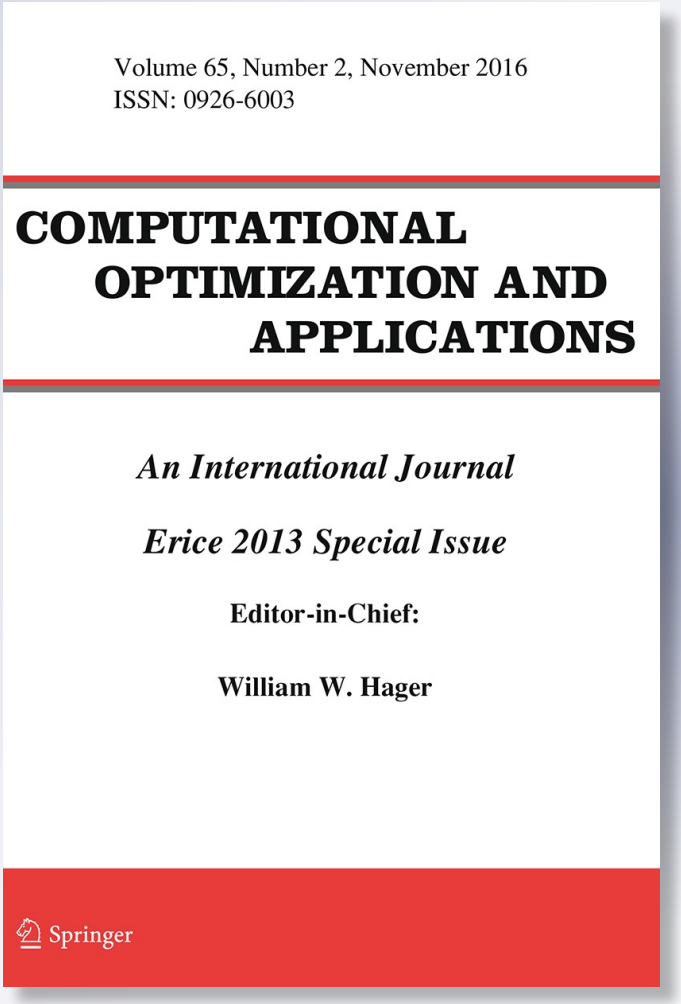
Exploiting derivative-free local searches in DIRECT*-type algorithms for global optimization*

G. Liuzzi, S. Lucidi & V. Piccialli

Computational Optimization and Applications
An International Journal

ISSN 0926-6003
Volume 65
Number 2

Comput Optim Appl (2016) 65:449-475
DOI 10.1007/s10589-015-9741-9



Springer

Comput Optim Appl (2016) 65:449–475
DOI 10.1007/s10589-015-9741-9

Exploiting derivative-free local searches in DIRECT-type algorithms for global optimization

G. Liuzzi[1] · **S. Lucidi** [2] · **V. Piccialli**[3]

Received: 28 February 2014 / Published online: 18 March 2015

Abstract In this paper we consider bound constrained global optimization problems where first-order derivatives of the objective function can be neither computed nor approximated explicitly. For the solution of such problems the DIRECT algorithm has been proposed which has a good ability to locate promising regions of the feasible domain and convergence properties based on the generation of a dense set of points over the feasible domain. However, the efficiency of DIRECT deteriorates as the dimension and the ill-conditioning of the objective function increase. To overcome these limits, we propose DIRECT-type algorithms enriched by the efficient use of derivative-free local searches combined with nonlinear transformations of the feasible domain and, possibly, of the objective function. We report extensive numerical results both on test problems from the literature and on an application in structural proteomics.

Keywords Global optimization · DIRECT-type algorithms · Local minimizations

✉ G. Liuzzi
giampaolo.liuzzi@iasi.cnr.it

S. Lucidi
lucidi@dis.uniroma1.it

V. Piccialli
veronica.piccialli@uniroma2.it

1 Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", CNR, via dei Taurini 19, 00185 Rome, Italy

2 Dipartimento di Ingegneria Informatica Automatica e Gestionale "A. Ruberti", "Sapienza" Università di Roma, via Ariosto 25, 00185 Rome, Italy

3 Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università degli Studi di Roma "Tor Vergata", viale del Politecnico 1, 00133 Rome, Italy

Mathematics Subject Classification 65K05 · 90C26

1 Introduction

In the paper we refer to the following general problem

$$\min_{x \in D} f(x), \tag{1}$$

where $D = \{x \in \Re^n : 0 \leq x_i \leq 1, i = 1, \ldots, n\}$, to which every box-constrained problem can be reduced.

