A modification of the DIRECT method for Lipschitz global optimization for a symmetric function

Ratko Grbić
Faculty of Electrical Engineering, University of Osijek
Kneza Trpimira 2b, HR – 31 000 Osijek, Croatia
e-mail: rgrbic@etfos.hr

Emmanuel Karlo Nyarko
Faculty of Electrical Engineering, University of Osijek
Kneza Trpimira 2b, HR – 31 000 Osijek, Croatia
e-mail: nyarko@etfos.hr

Rudolf Scitovski[1]
Department of Mathematics, University of Osijek
Trg Lj. Gaja 6, HR – 31 000 Osijek, Croatia
e-mail: scitowsk@mathos.hr

Abstract. In this paper, we consider a global optimization problem for a symmetric Lipschitz continuous function. An efficient modification of the well-known DIRECT (DIviding RECTangles) method called SymDIRECT is proposed for solving this problem. The method is illustrated and tested on several standard test functions. The application of this method to solving complex center-based clustering problems for the data having only one feature is particularly presented.

Key words: Lipschitz continuous function; Global optimization; DIRECT; Symmetric function; Center-based clustering

MSC2010: 65K05, 05E05, 90C26, 90C27, 90C56

1 Introduction

A real symmetric function $g\colon [a,b]^n \to \mathbb{R}$, of n variables is the one whose value at any n-tuple of arguments is the same as its value at any permutation of that n-tuple. These functions are often the subject of research in different applications. In this paper, we shall consider symmetric functions that occur naturally when solving the k-means problem and in cluster analysis (see, e.g., [20, 24, 47]), whereby special importance is attached to searching for a globally optimal partition of the data that have only one feature. If the function g attains its global minimum on $[a,b]^n$, then generally there exist at least $n!$ points from $[a,b]^n$ where this global minimum is attained. Namely, if the point $(\xi_1^\star,\ldots,\xi_n^\star) \in$

[1]Corresponding author: Rudolf Scitovski, e-mail: scitowsk@mathos.hr, telephone number: +385-31-224-800, fax number: +385-31-224-801

argmin g, then also $(\eta_1^\star, \dots, \eta_n^\star) \in \operatorname{argmin} g$, where $(\eta_1^\star, \dots, \eta_n^\star)$ is any permutation of the numbers $(\xi_1^\star, \dots, \xi_n^\star)$.

We consider a global optimization problem of a symmetric function $g\colon [a,b]^n \to \mathbb{R}$, which satisfies the *Lipschitz condition* on $[a,b]^n$

$$|g(x) - g(y)| \leq L\|x - y\|, \quad \forall x, y \in [a,b]^n, \tag{1}$$

where $L > 0$ is the Lipschitz constant. Furthermore, we shall simply refer to the set of all such functions as *Lipschitz continuous functions* and represent this set as $Lip_L[a,b]^n$. The set $Lip_L[a,b]^n$ is extensive and covers a wide range of applications. Among many recent works related to the global optimization problem for Lipschitz continuous functions, let us mention only [6, 10, 17, 18, 32, 34, 36, 39, 40, 54]. There are several approaches for solving this problem. Some of the most frequently referred ones are: partition and branch-and-bound strategies [23, 31, 36, 38, 39, 41, 42, 51], grid and random search [2, 15, 19, 25, 26, 33, 46, 52, 54], interval analysis [17]. Thereby, parallel programming is often used in the implementation of algorithms [16, 46]. It is believed that the first methods for searching for a global minimum of a univariate Lipschitz continuous function $f\colon [a,b] \to \mathbb{R}$ were proposed by Pijavskiy [35] and Shubert [43] in 1972, independently and at the same time. After that, numerous modifictions and other approaches came into existence (see, e.g., [6, 23, 26–28, 45, 48, 53]).

In this paper, we consider a global optimization problem for Lipschitz symmetric functions g and propose an efficient method for searching for a global minimum of such functions. Let us first mention two illustrative examples of symmetric functions which will be used later as test-functions.[2]

Example 1. *We consider the function* $g\colon [0,11] \times [0,11] \to \mathbb{R}$

$$g(y_1, y_2) = -\tfrac{1}{5}(y_1^2 + y_2^2) + 2y_1y_2 \cos y_1 \cos y_2$$

This differentiable symmetric function satisfies the Lipschitz condition with the constant $L \approx 150$ *and attains its local minimum in two points, and the global minimum* $g_{min} = -147.105$ *in two points* $y^\star = (9.56028, 6.45769)$, $y^{\star\star} = (6.45769, 9.56028)$ *(see Fig. 4).*

Example 2. *Let* $\mathcal{A} = \{a_i \in \mathbb{R} \colon i = 1, \dots, m\}$ *be a given set of real numbers. The elements of the set* $\mathcal{A}$ *should be partitioned into* $1 \leq k \leq m$ *nonempty disjoint clusters* $\pi_1, \dots, \pi_k$. *If* $d\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$, $\mathbb{R}_+ = [0, +\infty\rangle$ *is some distance-like function (see, e.g., [24, 47]), then to each cluster* π_j *from the partition* $\Pi = \Pi(\pi_1, \dots, \pi_k)$ *we can associate its center*

$$c_j = c(\pi_j) := \operatorname*{argmin}_{x \in \operatorname{conv}(\pi_j)} \sum_{a_i \in \pi_j} d(x, a_i),$$

[2]Other examples can be found in [7, 11, 31, 36] and on the web site
`http://www.geatbx.com/docu/fcnindex-01.html.`

and by introducing the objective function

$$\mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{a_i \in \pi_j} d(c_j, a_i),$$

we can define the quality of a partition. By using the minimal distance principle [24, 47] the problem of searching for a optimal partition is reduced to the following global optimization problem

$$\min_{(z_1,\ldots,z_k)\in \operatorname{conv} \mathcal{A}^k} F(z_1, \ldots, z_k), \qquad F(z_1, \ldots, z_k) = \sum_{i=1}^{m} \min_{j=1,\ldots,k} d(z_j, a_i). \tag{2}$$

The objective function F is a symmetric Lipschitz continuous [36, 37] function which can have a great number of independent variables, it does not have to be either convex or differentiable and it generally may have several local and global minima. Therefore, this becomes a complex global optimization problem, which can be found in the literature as a center-based clustering problem [21, 24, 37, 44, 47]. It is clear that the problem becomes even more complex if data have several features.

The most popular algorithm for finding locally optimal partitions is the k-means algorithm [14, 20, 24]. By providing a good initial approximation (see, e.g., [36, 49]), this method can produce acceptable solutions. In case we do not have a good initial approximation, multi-run algorithms with various random initializations are usually recommended [29].

Global optimization problems for a symmetric Lipschitz continuous function often occur in various applications. For example, in [5], the global optimization problem that arises in the detection of Gravitational Waves is considered. Thereby the objective function is symmetric Lipschitz continuous with unavailable derivatives and many local maxima.

The paper is organized as follows. In the next section, the `DIRECT` method for Lipschitz global optimization is briefly described. In Section 3, a modification of the `DIRECT` method referred to as `SymDIRECT` is described for symmetric Lipschitz continuous functions with special emphasis on the process of dividing hyperrectangles for a subdomain of the symmetric function. Initially, as a motivation, we briefly describe this process in $\mathbb{R}^2$, although it is not possible to naturally extend the generalization on $\mathbb{R}^n$. Hence, the process in $\mathbb{R}^3$ and $\mathbb{R}^n$ is analyzed and illustrated in detail. In Section 4, the `SymDIRECT` algorithm for Lipschitz global optimization of a symmetric function is described. Numerical results on a set of global optimization test problems are provided in Section 5. The application of `SymDIRECT` method to data clustering is emphasized in particular. The conclusions and future work are discussed in Section 6.

2 DIRECT method for Lipschitz global optimization

A derivative-free, deterministic sampling method for global optimization on a bound-constrained region named *Dividing Rectangles* (DIRECT) was proposed by [23]. This method emerged as a natural generalization of the method proposed by Pijavskiy [35] and Shubert [43]. For a Lipschitz function $f \in Lip_L[a,b]$ condition (1) can be written as

$$-L \le \frac{f(x)-f(y)}{x-y} \le L, \quad x \neq y,$$

from where it can be easily seen that the following holds:

$$\begin{aligned} f(x) &\ge f(c) + L(x-c), \quad x \le c, \\ f(x) &\ge f(c) - L(x-c), \quad x \ge c, \end{aligned}$$

where $c = \frac{a+b}{2}$ is the midpoint of the interval $[a,b]$. As a result, we easily obtain a simple concave function which represents the lower bound of the function f and which on the interval $[a,b]$ attains the least $\mathcal{B}$-value

$$\mathcal{B}(c,d) = f(c) - L\,d, \quad d = \tfrac{b-a}{2}, \tag{3}$$

that depends only on the value of the function f in the midpoint of the interval $[a,b]$ and on the width of the interval (see Fig 1).

