel versions of the proposed method within a general framework introduced in [14, 43, 47, 51] for constructing parallel global optimization techniques.

## References

- [1] Breiman, L., Cutler, A., 1993. A deterministic algorithm for global optimization. *Math. Program.* 58, 179–199.
- [2] Calvin, J.M., Žilinskas, A., 2000. One-dimensional P-algorithm with convergence rate  $O(n^{-3+\delta})$  for smooth function. *J. Optim. Theory Appl.* 106, 297–307.
- [3] Clausen, J., Žilinskas, A., 2002. Subdivision, sampling, and initialization strategies for simplicial branch and bound in global optimization. *Comput. Math. Appl.* 44, 943–955.

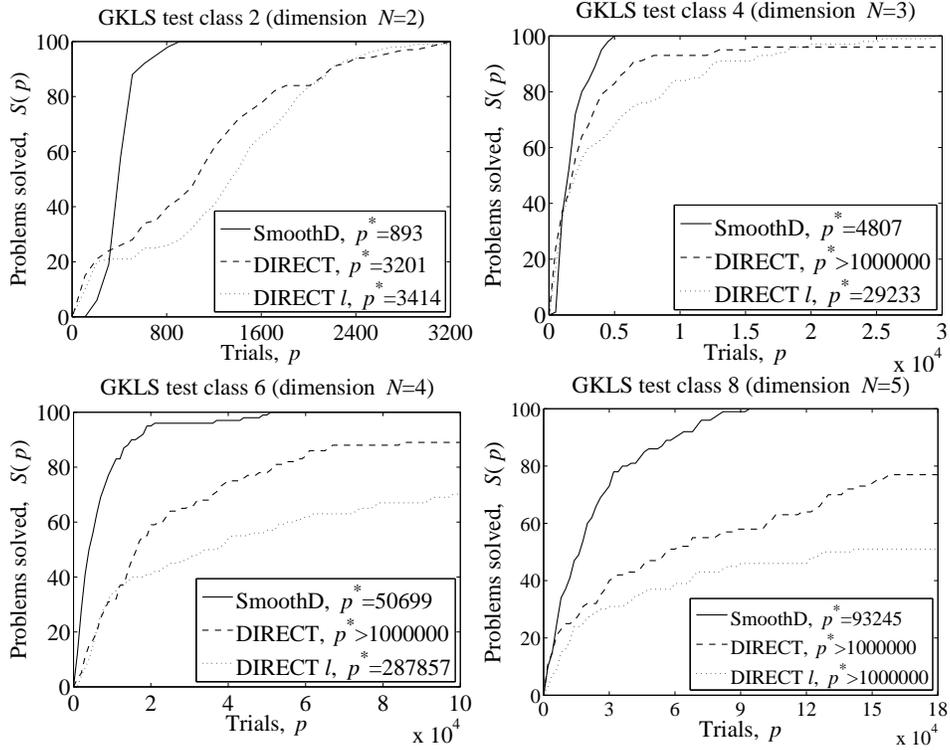


Figure 7: Operating characteristics for the compared methods SMOOTHD, DIRECT, and DIRECTl on ‘hard’ GKLS test classes 2, 4, 6, and 8 from Table 1.

- [4] Conn, A.R., Scheinberg, K., Vicente, L.N., 2009. Introduction to Derivative-Free Optimization. SIAM, Philadelphia, USA.
- [5] Custódio, A.L., Aguilar Madeira, J.F., Vaz, A.I.F., Vicente, L.N., 2011. Direct multisearch for multiobjective optimization. SIAM J. Optim. 21, 1109–1140.
- [6] Di Serafino, D., Liuzzi, G., Piccialli, V., Riccio, F., Toraldo, G., 2011. A modified Diving RECTangles algorithm for a problem in astrophysics. J. Optim. Theory Appl. 151, 175–190.
- [7] Evtushenko, Y.G., 1985. Numerical Optimization Techniques. Translations Series in Mathematics and Engineering, Springer–Verlag, Berlin.
- [8] Evtushenko, Y.G., Posypkin, M.A., 2011. An application of the nonuni-

form covering method to global optimization of mixed integer nonlinear problems. *Comput. Math. Math. Phys.* 51, 1286–1298.