When the objective function is Lipschitz-continuous, Problem (1) can be solved by means of the DIRECT (DIvide Rectangles) algorithm [1]. This algorithm is based on a space-partitioning technique which is designed to adaptively balance local and global search at each iteration. Convergence of the DIRECT algorithm to the global minimum of Problem (1) is guaranteed by the so-called everywhere dense property, that is DIRECT is able to generate a set of points which, in the limit, becomes dense in the feasible set [1–3].

The original DIRECT algorithm has some known weaknesses. First of all, in practice, it is hard to develop some efficient stopping condition other than, e.g., exhaustion of some preset computational resources.

Furthermore, DIRECT is typically quite fast in getting close to the global optimum, but it can be slow to converge to the solution with an high accuracy. In fact, it can often waste a lot of time exploring uninteresting regions of the feasible domain, thereby delaying the discovery of global minima.

Various attempts have been done in the literature to overcome this latter source of inefficiency. For example, different techniques for adaptive setting the DIRECT balancing parameter can be applied as reviewed, e.g., in [4]; transformations of the search domain can be considered [3,5,6]; smart schemes for balancing local and global information within the same global optimization procedure can be introduced [4,7–9]; hybridization of a DIRECT-type method by cleverly invoking separate local optimizers during the global optimization process is also very useful [2,3,10,11]. The two latter approaches are particularly beneficial for solving multiextremal large-scale problems.

More in detail, in [9] a locally-biased version of the DIRECT method is presented for solving low-dimensional problems with a few local minimizers, which tries to capture the best local minimum by subdividing less sub-regions at each iteration with respect to the DIRECT algorithm.

In [7,8] a strategy inspired by bilevel (or multilevel) methods for solving large scale algebraic systems arising from the discretization of partial differential equations is adopted to overcome the slow convergence of DIRECT.

In [4], a two-phase approach having proved to be efficient in the DIRECT-based schemes [12] is successfully adopted and experimentally investigated for the DIRECT simplicial partition algorithm (originally proposed in [13,14]).

We consider in detail Algorithm DIRMIN-TL from Ref. [3] where both local minimizations and nonlinear transformations of the variables are introduced in DIRECT.

The main contribution of the paper is the proposal of three new deterministic algorithms for black-box derivative-free global optimization. Drawing inspiration from [3], we present new variants of the DIRMIN-TL algorithm for black-box optimization to try and enhance both its efficiency (i.e., number of local searches to get convergence) and robustness (i.e., ability to find the global optimum of problem (1) within a prescribed number of iterations). The basic idea consists in exploiting as much as possible derivative-free local minimizations. This can be done mainly in two ways. First, we can substitute the local minimization step of algorithm DIRMIN-TL from Ref. [3] with derivative-free local minimization. This plain modification results in a quite reliable algorithm. Then, in order to improve the efficiency of the method, we propose to carry out the local minimizations in a "balanced" way. Second, drawing inspiration from [15–17], we can use the derivative-free local minimization routine to modify the objective function. It should be noted that all of the proposed modifications are also suitable to other more sophisticated partition-based algorithms, like for instance those based on simplicial or diagonal partitioning (see, e.g., [12–14,18]).

In Sect. 2 we present a simple adaptation of Algorithm DIRMIN-TL, from Ref. [3], to derivative-free optimization and present its numerical performances on a benchmark of difficult global optimization problems. In Sect. 3 we propose a "balanced" version of algorithm DIRMIN-TL where the local minimization are carried out in a "balanced" way thus considerably improving the efficiency of DIRMIN-TL. In Sect. 4 we propose a new DIRECT-type algorithm based on the so-called "plateau" transformation of the objective function which considerably improves the robustness of DIRMIN-TL. In Sect. 5 we present an application of the the latter algorithm to a protein structural alignment problem [19]. Finally, in Sect. 6 we draw some conclusions.

1.1 The original DIRECT algorithm

In this section we report a brief description of the original DIRECT algorithm. At the first step of DIRECT, $f(x)$ is evaluated at the center of the search domain D; the hypercube is then partitioned into a set of smaller hyperrectangles and $f(x)$ is evaluated at their centers. Let the partition of $\mathcal{D}$ at iteration k be defined as

$$\mathcal{H}_k = \{\mathcal{D}^i : i \in I_k\}, \text{ with } \mathcal{D}^i = \{x \in \Re^n : l^i \leq x \leq u^i\}, \ \forall\, i \in I_k,$$

where $l^i, u^i \in [0, 1], i \in I_k$, and I_k is the set of indices identifying the subsets defining the current partition.