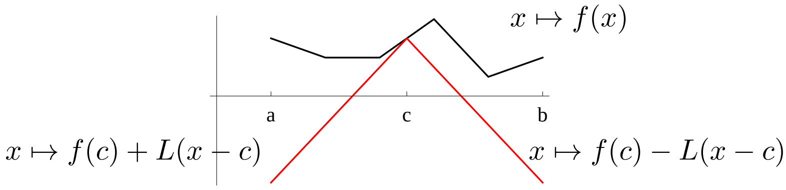

Figure 1: Lower bound of the function f

The DIRECT algorithm is based on the fact that the interval $[a,b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c. For each subinterval, the $\mathcal{B}$-value is determined according to (3) and the subinterval with the least $\mathcal{B}$-value is divided further. The global minimum of the function f is then searched for between the points representing the centers of subintervals. Based upon this, [23] proposed the following constructive idea.

Assuming that at a given stage in the algorithm there are m subintervals $[\alpha_1,\beta_1],\ldots,[\alpha_m,\beta_m]$ with centers $c_1,\ldots,c_m$ and half-widths $d_1,\ldots,d_m$. To each interval we associate the point

$$T_i = (d_i, f(c_i)), \quad i = 1\ldots,m.$$

These points can be displayed graphically as in Fig. 2. One notices that the points are arranged in columns, and sorted according to the function value of the center of each corresponding interval.

The straight line with the slope L which passes through the point T_i has an ordinate-intercept which represents the $\mathcal{B}$-value of the function f on the interval $[\alpha_i, \beta_i]$. Therefore the least $\mathcal{B}$-value, and thereby also the interval to be divided further, is chosen among $\mathcal{B}$-value intervals that are represented in Fig. 2 by points lying on the lower edge of the convex hull of the points T_i. These intervals are called *potentially optimal intervals*, and they can be defined without using the Lipschitz constant L (see Definition 1).

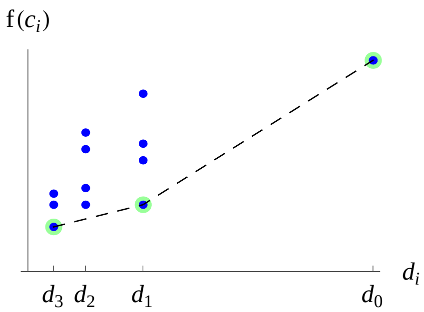

Figure 2: Clusters of points $T_i = (d_i, f(c_i))$, $i = 1 \dots, m$

Generally, we consider a global optimization problem for the Lipschitz function $g\colon \Omega \to \mathbb{R}$, $\Omega = \prod_{i=1}^{n}[a_i, b_i]$. By using mapping $T\colon \Omega \to [0,1]^n$,

$$T(x) = D(x-u), \qquad D = \operatorname{diag}\left(\tfrac{1}{b_1-a_1}, \dots, \tfrac{1}{b_n-a_n}\right), \quad u = (a_1, \dots, a_n), \tag{4}$$

which maps a hyperrectangle $\prod_{i=1}^{n}[a_i, b_i]$ to a unit hypercube $[0,1]^n$, a global optimization problem for the function g is reduced to the global optimization problem for the Lipschitz function $f\colon [0,1]^n \to \mathbb{R}$, $f = g \circ T^{-1}$, where $T^{-1}\colon [0,1]^n \to \Omega$, $T^{-1}(x) = D^{-1}x + u$.

By means of a standard strategy (see, e.g., [8, 9, 12, 23]), the unit hypercube $[0,1]^n$ with the center at the point $c = (\frac{1}{2}, \dots, \frac{1}{2}) \in \mathbb{R}^n$ is divided into smaller hyperrectangles, out of which one has again the center in the point c. In that way, we obtain new hyperrectangles $R_i(c_i, (h_1^{(i)}, \dots, h_n^{(i)}))$ with centers at points $c_i = (\zeta_1^{(i)}, \dots, \zeta_n^{(i)})$ and half side-lengths $h_j^{(i)}$, $j = 1, \dots, n$ in the direction of the j-th unit vectors e_j. If to every hyperrectangle R_i we associate the "size" of a hyperrectangle as the number $d_i = \max\{h_1^{(i)}, \dots, h_n^{(i)}\}$, then the aforementioned hyperrectangle may be denoted by $R_i(c_i, d_i)$. Of all available hyperrectangles we shall single out *potentially optimal* hyperrectangles according to the following definition [9, 12, 23]:

Definition 1. Let $\mathcal{H}$ be the set of hyperrectangles created by `DIRECT` after k iterations, let $\epsilon > 0$ be a positive constant and f_{min} the best value of the objective function found so far. A hyperrectangle $R_j(c_j, d_j) \in \mathcal{H}$ with center c_j and size d_j is said to be potentially optimal if there exists $\tilde{K} > 0$ such that

$$\begin{aligned} &f(c_j) - \tilde{K}d_j \le f(c_i) - \tilde{K}d_i, \qquad \forall R(c_i, d_i) \in \mathcal{H}, \\ &f(c_j) - \tilde{K}d_j \le f_{min} - \epsilon|f_{min}|. \end{aligned}$$

The recommended value for ϵ is 10^{-4} [9]. As mentioned earlier, it is shown that it suffices to divide further only potentially optimal hyperrectangles [7, 9, 23]. If the objective function is continuous in a neighborhood of a global optimizer, `DIRECT` converges to the globally optimal function value [23]. More information about convergence properties of the method can also be found in [8, 9].

`DIRECT` is a robust method because it works with black-box function evaluations and it is insensitive to discontinuities, missing values, and other common problems. The drawback of the `DIRECT` method is that sometimes it has problems identifying the exact solution even if it is close to it [16] or slow convergence when solving multiextremal problems [19, 30, 31, 42]. Additionally, defining a meaningful stopping criterion is also a problem [42].

There are numerous papers in which this method for global optimization is analyzed, modified, enhanced, and tested [3, 4, 9, 13, 16, 26, 51]. In our paper, we propose a modification of the `DIRECT` algorithm for Lipschitz global optimization for a symmetric function.

3 A modification of the `DIRECT` method for a symmetric Lipschitz function

Let us consider a Lipschitz global optimization problem for the symmetric function $g\colon [a,b]^n \to \mathbb{R}$. Suppose that there exists $y^\star = \operatorname*{argmin}_{y\in[a,b]^n} g(y)$. By using the corresponding mapping (4) the problem is reduced to a global optimization problem for the function $f\colon [0,1]^n \to \mathbb{R}$, $f = g\circ T^{-1}$, where $T^{-1}\colon [0,1]^n \to [a,b]^n$

$$T^{-1}(x) = (b-a)x + u, \quad u = (a,\dots,a) \in \mathbb{R}^n. \tag{5}$$

Since the function f is symmetric, as mentioned earlier in the Introduction, it suffices to solve the following global optimization problem [45]:

Find the point $x^\star = \operatorname*{argmin}_{x\in\Delta} f(x)$*, such that* $f(x^\star) = \inf_{x\in\Delta} f(x)$*, where*

$$\Delta = \{x \in [0,1]^n \colon x_1 \ge \cdots \ge x_n\}. \tag{6}$$

Note that the region Δ represents the $n!$-th part of the domain of the function f and in that way we have significantly reduced the region of searching for the global minimum of the function f. After finding $x^\star = \operatorname*{argmin}_{x\in\Delta} f(x)$, the global minimum point of the initial function g is given by $y^\star = u + (b-a)x^\star$.

The `DIRECT` method described in Section 2 shall be applied to this special situation. This means that in the procedure of dividing some hyperrectangle attention should be paid to the part of the region $[0,1]^n$ it appears in. Thereby, the following situations might occur:

(i) If the whole hyperrectangle is located in the region Δ, all rectangles that emerge by its division will also be contained in the region Δ. All of them are also liable to further division.

(ii) If a hyperrectangle appears outside the region Δ, the point of the global minimum we search for cannot appear therein. Hence such hyperrectangles shall not be divided further.