- [9] Floudas, C.A., Pardalos, P.M. (Eds.), 2009. *Encyclopedia of Optimization (6 Volumes)*. Springer. 2nd edition.
- [10] Gablonsky, J.M., Kelley, C.T., 2001. A locally-biased form of the DIRECT algorithm. *J. Global Optim.* 21, 27–37.
- [11] Gandomi, A.H., Yun, G.J., Yang, X.S., Talatahari, S., 2013. Chaos-enhanced accelerated particle swarm optimization. *Commun. Nonlinear Sci. Numer. Simulat.* 18, 327–340.
- [12] Gaviano, M., Kvasov, D.E., Lera, D., Sergeyev, Y.D., 2003. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Trans. Math. Software* 29, 469–480.
- [13] Gergel, V.P., 1997. A global optimization algorithm for multivariate function with Lipschitzian first derivatives. *J. Global Optim.* 10, 257–281.
- [14] Gergel, V.P., Sergeyev, Y.D., 1999. Sequential and parallel algorithms for global minimizing functions with Lipschitzian derivatives. *Comput. Math. Appl.* 37, 163–179.
- [15] Grishagin, V.A., 1978. Operating characteristics of some global search algorithms, in: *Problems of Stochastic Search*. Zinatne, Riga. volume 7, pp. 198–206. In Russian.
- [16] Grishagin, V.A., Sergeyev, Y.D., Strongin, R.G., 1997. Parallel characteristic algorithms for solving problems of global optimization. *J. Global Optim.* 10, 185–206.
- [17] Grossmann, I.E. (Ed.), 1996. *Global Optimization in Engineering Design*. Kluwer Academic Publishers, Dordrecht.
- [18] Horst, R., Pardalos, P.M. (Eds.), 1995. *Handbook of Global Optimization*. volume 1. Kluwer Academic Publishers, Dordrecht.
- [19] Horst, R., Tuy, H., 1996. *Global Optimization – Deterministic Approaches*. Springer–Verlag, Berlin.

- [20] Jones, D.R., Perttunen, C.D., Stuckman, B.E., 1993. Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.* 79, 157–181.
- [21] Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13, 455–492.
- [22] Kolda, T.G., Lewis, R.M., Torczon, V., 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.* 45, 385–482.
- [23] Kvasov, D.E., Pizzuti, C., Sergeyev, Y.D., 2003. Local tuning and partition strategies for diagonal GO methods. *Numer. Math.* 94, 93–106.
- [24] Kvasov, D.E., Sergeyev, Y.D., 2003. Multidimensional global optimization algorithm based on adaptive diagonal curves. *Comput. Math. Math. Phys.* 43, 40–56.
- [25] Kvasov, D.E., Sergeyev, Y.D., 2009. A univariate global search working with a set of Lipschitz constants for the first derivative. *Optim. Lett.* 3, 303–318.
- [26] Kvasov, D.E., Sergeyev, Y.D., 2012a. Lipschitz gradients for global optimization in a one-point-based partitioning scheme. *J. Comput. Appl. Math.* 236, 4042–4054.
- [27] Kvasov, D.E., Sergeyev, Y.D., 2012b. Univariate geometric Lipschitz global optimization algorithms. *Numer. Algebra Contr. Optim.* 2, 69–90.
- [28] Lera, D., Sergeyev, Y.D., 2010. An information global minimization algorithm using the local improvement technique. *J. Global Optim.* 48, 99–112.
- [29] Lera, D., Sergeyev, Y.D., 2013. Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. *SIAM J. Optim.* 23, 508–529.
- [30] Liuzzi, G., Lucidi, S., Sciandrone, M., 2010. Sequential penalty derivative-free methods for nonlinear constrained optimization. *SIAM J. Optim.* 20, 2614–2635.