At the generic kth iteration of the algorithm, starting from the current partition $\mathcal{H}_k$ of D into hyperrectangles, a new partition, $\mathcal{H}_{k+1}$, is built by subdividing a set of *potentially optimal* hyperrectangles of the previous partition $\mathcal{H}_k$. The identification of a potentially optimal hyperectangle is based on some measure of the hyperrectangle itself and on the value of f at its center. The refinement of the partition continues until a prescribed number of function evaluations has been performed, or another stopping criterion is satisfied (see [4,12,20]). The minimum of f over all the centers of the

$\mathcal{H}_1 = \{D\}$, $c =$ center of D, $f_{min} = f(c)$, $X_{min} = \{c\}$, $k = 1$

repeat

identify the set of indices $I_k^* \subseteq I_k$ of the potentially optimal hyperrectangles in $\mathcal{H}_k$

for each $i \in I_k^*$, subdivide $\mathcal{D}^i$ (generate the new partition $\mathcal{H}_{k+1}$)

evaluate f in the centers of the new hyperrectangles

$f_{min} = \min\{f(c) : c \in C_k\}$, $X_{min} = \{c \in C_k : f(c) = f_{min}\}$, $k = k + 1$
($C_k = \{$centers of the hyperrectangles in $\mathcal{H}_k\}$)

until (stopping criterion satisfied)

return f_{min}, X_{min}

Fig. 1 Sketch of the original DIRECT algorithm

final partition, and the corresponding centers, provide an approximate solution to the problem. The structure of DIRECT is outlined in Fig. 1.

For the sake of completeness, we recall that the new partition $\mathcal{H}_{k+1}$ is obtained by subdividing the potentially optimal hyperintervals according to a specific partition procedure. In particular, for every potentially optimal hyperinterval $\mathcal{D}^i$ the set J of the longest edges is computed, and every potentially optimal hyperinterval is subdivided along all the edges belonging to J. For each edge belonging to J, two new points are generated with symmetric distances from the centroid x^i, equal to one third of the length of the edges in J. The generated points are the centroids of $2m$ new hyperintervals, where $m = |J|$. If there are multiple longest edges, the division order is chosen in such a way that the biggest hyperrectangles contain the best objective function values.

Further details on the original DIRECT algorithm can be found in [1,3]. The convergence of DIRECT is proved (see, e.g., [1,3]) by showing that the set of sampled points becomes everywhere dense in D as the number of iterations k goes to infinity. For a convergence analysis of DIRECT-type, as other "divide-the-best" algorithms, we refere the interested reader to [12,18,20,21].

1.2 Algorithm DIRMIN-TL

In order to describe Algorithm DIRMIN-TL from [3], we first need to introduce a sketch of Algorithm DIRMIN (from Ref. [3]) where local minimizations starting from the centroids of potentially optimal hyperintervals are introduced in the DIRECT algorithm.

Algorithm DIRMIN

$\mathcal{H}_1 = \{D\},\ c =$ center of $D,\ f_{min} = f(c),\ X_{min} = \{c\}, tol, kmax,\ k = 1$

Repeat

(S.1) identify the potentially optimal hyperrectangles $\mathcal{P}_k$ in $\mathcal{H}_k$

(S.2) for all centroids c^i of hyperrectangles in $\mathcal{P}_k$ perform a local minimization and record the best function value f_{ml}

(S.3) subdivide the potentially optimal hyperrectangles to build a new partition $\mathcal{H}_{k+1}$

(S.4) evaluate f in the centers of the new hyperrectangles

(S.5) $f_{min} = \min\{f(c) : c \in C_k, f_{ml}\},\ X_{min} = \{x \in D : f(x) = f_{min}\},\ k = k + 1$

C_k is the set of centroids c of the hyperrectangles in $\mathcal{H}_k$

Until (stopping criterion satisfied)

Return f_{min}, X_{min}

Algorithm DIRMIN-TL is obtained by repeatedly applying DIRMIN to the problem obtained from Problem (1) by transforming the search space by means of the following piecewise linear transformation of variables: given a point $\tilde{x} \in (0, 1)^n$, let $y = T_{\tilde{x}}(x)$ be defined by

$$y_i = (T_{\tilde{x}}(x))_i = \begin{cases} \dfrac{x_i}{2\tilde{x}_i} & \text{if } \ x_i \le \tilde{x}_i, \\ \dfrac{1 - x_i}{2(\tilde{x}_i - 1)} + 1 & \text{if } \ x_i > \tilde{x}_i, \end{cases} \qquad i = 1, \ldots, n.$$

As reported in [3], operator $T_{\tilde{x}} : \Re^n \to \Re^n$ is invertible, maps $[0, 1]^n$ into $[0, 1]^n$, maps the point $\tilde{x}$ into the centroid of the transformed space ($T_{\tilde{x}}(\tilde{x}) = (1/2 \ldots 1/2)^\top$) and reduces to the identity if $\tilde{x} = (1/2 \ldots 1/2)^\top$.