(iii) If a hyperrectangle lies in the region Δ only partially, some hyperrectangles that come into existence by its division can be fully contained in the region Δ (classified under case (i)), some might be completely outside the region Δ (classified under case (ii)), and some might lie only partially in the region Δ (again classified under case (iii)).

Sufficient conditions should be determined for all given cases, and they should be tested in the iterative procedure. After a certain number of steps, only case (i) will occur and then it will not be necessary to check any of the conditions (see Example 4 and Fig. 7).

All hyperrectangles that are at least partially located in the region Δ are candidates for further division. A set of potentially optimal hyperrectangles should be identified among them. This can be done efficiently in the following way. First, a set of all hyperrectangles, that are at least partially located in the region Δ, is partitioned into clusters, where in some cluster there are hyperrectangles with equal size. The set of indices of these clusters of rectangles is denoted by $I = \{1, \ldots, k\}$. Inside the cluster, hyperrectangles are sorted according to increasing values of the objective function in their centers. By choosing from each cluster only the hyperrectangle with the smallest value of the objective function in its center, we form an *expanded set of potentially optimal hyperrectangles* $\mathcal{E} = \{R_i(c_i, d_i)\colon i \in I\}$. The following lemma from the set $\mathcal{E}$ separates the set of *potentially optimal hyperrectangles* $\mathcal{P}$.

Lemma 1. *[12] Let $f\colon [0,1]^n \to \mathbb{R}$ be a Lipschitz function with constant $L > 0$, let $\epsilon > 0$ be a positive constant and f_{min} the current best function value. Let $\mathcal{E} = \{R_i(c_i, d_i)\colon i \in I\}$ be the expanded set of potentially optimal hyperrectangles, and*

$$I_1 = \{i \in I\colon d_i < d_j\}, \qquad I_2 = \{i \in I\colon d_i > d_j\}.$$

Hyperrectangle $R_j(c_j, d_j) \in \mathcal{E}$ is potentially optimal *according to Definition 1 if*

$$(i) \quad \max_{i \in I_1} \frac{f(c_j) - f(c_i)}{d_j - d_i} \le \min_{i \in I_2} \frac{f(c_j) - f(c_i)}{d_j - d_i}, \qquad \text{if } I_1 \neq \emptyset \text{ and } I_2 \neq \emptyset, \tag{7}$$

and

$$(ii) \qquad \epsilon \le \frac{f_{min} - f(c_j)}{|f_{min}|} + \frac{d_j}{|f_{min}|}K, \qquad \textit{if } f_{min} \neq 0 \tag{8}$$

or

$$(iii) \qquad f(c_j) \le d_j K, \qquad \textit{if } f_{min} = 0, \tag{9}$$

where

$$K = \begin{cases} \min\limits_{i \in I_2} \frac{f(c_j) - f(c_i)}{d_j - d_i}, & \textit{if } I_2 \neq \emptyset, \\ L, & \textit{if } I_2 = \emptyset. \end{cases} \tag{10}$$

The current minimum value of the objective function is updated by using the set of potentially optimal rectangles $\mathcal{P}$ with centers in the region Δ. Note that in this way it is possible to determine potentially optimal hyperrectangles, whereby it is not necessary to know the Lipschitz constant L. In (10), for L it suffices to take some large number. More detail about the problem of determining the Lipschitz constant and possibilities referring to elimination of the necessity of specifying the Lipschitz constant can be found in [26–28, 40–42, 45, 46, 50, 51, 53].

It remains to find an efficient method for identification of those hyperrectangles that are completely or at least partially located in the region Δ.

3.1 Dividing rectangles in $\mathbb{R}^2$

Let us first consider a geometrically simple global optimization problem for a symmetric function $f\colon [0,1]^2 \to \mathbb{R}$, on which all necessary facts can be noticed easily.

Find the point $x^\star = \operatorname{argmin}_{x\in\Delta} f(x)$, *such that* $f(x^\star) = \inf_{x\in\Delta} f(x)$, *where*

$$\Delta = \{x = (x_1, x_2) \in [0,1]^2 \colon x_1 \ge x_2\}. \tag{11}$$

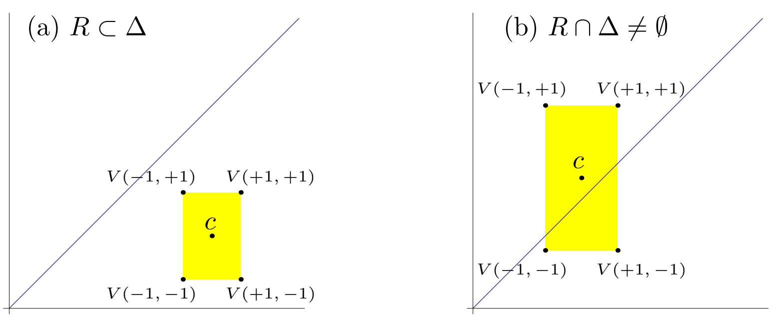

Figure 3: Rectangle $R(c,(h_1,h_2))$ completely or partially contained in the region Δ

Let $R(c,(h_1,h_2))$ be a rectangle contained in the unit square $[0,1]^2$ with center in the point $c = (\zeta_1, \zeta_2)$, and the half side-lengths h_1, h_2 in the direction of unit vectors e_1, e_2 and vertices $V(\sigma_1,\sigma_2) = (\zeta_1 + \sigma_1 h_1, \zeta_2 + \sigma_2 h_2)$, where $\sigma_1, \sigma_2 \in S = \{-1,+1\}$ (see Fig. 3).

Note first that some point $T = (\mu_1, \mu_2)$ lies in the region Δ if and only if $1 \geq \mu_1 \geq \mu_2 \geq 0$. It is easy to see that the whole rectangle R lies in the region Δ if and only if the vertex $V(-1, +1)$ lies in the region Δ (see Fig. 3a), and this is fulfilled if and only if there holds

$$\zeta_1 - h_1 \geq \zeta_2 + h_2, \tag{12}$$

A rectangle is located at least partially in the region Δ if at least the vertex $V(+1, -1)$ is located in the region Δ (see Fig. 3b), i.e. if the following condition is fulfilled

$$\zeta_1 + h_1 \geq \zeta_2 - h_2. \tag{13}$$

By knowing conditions (12) and (13), we can considerably accelerate the procedure of dividing rectangles while searching for the global minimum of the symmetric function.

Example 3. *For a symmetric Lipschitz function from Example 1 we shall conduct the described modification of the* `DIRECT` *algorithm. First, by means of mapping* $T\colon [0, 11]^2 \to [0, 1]^2$, $T(x) = Dx$, *where* $D = \operatorname{diag}(\frac{1}{11}, \frac{1}{11})$, *the problem is normed as a global optimization problem for the symmetric function* $f = g \circ T^{-1}\colon [0, 1]^2 \to \mathbb{R}$, *where* $T^{-1}(x) = D^{-1}x$, $D^{-1} = \operatorname{diag}(11, 11)$.

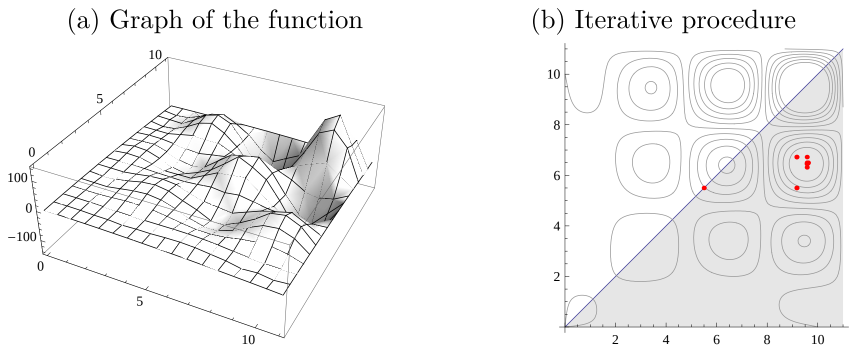

Figure 4: Searching for the global minimum of function $g(y_1, y_2) = -\frac{1}{5}(y_1^2 + y_2^2) + 2y_1y_2 \cos y_1 \cos y_2$

By using the aforementioned modification of the `DIRECT` method, after 21 iterations we obtain the approximation of the global minimum correct to one decimal place

$$\hat{y} = (9.6, 6.5), \quad f(\hat{y}) = -146.894.$$

Consequently, 18 rectangles appeared outside the region Δ, 256 within the region Δ and 9 partially in the region Δ. Also, the total of 245 function evaluations were performed. Fig. 4b shows a `ContourPlot` of the function f with the denoted region Δ and centers of rectangles generated during the division procedure and in which the value of the minimizing function has been updated.