- [31] Mockus, J., 2000. A Set of Examples of Global and Discrete Optimization: Applications of Bayesian Heuristic Approach. Kluwer Academic Publishers, Dordrecht.
- [32] Pardalos, P.M., Romeijn, H.E. (Eds.), 2002. Handbook of Global Optimization. volume 2. Kluwer Academic Publishers, Dordrecht.
- [33] Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J., 2014. Globally-biased DISIMPL algorithm for expensive global optimization. *J. Global Optim.* 59, 545–567.
- [34] Paulavičius, R., Žilinskas, J., 2014a. *Simplicial Global Optimization*. Springer, New York.
- [35] Paulavičius, R., Žilinskas, J., 2014b. Simplicial Lipschitz optimization without the Lipschitz constant. *J. Global Optim.* 59, 23–40.
- [36] Pintér, J.D., 1996. *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)*. Kluwer Academic Publishers, Dordrecht.
- [37] Rios, L.M., Sahinidis, N.V., 2013. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.* 56, 1247–1293.
- [38] Schneider, J.J., Kirkpatrick, S., 2006. *Stochastic Optimization*. Springer, Berlin.
- [39] Sergeyev, Y.D., 1998a. Global one-dimensional optimization using smooth auxiliary functions. *Math. Program.* 81, 127–146.
- [40] Sergeyev, Y.D., 1998b. On convergence of “Divide the Best” global optimization algorithms. *Optimization* 44, 303–325.
- [41] Sergeyev, Y.D., 2000. An efficient strategy for adaptive partition of  $N$ -dimensional intervals in the framework of diagonal algorithms. *J. Optim. Theory Appl.* 107, 145–168.
- [42] Sergeyev, Y.D., Daponte, P., Grimaldi, D., Molinaro, A., 1999. Two methods for solving optimization problems arising in electronic measurements and electrical engineering. *SIAM J. Optim.* 10, 1–21.

- [43] Sergeyev, Y.D., Grishagin, V.A., 1994. A parallel method for finding the global minimum of univariate functions. *J. Optim. Theory Appl.* 80, 513–536.
- [44] Sergeyev, Y.D., Kvasov, D.E., 2006. Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM J. Optim.* 16, 910–937.
- [45] Sergeyev, Y.D., Kvasov, D.E., 2008. Diagonal Global Optimization Methods. FizMatLit, Moscow. In Russian.
- [46] Sergeyev, Y.D., Kvasov, D.E., 2011. Lipschitz global optimization, in: Cochran, J.J. (Ed.), *Wiley Encyclopedia of Operations Research and Management Science*. Wiley, New York. volume 4, pp. 2812–2828.
- [47] Sergeyev, Y.D., Strongin, R.G., 1989. A global minimization algorithm with parallel iterations. *USSR Comput. Math. Math. Phys.* 29, 7–15.
- [48] Sergeyev, Y.D., Strongin, R.G., Lera, D., 2013. *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer, New York.
- [49] Stephens, C.P., Baritompa, W., 1998. Global optimization requires global information. *J. Optim. Theory Appl.* 96, 575–588.
- [50] Strongin, R.G., 1978. *Numerical Methods in Multiextremal Problems (Information-Statistical Algorithms)*. Nauka, Moscow. In Russian.
- [51] Strongin, R.G., Sergeyev, Y.D., 2000. *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*. Kluwer Academic Publishers, Dordrecht.
- [52] X.-S. Yang, 2010. *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, USA.
- [53] Zhigljavsky, A.A., 1991. *Theory of Global Random Search*. Kluwer Academic Publishers, Dordrecht.
- [54] Zhigljavsky, A.A., Žilinskas, A., 2008. *Stochastic Global Optimization*. Springer, New York.
- [55] Žilinskas, A., Žilinskas, J., 2010. Interval arithmetic based optimization in nonlinear regression. *Informatika* 21, 149–158.