Thus, given $\tilde{x} \in (0, 1)^n$ and by using operator $T_{\tilde{x}}$, we can write

$$f(x) = f(T_{\tilde{x}}^{-1}(y)) = f_{\tilde{x}}(y).$$

After a fixed maximum number of partitioning steps, DIRMIN produces an estimate $x_{\min} \in (0, 1)^n$ of the global minimum point. Then, as proposed in [3], we use the above transformation $T_{\tilde{x}}$ with $\tilde{x} = x_{\min}$ and apply again DIRMIN to the problem

$$\min_{y \in [0,1]^n} f_{\tilde{x}}(y). \tag{2}$$

DIRMIN applied to Problem (2) will try to improve the current estimate of the global minimum point by generating a different partition of the domain $[0, 1]^n$. This process is reiterated if DIRMIN improves on the initial point $\tilde{x}$. Otherwise, DIRMIN is restarted by choosing $\tilde{x}$ among the set of promising stationary points produced in the previous iteration, which is updated during the iterations of the new algorithm.

We report below the sketch of Algorithm DIRMIN-TL.

Algorithm DIRMIN-TL

$x_{min} = \tilde{x} = (1/2 \dots 1/2)^\top$, $f_{min} = f(x_{min})$, $\mathcal{N} = \emptyset$, $\mathcal{O} = \emptyset$, $maxint \gg 0$

Repeat

(S.1) Apply DIRMIN to Problem (2) until $maxint$ hyperintervals have been generated and let $\hat{x}$ be the best point produced and $\mathcal{W}$ be the set of "promising" stationary points.

(S.2) If ($f_{\tilde{x}}(\hat{x}) < f_{min}$) then set $f_{min} = f_{\tilde{x}}(\hat{x})$, $x_{min} = \tilde{x} = \hat{x}$, $\mathcal{N} = \emptyset$, $\mathcal{O} = \emptyset$ and **go to** Step (S.1).

(S.3) Otherwise set

$$\mathcal{N} = \mathcal{N} \cup \left\{ y \in \mathcal{W} : f(y) - f_{min} \leq \epsilon_f \quad \text{and} \quad \min_{x \in \mathcal{N} \cup \{x_{min}\}} d(y, x) > \epsilon_d \right\}.$$

(S.4) choose $\bar{x} \in \mathcal{N} \setminus \mathcal{O}$, set $\mathcal{O} = \mathcal{O} \cup \{\bar{x}\}$, $\tilde{x} = \bar{x}$.

Until ($\mathcal{N} \setminus \mathcal{O} = \emptyset$)

Return f_{min}, x_{min}

In the algorithm, $\mathcal{N}$ represents the set of candidate points to restart DIRMIN, $\mathcal{O}$ is the set of already used points, and $\mathcal{W}$ is the set of stationary points produced at Step S.2 of Algorithm DIRMIN. After DIRMIN has generated $|I_k| = maxint$ hyperintervals, $\mathcal{N}$ and $\mathcal{O}$ are updated, on the basis of the information gained up to that point. Any time $f_{\min}$ is updated, they are initialized to the empty set. Otherwise, at step S.3 the set of candidate points $\mathcal{N}$ is updated by setting $\mathcal{N} = \mathcal{N} \cup \{x \in \mathcal{W} : x$ is "promising"$\}$ where a stationary point x is considered as promising when, given two scalars $\epsilon_f, \epsilon_d > 0$, $f(x) - f_{min} \leq \epsilon_f$ and $\min_{x \in \mathcal{N} \cup \{x_{min}\}} d(y, x) > \epsilon_d$, that is x is sufficiently distant from the points in $\mathcal{N} \cup \{x_{\min}\}$. Then, the new point $\tilde{x}$ to restart DIRMIN is chosen in the set $\mathcal{N} \setminus \mathcal{O}$ and the set $\mathcal{O}$ is updated consequently: $\mathcal{O} = \mathcal{O} \cup \{\tilde{x}\}$.

2 A plain modification of algorithm DIRMIN-TL

In principle, Algorithm DIRMIN-TL (from Ref. [3]) cannot be used in the present context of derivative-free black-box global optimization since the local minimizations are carried out by means of a gradient based algorithm.

However, it is worth noting that Algorithm DIRMIN-TL, like other DIRECT-type algorithms, is able to guarantee the following convergence property provided that parameter $maxint = +\infty$.