3.2 Dividing rectangles in $\mathbb{R}^3$

Unfortunately, the application of the `DIRECT` method for solving a global optimization problem for a symmetric function in $\mathbb{R}^3$ cannot be obtained directly by generalizing the conclusion from Subsection 3.1. In the sequel, we consider in detail the problem in $\mathbb{R}^3$ since in that way a generalization for solving a global optimization problem for a symmetric function in $\mathbb{R}^n$ is obtained easily.

Since f is a symmetric function, we shall search only for the point of the global minimum that lies in the tetrahedron

$$\Delta = \{(x_1, x_2, x_3) \in [0,1]^3 \colon x_1 \geq x_2 \geq x_3\}, \tag{14}$$

(see the model in Fig. 5a). Note that some point $T = (\mu_1, \mu_2, \mu_3) \in [0,1]^3$ from the region Δ if and only if $1 \geq \mu_1 \geq \mu_2 \geq \mu_3 \geq 0$.

The following lemma yields conditions by which some parallelepiped $R \subset [0,1]^3$ lies completely or only partially in tetrahedron Δ.

Lemma 2. *Let $R(c, (h_1, h_2, h_3))$ be a parallelepiped contained in the unit cube $[0,1]^3$ with the center $c = (\zeta_1, \zeta_2, \zeta_3)$, half side-lengths h_i in the direction of unit vectors e_i and vertices $V(\sigma_1, \sigma_2, \sigma_3) = (\zeta_1 + \sigma_1 h_1, \zeta_2 + \sigma_2 h_2, \zeta_3 + \sigma_3 h_3)$, where $\sigma_1, \sigma_2, \sigma_3 \in S = \{-1, +1\}$. Then it holds*

(i) $R \subset \Delta$ if and only if

$$\zeta_1 - h_1 \geq \zeta_2 + h_2 \quad \textit{and} \quad \zeta_2 - h_2 \geq \zeta_3 + h_3. \tag{15}$$

(ii) $R \cap \Delta \neq \emptyset$ if and only if

$$(\zeta_1 + h_1 \geq \zeta_2 - h_2 \geq \zeta_3 - h_3), \quad \textit{or} \quad (\zeta_1 + h_1 \geq \zeta_2 + h_2 \geq \zeta_3 - h_3). \tag{16}$$

Proof. Let us notice first that the following holds

$$\zeta_i + h_i \geq \zeta_i - h_i, \quad i = 1, 2, 3. \tag{17}$$

(i) If $R \subset \Delta$, then all vertices of R are contained in Δ, i.e. there holds

$$\zeta_1 + \sigma_1 h_1 \geq \zeta_2 + \sigma_2 h_2 \geq \zeta_3 + \sigma_3 h_3 \quad \forall \sigma_1, \sigma_2, \sigma_3 \in S, \tag{18}$$

from where follows (15).

Conversely, if (15) holds, by using (17), we obtain

$$\zeta_1 + h_1 \overset{(17)}{\geq} \zeta_1 - h_1 \overset{(15)}{\geq} \zeta_2 + h_2 \overset{(17)}{\geq} \zeta_2 - h_2 \overset{(15)}{\geq} \zeta_3 + h_3 \overset{(17)}{\geq} \zeta_3 - h_3,$$

from where holds (18), i.e. $V(\sigma_1, \sigma_2, \sigma_3) \in \Delta$ for every $\sigma_1, \sigma_2, \sigma_3 \in S = \{-1, +1\}$, and this means that $R \subset \Delta$.

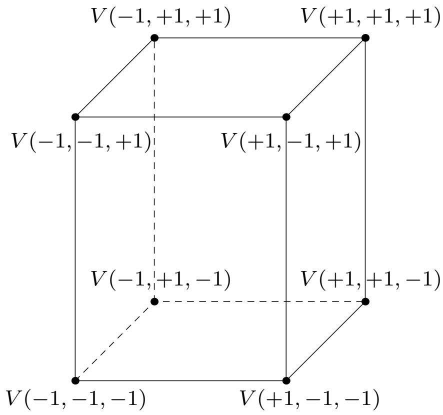

Figure 5: Tetrahedron Δ and parallelepiped $R(c,(h_1,h_2,h_3))$

(*ii*) Suppose that $R\cap\Delta\neq\emptyset$. This means that at least one of the vertices of the parallelepiped lower basis is contained in Δ, i.e. there exist $\sigma_1,\sigma_2\in S=\{-1,+1\}$, such that $V(\sigma_1,\sigma_2,-1)\in\Delta$, i.e.

$$\zeta_1+\sigma_1h_1\geq\zeta_2+\sigma_2h_2\geq\zeta_3-h_3.$$

Since $\sigma_1\leq 1$, we obtain

$$\zeta_1+h_1\geq\zeta_1+\sigma_1h_1\geq\zeta_2+\sigma_2h_2\geq\zeta_3-h_3,$$

which means that there exists $\sigma_2\in S$, such that $V(+1,\sigma_2,-1)\in\Delta$ i.e. there holds (16).

Conversely, if (16) holds, then at least one of the vertices $V(+1,-1,-1)$, $V(+1,+1,-1)$ is contained in Δ, so that $R\cap\Delta\neq\emptyset$.

$\square$

Example 4. *The set of $m=10$ uniform distributed random numbers from* $[0,1]$

$\mathcal{A}=\{0.00837,0.01431,0.04052,0.25126,0.45654,0.70427,0.74395,0.795,0.86823,0.95783\}.$

should be partitioned into 3 *clusters.*

In this case, minimizing function (2) attains its local minimum and its global minimum at the point $\hat{z}$ and the point $z^\star$, respectively (see also Fig. 6).

$$\begin{aligned}&\hat{z}=(0.81386,0.45654,0.07861)\in\Delta, &&F(\hat{z})=0.08126,\\ &z^\star=(0.81385,0.35390,0.02106)\in\Delta, &&F(z^\star)=0.06258,\\ &\text{where}\quad \Delta=\{(x_1,x_2,x_3)\in[0,1]^3\colon x_1\geq x_2\geq x_3\}.\end{aligned}$$

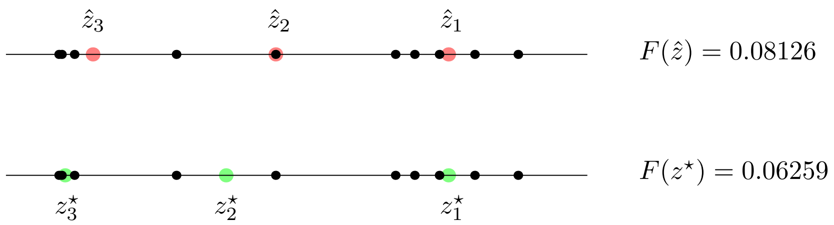

Figure 6: The data and the centers

However, local and global minima are also attained at every point from the unit cube $[0,1]^3$, whose coordinates are a permutation of the coordinates of the point $\hat{z}$ (i.e. the point $z^{\star}$).

After 36 iterations, a modified `DIRECT` algorithm for this symmetric function attains a global minimum correct to 4 decimal places, whereby 491 evaluations of function values were performed. 367, 68 and 24 rectangles appeared completely, partially and outside of the region Δ, respectively. Dynamics of their appearance per iteration is shown in Fig. 7, whereas the number of rectangles of equal size per iteration is given in Fig. 8a.

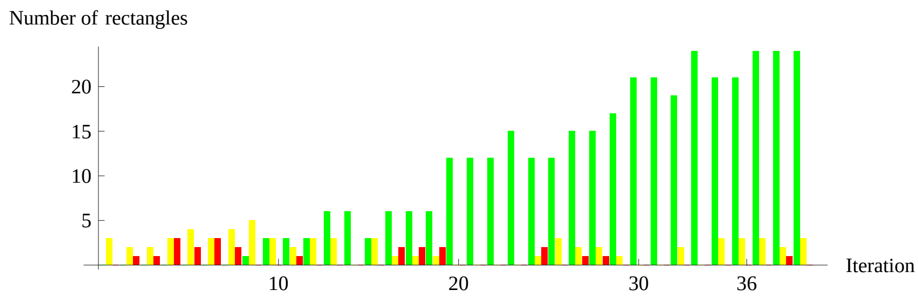

Figure 7: Appearance of rectangles inside (green), on boundary (yellow), and outside (red) of the region Δ

Convergence of the iterative procedure is illustrated in Fig. 8b and Fig. 9. Fig. 8b shows the data (small black dots) and shifting of cluster centers during the iterative process. Fig. 9a and Fig. 9b show shifting of values of the objective function and the approximation error of the global minimum per iteration, respectively.