Lemma 2.1 *For every global minimum point x^* of Problem (1) and for every $\epsilon > 0$, there exists an iteration k and a centroid $\bar{x} \in C_k$ such that $\|x^* - \bar{x}\| \leq \epsilon$.*

This property can be exploited to accelerate convergence of DIRECT-type algorithms by using suitable local minimization algorithms (which is the fundamental consideration of [3]). In particular, the local minimization algorithm should be able to converge to the global minimum point once the global optimization scheme has generated a point sufficiently close to it. To this aim, we recall from Ref. [3] the following proposition concerning some minimal assumptions needed by an iterative algorithm to be attracted by a global minimum point.

Proposition 2.1 ([22]) *Let $f \in \mathcal{C}^2$ and $\{x_k\}$ be a sequence of feasible points generated by an iterative method $x_{k+1} = x_k + \alpha_k d_k$ such that*

(i) $f(x_{k+1}) \leq f(x_k) - \theta(\alpha_k)^2 \|d_k\|^2$, for all k, where $\theta > 0$;
(ii) any accumulation point of the sequence $\{x_k\}$ is stationary for Problem (1).

For every global minimum point x^* of $f(x)$ on $\mathcal{D}$ where $\nabla^2 f(x^*)$ is positive definite, there exists an open set $\mathcal{L}$ containing x^* such that, if $x_{\bar{k}} \in \mathcal{L}$ for some $\bar{k} \geq 0$, then $x_k \in \mathcal{L}$ for all $k \geq \bar{k}$ and $\{x_k\} \to x^*$.

By using a derivative-free local minimization algorithm satisfying the assumptions of above Proposition 2.1 (like, e.g., those of Refs. [23,24]), we can thus propose straightforward modifications of `DIRMIN` and `DIRMIN-TL` (which we call `DFO-DIRMIN` and `DFO-DIRMIN-TL`, respectively) for black-box optimization that consists in substituting derivative-based with derivative-free local minimizations. Algorithm `DFO-DIRMIN-TL` obviously preserves the convergence property stated in Lemma 2.1.

Here, we represent a derivative-free local minimization algorithm as

$$(\hat{x}, \hat{\alpha}) = DF(x_0, \alpha_0, tol, kmax),$$

where x_0 is the starting point of the minimization, $\alpha_0 \in \Re^n$ is a vector of tentative steps such that $\max_{i=1,\ldots,n}\{\alpha_0^i\}$ is an estimate of the stationarity measure [25] of x_0, tol is the target measure of stationarity and $kmax$ is the maximum number of allowed iterations. In output, the algorithm produces a feasible point $\hat{x}$, and the current stepsizes $\hat{\alpha} \in \Re^n$ (a sketch of a possible algorithm DF is reported in Appendix 1 for the interested reader). We denote by $\hat{\alpha}_{\max} = \max_{i=1,\ldots,n} \hat{\alpha}_i$ the stationarity measure of $\hat{x}$ [25].

2.1 Efficient partition management in `DIRECT`

The efficiency of Algorithm `DIRECT` heavily depends on the data structures that are used to store information on the current feasible domain partition and on how the selection and partition procedures are implemented. In [26] a partly dynamic data structure has been proposed with the aim of combining an efficient management of the data structures with the efficiency of the algorithm. In our implementation of `DIRECT`, we adopt a completely dynamic data structure for box information storage (see Figure 2). We use two derived data types, `Box` and `Column`. A `Box` structure contains information on a hyperinterval, that is, the objective function value on the centroid, the centroid coordinates, the hyperbox dimensions and pointers to previous and next `Box` structures. The `Column` derived type is used to define a double-linked list of columns. Each element of the list contains the diameter of the column of hyperboxes, a pointer to the corresponding list of `Box` structures and pointers to previous and next `Column` structures. The list of columns is kept sorted by increasing diameter size, whereas all the lists of boxes are kept sorted by increasing objective function value. It is worth noting that, by exploiting the above dynamic data structure, computing the potentially optimal hyperintervals, adding,

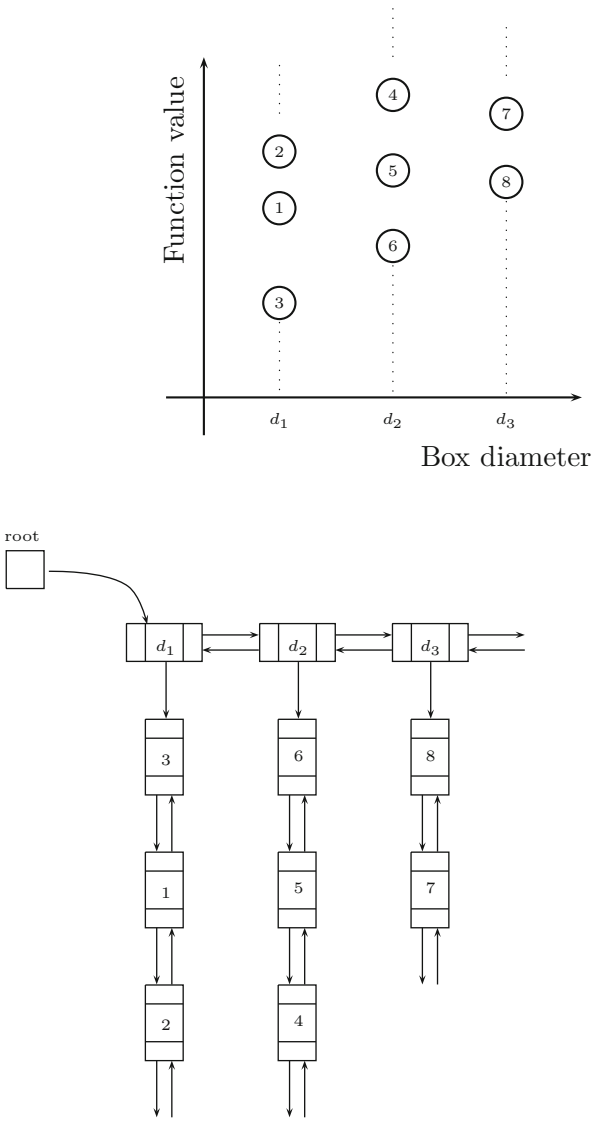


Fig. 2 Potentially optimal hyperintervals

removing and keeping columns and boxes ordered can be done very efficiently. In particular, the set of potentially optimal hyperintervals is computed by applying the Jarvis's march [27] just to the top elements of the list of boxes of each column (see Fig. 2), which is of limited cardinality. An implementation qof the above partition management for `DIRECT` algorithms combined with the use of derivative-free local searches, i.e., Algorithm `DFO-DIRMIN`, is freely available at the URL http://www.dis.uniroma1.it/~lucidi/DFL as part of the derivative-free library as package `DIRMIN`.

2.2 Numerical results with DFO-DIRMIN-TL

We begin this section by recalling that the main drawbacks of partition-based deterministic algorithms for global optimization are a possibly huge requirement in terms of memory occupancy and a possibly high execution time. Of course, the latter aspect depends both on the implementation of the algorithm and on the cost of a single function evaluation. In the paper we concentrate on the former aspect since it is the most critical one.

For this reason, we choose a set of test problems with negligible computing time per single function evaluation. Namely, our test set is composed of problems selected from Refs. [3,28,29] (see Table 5 in Appendix 2 for problem dimensions and optimal values) and is freely available at the URL http://www.dis.uniroma1.it/~lucidi/DFL as package TESTGO.

We applied Algorithm DFO-DIRMIN-TL to this set of global optimization problems and, in particular, within DFO-DIRMIN-TL we allow the generation of at most $50,000 \times n$ hyperrectangles by algorithm DFO-DIRMIN and we set the maximum number of "restarts" to 100, where by restart we mean the number of times that the *go to* instruction is executed in Step (S.2) of DFO-DIRMIN-TL.

Furthermore, we use the following stopping criterion

$$\frac{f(x_{\min}) - f^*}{\max\{1, |f^*|\}} \leq 10^{-4},$$

where f^* is sthe known optimal function value. The results are reported in the table below, where:

- Problem, is the name of the problem;
- n, is the dimension of the problem;
- $f(\bar{x})$, is the best function value produced by the algorithm, and it is in boldface whenever the stopping criterion is not met. In this case the algorithm stops either because $\mathcal{N} \setminus \mathcal{O}$ is empty or because the maximum number of restarts has been reached;
- n.f., is the number of computed function evaluations;
- n.loc., is the number of performed local minimizations;
- n.int., is the number of hyperrectangles.

We remark that the number of computed function evaluations (n.f.) is given by the sum between n.int. and the overall number of function evaluations required by the n.loc. local minimizations. It thus measures the overall computational burden of the method on a single problem. Note that, the number of executed restarts is not displayed because it is given by

$$\text{number of restarts } = \left\lfloor \frac{\text{n.int.}}{50,000 \times n} \right\rfloor.$$

In the first part of the table, we report the same test problems used in [3], in the second part of the table, we test the algorithm on a further set of problems from

the literature that can be found on the webpage [28]. Finally, the third part of the table contains the test problems recently proposed in [29] for the special session and competition on Real Parameter Single Objective Optimization at the Conference on Evolutionary Computation (CEC) 2013.