3.3 Dividing rectangles in $\mathbb{R}^n$

The problem of searching for a global minimum of the symmetric function $f\colon [0,1]^n \to \mathbb{R}$ is generally reduced to optimization in hypertetrahedron

$$\Delta = \{(x_1, \ldots x_n) \in [0,1]^n \colon x_1 \geq x_2 \geq \cdots \geq x_n\}. \tag{19}$$

Note that some point $T = (\mu_1, \ldots, \mu_n) \in [0,1]^n$ is contained in the set Δ if and only if $\mu_1 \geq \cdots \geq \mu_n$. The next theorem is a generalization of Lemma 2 and it gives conditions by which some hyperrectangle completely or partially lies in the region Δ.

(a) Density of rectangles of equal size ($d_i = \frac{1}{2\cdot 3^i}$) (b) Position of cluster centers

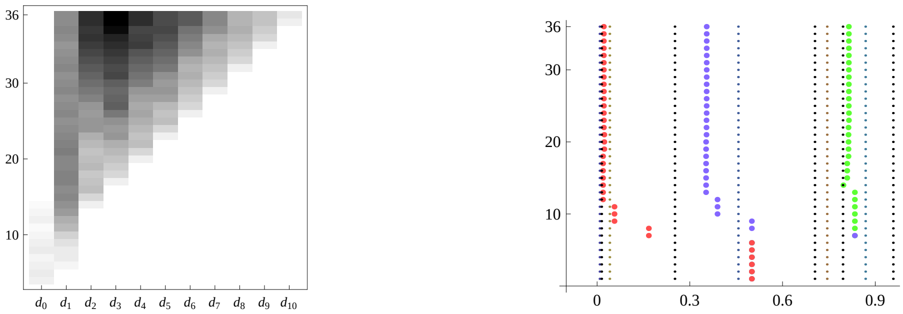

Figure 8: Illustration of the iterative process

(a) $i \mapsto F(z_1^{(i)}, z_2^{(i)}, z_3^{(i)})$ (b) $i \mapsto \|(z^{(i)}, z_2^{(i)}, z_3^{(i)}) - (z_1^\star, z_2^\star, z_3^\star)\|$

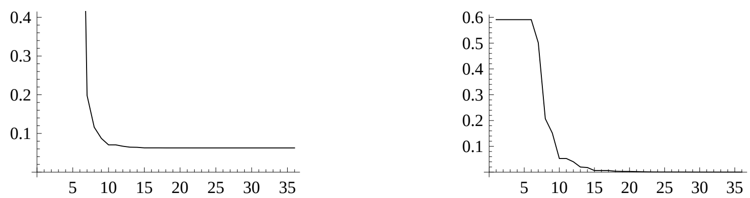

Figure 9: Illustration of convergence of the iterative process

Theorem 1. *Let $R(c,(h_1,\ldots,h_n))$ be a hyperrectangle contained in the unit hypercube $[0,1]^n$ with the center $c=(\zeta_1,\ldots,\zeta_n)$, half side-lengths h_i in the direction of unit vector e_i and vertices $V(\sigma_1,\ldots,\sigma_n)=(\zeta_1+\sigma_1 h_1,\ldots,\zeta_n+\sigma_n h_n)$, where $\sigma_1,\ldots,\sigma_n\in S=\{-1,+1\}$. Then the following holds:*

(i) $R\subset\Delta$ if and only if the following $(n-1)$ conditions hold:

$$\zeta_i - h_i \geq \zeta_{i+1} + h_{i+1}, \quad \forall\, i=1,\ldots,n-1 \tag{20}$$

(ii) $R\cap\Delta\neq\emptyset$ if and only if there exists $\sigma_2,\ldots\sigma_{n-1}\in S$ such that (2^{n-2} possibilities)

$$\zeta_1 + h_1 \geq \zeta_2 + \sigma_2 h_2 \geq \cdots \geq \zeta_{n-1} + \sigma_{n-1} h_{n-1} \geq \zeta_n - h_n. \tag{21}$$

Proof. Let us notice first that for any $x=(x_1,\ldots,x_n)\in\Delta\subset[0,1]^n$ there holds

$$1 \geq x_1 \geq \cdots \geq x_n \geq 0,$$

whereby

$$0 \leq \zeta_i - h_i \leq x_i \leq \zeta_i + h_i \leq 1, \quad i=1,\ldots,n. \tag{22}$$

(i) If $R\subset\Delta$, then all vertices of hyperrectangle R are contained in the region Δ, i.e. there holds

$$\zeta_1 + \sigma_1 h_1 \geq \cdots \geq \zeta_n + \sigma_n h_n \quad \forall \sigma_1,\ldots,\sigma_n \in S,$$

from where follows (20).

Conversely, if (20) holds, let us show that then $R \subset \Delta$. Let $x = (x_1, \ldots, x_n) \in R$ be an arbitrary point from R. Because of (20) and (22) for all $i = 1, \ldots, n-1$ there holds

$$x_i \overset{(22)}{\geq} \zeta_i - h_i \overset{(20)}{\geq} \zeta_{i+1} + h_{i+1} \overset{(22)}{\geq} x_{i+1},$$

and this means that $x \in \Delta$.

(ii) Suppose first that $R \cap \Delta \neq \emptyset$. This means that at least one vertex of R is contained in Δ, i.e. there exists $\sigma_1, \ldots, \sigma_n \in S$, such that $V(\sigma_1, \ldots, \sigma_n) \in \Delta$, i.e. the following holds:

$$\zeta_1 + \sigma_1 h_1 \geq \zeta_2 + \sigma_2 h_2 \geq \cdots \geq \zeta_{n-1} + \sigma_{n-1} h_{n-1} \geq \zeta_n + \sigma_n h_n.$$

Since $-1 \leq \sigma_i \leq 1$, we obtain

$$\zeta_1 + h_1 \leq \zeta_1 + \sigma_1 h_1 \geq \zeta_2 + \sigma_2 h_2 \geq \cdots \geq \zeta_{n-1} + \sigma_{n-1} h_{n-1} \geq \zeta_n + \sigma_n h_n \leq \zeta_n + h_n.$$

Hence there exists $\sigma_1, \ldots, \sigma_{n-1} \in S$, such that (21) holds.

Conversely, if there exists $\sigma_1, \ldots, \sigma_n \in S$, such that (21) holds, then the vertex $V(+1, \sigma_2, \ldots, \sigma_{n-1}, -1)$ is contained in Δ, so that $R \cap \Delta \neq \emptyset$.

$\square$

4 `SymDIRECT` – an algorithm for Lipschitz global optimization for a symmetric function

For the given symmetric Lipschitz function $g\colon [a,b]^n \to \mathbb{R}$ we first define the function $f = g \circ T^{-1}\colon [0,1]^n \to \mathbb{R}$, where mapping T^{-1} is given by (5).

The global minimum of the function f will be searched for in the region Δ given by (19) starting by dividing a hypercube $[0,1]^n$ in the way described in [8, 9, 12, 23]. Thereby if some hyperrectangle obtained in the process of dividing a potentially optimal hyperrectangle falls outside the region Δ, it should not be divided further. If such a hyperrectangle lies in the region Δ at least partially, it will be divided further into corresponding subhyperrectangles. Subsequently, only subhyperrectangles lying at least partially in the region Δ will be analyzed. This shall be checked by means of conditions (20)–(21). Every hyperrectangle is associated with its *"size"* defined as the length of the maximal side.

The following algorithm searches for the position in $[0,1]^n$ for the hyperrectangle $R(c, (h_1, \ldots, h_n)) \subset [0,1]^n$ with the center at the point $c = (\zeta_1, \ldots, \zeta_n)$, half side-lengths h_i in the direction of unit vectors e_i and vertices $V(\sigma_1, \ldots, \sigma_n) = (\zeta_1 + \sigma_1 h_1, \ldots, \zeta_n + \sigma_n h_n)$, where $\sigma_1, \ldots, \sigma_n \in S = \{-1, +1\}$: is R partially or completely contained in the region Δ or not contained in Δ at all.