From Table 1 the following can be observed:

Table 1 Results of `DFO-DIRMIN-TL`

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Test problems from Ref. [3]					
Schubert	2	−1.8673e+02	365	5	21
Schub. pen. 1	2	−1.8673e+02	2520	32	113
Schub. pen. 2	2	−1.8673e+02	881	11	45
S-H. Camel B.	2	−1.0316e+00	75	1	5
Goldstein-Price	2	3.0000e+00	107	1	5
Treccani mod.	2	7.1314e−09	78	1	5
Quartic	2	−3.5239e−01	499	6	27
Shekel $m = 5$	4	−1.0153e+01	142	1	9
Shekel $m = 7$	4	−1.0403e+01	500	3	21
Shekel $m = 10$	4	−1.0536e+01	1005	6	33
Espon. mod.	2	−1.0000e+00	76	1	5
Espon. mod.	4	−1.0000e+00	150	1	9
Cos-mix mod.	2	−2.0000e−01	70	1	5
Cos-mix mod.	4	−4.0000e−01	138	1	9
Hartman	3	−3.8628e+00	105	1	7
Hartman	6	−3.3224e+00	229	1	13
$5n$ loc-min	2	2.3557e−31	62	1	5
$5n$ loc-min	5	9.4226e−32	152	1	11
$5n$ loc-min	10	4.7113e−32	302	1	21
$5n$ loc-min	20	2.3557e−32	602	1	41
$10n$ loc-min	2	2.3557e−31	62	1	5
$10n$ loc-min	5	9.4226e−32	152	1	11
$10n$ loc-min	10	4.7113e−32	302	1	21
$10n$ loc-min	20	2.3557e−32	602	1	41
$15n$ loc-min	2	1.3497e−32	62	1	5
$15n$ loc-min	5	1.3497e−32	152	1	11
$15n$ loc-min	10	1.3497e−32	302	1	21
$15n$ loc-min	20	1.3497e−32	602	1	41
Griewank mod.	2	1.3472e−11	78,663	858	3089
Griewank mod.	5	6.2154e−10	490,515	1896	11,553
Griewank mod.	10	9.2333e−10	411,178	753	7359
Griewank mod.	20	9.4514e−10	942	1	41
Pinter	2	3.3940e−09	170	2	7

Table 1 continued

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Pinter	5	1.5414e−06	25,618	101	629
15n loc-min	20	1.3497e−32	602	1	41
Griewank mod.	2	1.3472e−11	78,663	858	3089
Griewank mod.	5	6.2154e−10	490,515	1896	11,553
Griewank mod.	10	9.2333e−10	411,178	753	7359
Griewank mod.	20	9.4514e−10	942	1	41
Pinter	2	3.3940e−09	170	2	7
Pinter	5	1.5414e−06	25,618	101	629
Pinter	10	6.0700e−05	444,607	1048	10,259
Pinter	20	7.1553e−05	42,682,351	55,392	1,000,478
Griewrot2	2	−1.7999e+02	80	1	5
Griewrot2	10	−1.7999e+02	816	1	21
Griewrot2	30	−1.7999e+02	5063	1	61
Griewrot2	50	−1.7998e+02	10,205	2	199
Ackley	2	3.9968e−15	3348	38	129
Ackley	10	4.4409e−16	412	1	21
Ackley	30	4.4409e−16	1232	1	61
Ackley	50	4.4409e−16	2052	1	101
Dixon Price	2	3.3621e−09	87	1	5
Dixon Price	10	5.7617e−08	178,195	503	7531
Dixon Price	25	7.6712e−08	283,077,766	339,219	12,523,913
Dixon Price	50	**6.6667e−01**	36,182,567	21,107	1,863,678
Easom	2	−1.0000e+00	131,165	2190	6579
Michalewicz	2	−1.8013e+00	69	1	5
Michalewicz	5	−4.6877e+00	130,058	920	6,137
Michalewicz	10	−9.6601e+00	21,699,660	67,701	662,976
Rastrigin	2	1.9443e−07	336	4	13
Rastrigin	10	9.7216e−07	12,751	30	383
Rastrigin	30	2.9165e−06	280,683	221	8491
Rastrigin	50	4.8608e−06	1,265,672	601	37,703
Test problems from Ref. [28]					
Beale	2	0.0000e+00	137	1	5
Bohachevsky 1	2	2.5101e−08	96	1	5
Bohachevsky 2	2	2.0964e−08	96	1	5
Bohachevsky 3	2	1.0140e−07	122	1	5
Booth	2	0.0000e+00	75	1	5
Colville	4	6.1275e−05	225,066	62	387
Perm1	2	1.0801e−06	272	1	5
Perm1	5	8.9149e−05	1,568,231,384	582,059	4,009,631
Perm2	2	0.0000e+00	62	1	5