Special attention should be paid to the optimization of checking the conditions from Theorem 1. Firstly, in the first 16 lines of the algorithm it is successively checked whether at least one of 2^{n-2} conditions from (21) is fulfilled. If none of the mentioned conditions is fulfilled, R lies outside Δ and it is not considered further. If at least one of the mentioned conditions is fulfilled, R is at least partially contained in Δ. In that case, $(n-1)$ conditions from (20) are checked in lines $19 - 24$. Only when these conditions are fulfilled, we may conclude that R is completely contained in Δ. In Algorithm 1, this procedure is optimized.

Algorithm 1 Searching for the position of hyperrectangle $R(c,(h_1,\dots,h_n))$, $c=(\zeta_1,\dots,\zeta_n)$

1. $Q := 1$, $\sigma_2 := 1$
2. **for** $i := 1$ to $n-2$ **do**
3. $\sigma_1 := \sigma_2$
4.   **if** $\zeta_i + \sigma_1 h_i \geq \zeta_{i+1} + h_{i+1}$ **then**
5. $\sigma_2 := 1$
6.   **else**
7.   **if** $\zeta_i + \sigma_1 h_i \geq \zeta_{i+1} - h_{i+1}$ **then**
8. $\sigma_2 := -1$
9.   **else**
10. $Q := 0$
11.   **exit for loop**
12. **if** $Q = 1$ **then**
13.   **if** $\zeta_{n-1} + \sigma_2 h_{n-1} \geq \zeta_n - h_n$ **then**
14. $Q := 1$
15.   **else**
16. $Q := 0$
17. **if** $Q = 0$ **then**
18.   hyperrectangle $R \not\subseteq \Delta$
19. **if** $Q = 1$ **then**
20. $T = 1$
21.   **for** $i = 1,\dots,n-1$ **do**
22.   **if** $\zeta_i - h_i < \zeta_{i+1} + h_{i+1}$ **then**
23. $T = 0$
24.   **exit for loop**
25. **if** $T = 0$ **then**
26.   condition (20) is not fulfilled: hyperrectangle R is partially in Δ
27. **if** $T = 1$ **then**
28.   condition (20) is fulfilled: hyperrectangle $R \subset \Delta$

Suppose in some step of the iterative process we have at our disposal a certain number of hyperrectangles grouped according to the size of hyperrectangles, and in every group hyperrectangles are sorted according to the value of the function in their centers. Note that the partition of these groups corresponds to the partition of the points T_i shown in Fig. 2. From every group we separate a hyperrectangle with the smallest function value of the center constructing in that way an *expanded set of potentially optimal hyperrectangles* $\mathcal{E}$. Applying Lemma 1, from this set we separate a set of potentially optimal hyperrectangles $\mathcal{P}$. All hyperrectangles from the set $\mathcal{P}$ are divided further, and the global miminum (being searched for) is updated using the hyperrectangles in the set $\mathcal{P}$, having centers in the region Δ. As we have mentioned earlier in Section 2, there remains the problem of defining a meaningful stopping criterion (see, e.g., [7, 11, 23, 42]. Except for some special situations where the global optimum is known (see Example 3, Example 4, and Section 5), our `SymDIRECT` algorithm will generally be stopped when either the size of a hyperrectangle in the division procedure becomes smaller than some number $\eta > 0$ given in advance or a maximum number of iterations (i_{max}) is performed.

Algorithm 2 (SymDIRECT)

Let c_{min} be the center of hypercube $[0,1]^n$, $d = 0.5$, $0 < \eta < d$, $f_{min} = f(c_{min})$, $i_{max} \geq 1$;
for $iter = 1, \ldots, i_{max}$ **do**
  Let $\mathcal{H}$ be the set of current hyperrectangles which completely or only partially lies in Δ;
  Group all hyperrectangles from $\mathcal{H}$ on the basis of their size, and within each group sort hyperrectangles according to the function values of their centers;
  Hyperrectangles with a smaller function value from each group form an expanded set of potentially optimal hyperrectangles $\mathcal{E}$;
  According to Lemma 1 from the set $\mathcal{E}$, form a set of potentially optimal hyperrectangles $\mathcal{P}$;
  for $R \in \mathcal{P}$ **do**
    divide R into subhyperrectangles $r_1, \ldots, r_s$;
    drop hyperrectangle R
    for $i = 1, \ldots, s$ **do**
      apply Algorithm 1 to r_i
      if $r_i \nsubseteq \Delta$ **then**
        drop r_i
      else
        determine the size d_i, center c_i and function value $f(c_i)$;
        update minimal size d and and $(c_{min}, f(c_{min}))$
      end if
    end for
  end for
  if $d \leq \eta$, STOP
end for

4.1 Some remarks on the convergence of the algorithm

The convergence of the DIRECT algorithm is proved in [8, 9]. Various modifications of this algorithm (see, e.g., [5, 30, 31]) by convergence analysis rely on or directly refer to the mentioned proof of convergence of the DIRECT algorithm.

Our modification of the DIRECT algorithm is adopted to solving the global optimization problem for a symmetric Lipschitz continuous function. Thereby, in the process of dividing the unit hypercube $[0,1]^n$ only hyperrectangles that are completely or only partially contained in the region Δ are considered. Therefore the convergence properties of the DIRECT algorithm can be directly applied to our case.

5 Numerical implementation of the method and application to data clustering

The SymDIRECT global optimization method for a symmetric function is compared with the standard DIRECT global optimization method by using an open-source MATLAB implementation by Finkel [7] (see also [11]) and with the Firefly heuristic algorithm [52], which has been shown to be superior to both *Particle Swarm Optimisation Algorithm* and *Genetic Algorithm* in terms of both efficiency and success rate[3]. First, the aforementioned methods will be tested on several standard test functions, and after that their efficiency will be compared with respect to solving a center-based clustering problem for the data that have only one feature.

5.1 Testing on standard test functions

Some standard test functions available on http://www.geatbx.com/docu/fcnindex-01.html (see also [1, 7, 11]) are often used for testing global optimization methods. First, we have selected several symmetric functions with two independent variables and the known global minimum $f^\star$. Thereby we use the standard stopping criteria based on percent error [11] $100\frac{f_{min}-f^\star}{|f^\star|} < \rho$ if $f^\star \neq 0$ and $100 f_{min} < \rho$ if $f^\star = 0$. We use $\rho = 0.01$ because it is the value originally used by [23].

(a) $f(x_1,x_2) = x_1x_2^2 + x_2x_1^2 - x_1^3 - x_2^3$ (Alolyan's function)
$x_i \in [-1,1]$, $x^\star = (-\frac{1}{3}, 1)$, $f^\star = -\frac{32}{27} = -1.18519$.

(b) $f(x_1,x_2) = -\cos x_1 \cos x_2\, e^{-((x_1-\pi)^2+(x_2-\pi)^2)}$ (Easom's function)
$x_i \in [-30,30]$, $x^\star = (\pi,\pi)$, $f^\star = -1$,

(c) $f(x_1,x_2) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$ (Rastrigin's function)
$x_i \in [-5,5]$, $x^\star = (0,0)$, $f^\star = 0$,

[3] All the experiments were executed on a computer with a 2.00 GHz Intel Pentium Core 2 Duo CPU with 4GB of RAM.

$$(d) \quad f(x_1,x_2) = \left(\sum_{j=1}^{5} j\cos((j+1)x_1+j)\right)\left(\sum_{j=1}^{5} j\cos((j+1)x_2+j)\right) \text{ (Shubert's function)}$$

$x_i \in [-10,10]$. Attains global minimum $f^\star = -186.72154$ at 18 different points.

Since these are simple test functions, all algorithms found the required global minimum fast and easily. Table 1 shows the required number of function evaluations necessary by `SymDIRECT`, `DIRECT` and the `Firefly heuristic algorithm`. It is shown that `SymDIRECT` requires significantly less function evaluations than other algorithms.

Methods	Alolyan	Easom	Rastrigin	Shubert
`SymDIRECT`	107	867	347	1 573
`DIRECT`	481	6 965	379	2 967
`Firefly`	360	387 200	9 100	56 760

Table 1: Number of function evaluations

The method was also tested on the test function with several independent variables [31]

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n}(x_ix_j + a\cos x_i\cos x_j), \quad x_i \in [-5,5]. \tag{23}$$

Testing was conducted for $n = 2,\ldots,7$. `The SymDIRECT` algorithm shows significantly better results compared to the standard `DIRECT` algorithm, what is illustrated in Fig. 10 by the necessary number of function evaluations.

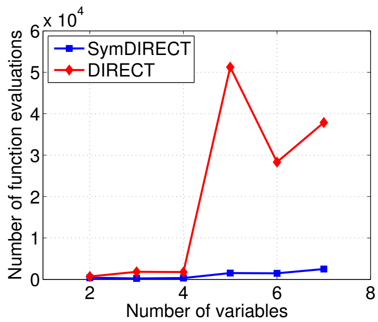

Figure 10: The number of function evaluations for the test function (23)

5.2 Center-based clustering problem for the data having only one feature

Our method will also be tested on the center-based clustering problem for the data that have only one feature (see Example 2). The data are constructed similarly to [37]. In interval $\mathcal{I} = [0, 100]$, we choose k centers $c_1, \ldots, c_k \in \mathcal{I}$ at random. The data set $\mathcal{A}$ containing $m = 100$ randomly chosen real numbers from the interval $\mathcal{I}$ is generated in the following way:

(i) let $i_1, \ldots, i_k$ be randomly generated integers such that $\sum_{s=1}^{k} i_s = m$;

(ii) in the neighborhood of the center c_s we generate a set A_s, which consists of i_s random real numbers from $\mathcal{N}(c_s, 10)$;

(iii) $\mathcal{A} = \bigcup_{s=1}^{k} A_s$.

The data set $\mathcal{A} = \{a_1, \ldots, a_m\}$ will be partitioned into $1 \le k \le m$ nonempty disjoint clusters $\pi_1, \ldots, \pi_k$ by solving center-based clustering problem (2), where $d(x, y) = |x - y|$. Let $c_1^\star, \ldots, c_k^\star \in \mathcal{I}$ be the reconstructed centers obtained in the following way.

(i) Applying `SymDIRECT` for solving global optimization problem (2) for the data set $\mathcal{A}$ and with some accuracy $\eta > 0$ we obtain $\hat{c}_1, \ldots, \hat{c}_k \in \mathcal{I}$;

(ii) Applying the k-means algorithm (see, e.g., [24, 29, 49]) with the initial approximation $\hat{c}_1, \ldots, \hat{c}_k$ we obtain reconstructed centers $c_1^\star, \ldots, c_k^\star \in \mathcal{I}$.

The `SymDIRECT` global optimization method for a symmetric function is compared with the standard `DIRECT` global optimization method and the `Firefly heuristic algorithm` when solving global optimization problem (2) with the data set $\mathcal{A}$. For the given initial approximation $(c_1^{(0)}, \ldots, c_k^{(0)})$, the mentioned algorithms generate the sequence of approximations $(c_1^{(i)}, \ldots, c_k^{(i)})$, $i = 1, 2, \ldots$ as long as the following stopping criterion is not fulfilled:

$$\max_{1 \le s \le k} |c_s^{(i)} - c_s^\star| < \alpha,$$

whereby we use $\alpha = 0.01$.

Methods	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
`SymDIRECT`	0 : 0 : 02	0 : 0 : 06	0 : 00 : 27	00 : 15 : 28	0 : 20 : 45
`DIRECT`	0 : 0 : 08	0 : 1 : 23	0 : 13 : 57	16 : 47 : 13	34 : 45 : 32
`Firefly`	0 : 3 : 54	0 : 6 : 23	0 : 06 : 37	00 : 10 : 42	0 : 08 : 27

Table 2: `CPU time (hh:mm:ss)`

For $k = 3, 4, 5, 6, 7$, Table 2 and Table 3 show the required `CPU` time (in sec) and the required number of function evaluations, respectively.

Similar evaluations were carried out for different values of α by using the `DIRECT` and the `SymDIRECT` algorithm. Numerical results are compared according to the necessary CPU time (in sec) and the number of function evaluations. Both criteria show the superior

Methods	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
`SymDIRECT`	869	3 091	7 513	108 773	214 341
`DIRECT`	5 097	50 861	160 189	1 142 959	2 012 589
`Firefly`	610 380	939 000	1 011 120	1 611 160	1 279 980

Table 3: Number of function evaluations

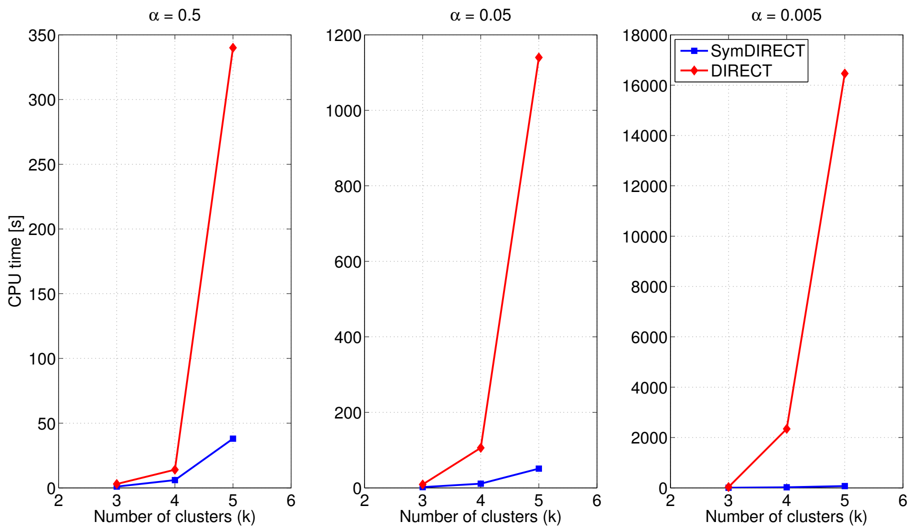

Figure 11: Necessary CPU time (in sec) for solving the center-based clustering problem

efficiency of our algorithm. For example, Fig. 11 shows the necessary CPU time (in sec) depending on the number of clusters k in both algorithms and for various values of α.

Based upon the results given in Table 2 and Table 3 and Fig. 11, superiority of the proposed `SymDIRECT` global optimization method for a symmetric function is shown: the number of function evaluations is significantly less than by the other two compared algorithms, and the `CPU` time is considerably shorter than by the standard `DIRECT` algorithm. What is unusual at first sight is that the `CPU` time needed by the `Firefly` heuristic algorithm decreases with an increase in the number of clusters. This effect occurs due to the character of the algorithm itself and the fact that an increase in the number of clusters causes the number of points in the hypercube $[0, 1]^n$ where the global minimum is attained to increase significantly. For example, for $k = 7$, the global minimum of the corresponding functional (2) is attained at as many as $7! = 5040$ points from the hypercube $[0, 1]^7$.

Let us mention that comparative data for `SymDIRECT` in Table 2 and Table 3 could be even better if some procedures in the corresponding `MATLAB` software were optimized.

6 Conclusions

A global optimization problem for a symmetric Lipschitz function is often found in various applications. It was shown that the proposed SymDIRECT method is efficient when it comes to solving such problems. It should be especially stressed that this method can be used efficiently for solving complex center-based clustering problems for the data having only one feature.

It would be very interesting to expand this method to functions that are symmetric in two or more vector variables (n-tuples). Such functions appear in characterizations of zero derivative points (see [55, 56]).

In addition, it could be analyzed in a similar way how some other known methods of global optimization could be adapted for the case of a symmetric function. For example, we could try to conduct such analysis for the αBB method [10, 34], the Generating Set Search algorithm [16, 25], the Multilevel Coordinate Search method [19] or global optimization methods based on response surfaces [22].

7 Acknowledgments

The authors would like to thank Prof. Sanjo Zlobec (McGill University, Montreal, Canada) and Prof. Dragan Jukić and Prof. Kristian Sabo (University of Osijek, Croatia) for their useful comments and remarks. We are also thankful to anonymous referees and journal editors for their careful reading of the paper and insightful comments that helped us improve the paper. This work is supported by the Ministry of Science, Education and Sports, Republic of Croatia, through research grants 235-2352818-1034 and 165-0361621-2000.

References

[1] I. Alolyan, *A new exclusion test for finding the global minimum*, Journal of Computational and Applied Mathematics, **200**(2007) 491 – 502.

[2] A. Auger, B. Doerr, *Theory of Randomized Search Heuristics*, Volume 1 of *Theoretical Computer Science*, World Scientific, Danvers, 2011.

[3] L. Chiter, *Direct algorithm: A new definition of potentially optimal hyperrectangles*, Applied Mathematics and Computation, **179**(2006) 742–749.

[4] L. Chiter, *A new sampling method in the direct algorithm*, Applied Mathematics and Computation, **175**(2006) 297–306.

[5] D. di Serafino, G. Liuzzi, V. Piccialli, F. Riccio, G. Toraldo, *A modified dividing rectangles algorithm for a problem in astrophysics*, J Optim Theory Appl, **151**(2011) 175–190.

[6] Y. G. Evtushenko, *Numerical Optimization Techniques (Translations Series in Mathematics and Engineering)*, Springer-Verlag, Berlin, 1985.

[7] D. E. Finkel, *DIRECT Optimization Algorithm User Guide*, Center for Research in Scientific Computation. North Carolina State University, 2003, `http://www4.ncsu.edu/definkel/research/index.html`.

[8] D. E. Finkel, C. T. Kelley, *Convergence analysis of the direct algorithm crsc-tr04-28*, Technical Report, Center for Research in Scientific Computation, North Carolina State University, 2004.

[9] D. E. Finkel, C. T. Kelley, *Additive scaling and the* ***DIRECT*** *algorithm*, Journal of Global Optimization, **36**(2006) 597–608.

[10] C. A. Floudas, C. E. Gounaris, *A review of recent advances in global optimization*, Journal of Global Optimization, **45**(2009) 3–38.

[11] J. M. Gablonsky, *Direct version 2.0*, Technical report, Center for Research in Scientific Computation. North Carolina State University, 2001.

[12] J. M. Gablonsky, *Modifications of the DIRECT Algorithm*, Ph.D. thesis, North Carolina State University, 2001.

[13] J. M. Gablonsky, C. T. Kelley, *A locally-biased form of the direct algorithm*, Journal of Global Optimization, **21**(2001) 27–37.

[14] G. Gan, C. Ma, J. Wu, *Data clustering: theory, algorithms, and applications*, SIAM, Philadelphia, 2007.

[15] M. Gaviano, D. Lera, *A global minimization algorithm for Lipschitz functions*, Optimization Letters, **2**(2008) 1–13.

[16] J. D. Griffin, T. G. Kolda, *Asynchronous parallel hybrid optimization combining* ***DIRECT*** *and* ***GSS***, Optimization Methods and Software, **25**(2010) 797–817.

[17] E. Hansen, G. W. Walster, *Global optimization using interval analysis*, Marcel Dekker, New York, 2004, 2nd edition.

[18] R. Horst, P. M. Pardalos, editors, *Handbook of Global Optimization*, Volume 1, Kluwer Academic Publishers, Dordrecht, 1995.

[19] W. Huyer, A. Neumaier, *Global optimization by multilevel coordinate search*, Journal of Global Optimization, **14**(1999) 331–355.

[20] C. Iyigun, *Probabilistic distance clustering*, Ph.D. thesis, Graduate School – New Brunswick, Rutgers, 2007.

[21] C. Iyigun, A. Ben-Israel, *A generalized weiszfeld method for the multi-facility location problem*, Operations Research Letters, **38**(2010) 207–214.

[22] D. R. Jones, *A taxonomy of global optimization methods based on response surfaces*, Journal of Global Optimization, **21**(2001) 345–383.

[23] D. R. Jones, C. D. Perttunen, B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, Journal of Optimization Theory and Applications, **79**(1993) 157–181.

[24] J. Kogan, *Introduction to clustering large and high-dimensional data*, Cambridge University Press, 2007.

[25] T. G. Kolda, R. M. Lewis, V. Torczon, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Review, **45**(2003) 385–482.

[26] D. E. Kvasov, Y. D. Sergeyev, *A univariate global search working with a set of Lipschitz constants for the first derivative*, Optimization Letters, **3**(2009) 303–318.

[27] D. E. Kvasov, Y. D. Sergeyev, *Univariate geometric Lipschitz global optimisation algorithms*, Numerical Algebra, Control and Optimization, **2**(2012) 69–90.

[28] D. E. Kvasov, Y. D. Sergeyev, *Lipschitz gradients for global optimization in a one-point-based partitioning scheme*, Journal of Computational and Applied Mathematics, **236**(2012), 4042–4054.

[29] F. Leisch, *A toolbox for k-centroids cluster analysis*, Computational Statistics & Data Analysis, **51**(2006) 526–544.

[30] G. Liuzzi, S. Lucidi, V. Piccialli, *A direct-based approach for large-scale global optimization problems*, Computational Optimization and Applications, **45**(2010) 353–375.

[31] J. Mockus, *On the pareto optimality in the context of lipschitzian optimization*, Informatica, **22**(2011) 524–536.

[32] A. Neumaier, *Complete search in continuous global optimization and constraint satisfaction*, In: *Acta Numerica*. Cambridge University Press, 2006, 271–369.

[33] E. K. Nyarko, R. Scitovski, *Solving the parameter identification problem of mathematical model using genetic algorithms*, Applied Mathematics and Computation, **153**(2004) 651–658.

[34] P. M. Pardalos, T. F. Coleman, editors, *Lectures on Global Optimization*. AMS, 2009.

[35] S. A. Pijavskij, *An algorithm for searching for a global minimum of a function*, USSR Computational Mathematics and Mathematical Physics, **12**(1972) 888–896, (in Russian).

[36] J. D. Pintér, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)*, Kluwer Academic Publishers, Dordrecht, 1996.

[37] K. Sabo, R. Scitovski, I. Vazler, *One-dimensional center-based l_1-clustering method*, Optimization Letters, (accepted) DOI: 10.1007/s11590–011–0389–9.

[38] A. Schöbel, D. Scholz, *The big cube small cube solution method for multidimensional facility location problems*, Computers & Operations Research, **37**(2010) 115–122.

[39] Y. D. Sergeyev, D. E. Kvasov, *Diagonal Global Optimization Methods*, FizMatLit, Moscow, 2008, (in Russian).

[40] Y. D. Sergeyev, D. E. Kvasov, *Lipschitz global optimization*, In: J. Cochran, editor, *Wiley Encyclopedia of Operations Research and Management Science*, Volume 4, Wiley, New York, 2011, 2812–2828.

[41] Y. D. Sergeyev, D. Famularo, P. Pugliese, *Index branch-and-bound algorithm for Lipschitz univariate global optimization with multiextremal constraints*, Journal of Global Optimization, **21**(2001) 317–341.

[42] Y. D. Sergeyev, D. E. Kvasov, *Global search based on efficient diagonal partitions and a set of Lipschitz constants*, SIAM Journal on Optimization, **16**(2006) 910 – 937.

[43] B. Shubert, *A sequential method seeking the global maximum of a function*, SIAM Journal on Numerical Analysis, **9**(1972) 379–388.

[44] H. Späth, *Cluster-Formation und Analyse*, R. Oldenburg Verlag, München, 1983.

[45] R. G. Strongin, *Numerical Methods in Multiextremal Problems*, Nauka, Moscow, 1978, (in Russian).

[46] R. G. Strongin, Y. D. Sergeyev, *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*, Kluwer Academic Publishers, Dordrecht, 2000.

[47] M. Teboulle, *A unified continuous optimization framework for center-based clustering methods*, Journal of Machine Learning Research, **8**(2007) 65–102.

[48] R. J. Vanderbei, *Extension of piyavskii's algorithm to continuous global optimization*, Journal of Global Optimization, **14**(1999) 205–216.

[49] V. Volkovich, J. Kogan, C. Nicholas, *Building initial partitions through sampling techniques*, European Journal of Operational Research, **183**(2007) 1097–1105.

[50] G. R. Wood, B. P. Zhang, *Estimation of the Lipschitz constant of a function*, Journal of Global Optimization, **8**(1996) 91–103.

[51] Y. Wu, L. Ozdamar, A. Kumar, *Triopt: a triangulation-based partitioning algorithm for global optimization*, Journal of Computational and Applied Mathematics, **177**(2005) 35–53.

[52] X. S. Yang, *Firefly algorithms for multimodal optimization*, In: *Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications.* 2009, 169–178.

[53] Y. Zhang, Y. Xua, L. Zhang, *A filled function method applied to nonsmooth constrained global optimization*, Journal of Computational and Applied Mathematics, **232**(2009) 415–426.

[54] A. Zhigljavsky, A. Žilinskas, *Stochastic Global Optimization*, Springer, 2008.

[55] S. Zlobec, *The fundamental theorem of calculus for Lipschitz functions*, Mathematical Communications, **13**(2008) 215–232.

[56] S. Zlobec, *Equivalent formulations of the gradient*, J. Glob. Optim., **50**(2011) 549-553