Table 1 continued

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Perm2	5	6.3056e−07	57,766	29	195
Powell	4	0.0000e+00	141	1	9
Powell	8	0.0000e+00	286	1	17
Powell	16	0.0000e+00	576	1	33
Powell	24	0.0000e+00	866	1	49
Powersum	4	0.0000e+00	127	1	9
Schwefel	2	5.5892e−08	1624	15	57
Schwefel	5	1.3973e−07	201,763	743	6013
Schwefel	10	2.7574e−07	24,909,898	45,504	500,094
Schwefel	20	5.5133e−07	431,426,380	496,122	10,407,001
Test problems form Ref. [29]					
Sphere	10	−1.4000e+03	573	1	21
Rot. Elliptic	10	−1.2999e+03	5,642,213,969	1,261,645	14,213,501
Rot. Discus	10	−1.2000e+03	47,680	31	397
Rot. Bent Cigar	10	**−4.1537e+02**	10,584,086,678	2,109,744	31,004,082
Different Powers	10	−1.0000e+03	589	1	21
Rot. Rosenbrock	10	−8.9997e+02	36,686	9	145
Rot. Schaffers (F7)	10	−7.9996e+02	374,242,440	91,928	1,014,337
Rot. Ackley	10	**−6.8008e+02**	943,365,696	2,134,932	27,501,329
Rot. Weierstrass	10	**−5.9914e+02**	359,339,196	871,942	9,000,562
Rot. Griewank	10	−4.9999e+02	14,487	16	237
Rastrigin	10	−4.0000e+02	52,620,517	11,6714	1,213,659
Rot. Rastrigin	10	**−2.9801e+02**	3,208,062,726	4,780,133	50,004,016
Non-continuous rot. rastrigin	10	**−1.9801e+02**	1,939,249,826	4,921,080	50,004,772
Schwefel	10	**−9.3295e+01**	1,597,692,027	4,458,022	50,004,906
Rot. Schwefel	10	**1.1512e+02**	189,035,848	491,308	5,000,954
Rot. Katsuura	10	**2.0004e+02**	204,960,561	570,002	7,000,330
Lunacek Bi-Rastrigin	10	3.0003e+02	87,808,133	205,922	2,009,957
Rot. Lunacek Bi-Rastrigin	10	**4.0202e+02**	5,511,932,601	5,700,552	50,003,148
Expanded Griewank + Rosenbrock	10	5.0005e+02	12,087,229	3048	28,155
Expanded Schaffer (F6)	10	**6.0154e+02**	2,396,769,984	3,454,923	50,000,962
Comp. Function 1	10	7.0000e+02	104,003	157	1645
Comp. Function 2	10	**9.0284e+02**	1,655,305,428	4,450,965	50,006,510
Comp. Function 3	10	**1.0268e+03**	265,301,785	645,319	6,500,803
Comp. Function 4	10	**1.0435e+03**	363,888,151	704,658	7,500,899
Comp. Function 5	10	**1.2019e+03**	2,518,222,053	4,926,281	50,003,832
Comp. Function 6	10	**1.2288e+03**	1,265,561,798	936,028	10,501,167
Comp. Function 7	10	**1.4717e+03**	118,906,294	282,856	3,000,676
Comp. Function 8	10	1.4000e+03	33,824,697	49,859	412,627

- The derivative-free version of Algorithm DFO-DIRMIN-TL fails only on one problem from Ref. [3] (as opposed to the derivative-based version which never fails). This confirms the good behavior of the local minimization routine that, without using derivatives, is attracted by any global minimum point (see, e.g., [22] for smooth problems and [30] for nonsmooth optimization).
- DFO-DIRMIN-TL fails on 17 out of 103 test problems, which can be considered quite a good result for a derivative-free algorithm;
- 45 problems are solved by a single local minimization performed by Algorithm DF starting from the centroid of the feasible domain;
- The huge number of function evaluations (especially corresponding to the failures) is due to the computational limits we adopt. Indeed the maximum number of $50000 \times n$ hyperrectangles is reasonably low in a DIRECT context, but the introduction of local minimizations and the maximum number of 100 restarts lead to a number of function evaluations that can be extremely high (see, e.g., problem Rot. Bent Cigar). Of course, these numbers are not reasonable if the objective function is expensive, but they can be decreased by reducing the number of allowed restarts. We choose a large number of restarts in order to evaluate the behavior of the algorithm in the long run.

From now on, we focus on the subset of 58 difficult test problems: in particular we drop from the test set the 45 "easy" problems, which are those solved by the first local minimization.

Note that algorithm DFO-DIRMIN-TL is able to solve 41 out of 58 difficult test problems, thus showing that the introduction of a local minimization phase within the DIRECT framework is significantly beneficial (see, e.g., [3] for the results of the original DIRECT method on a subset of the problems). Considering the structure of DFO-DIRMIN-TL, Lemma 2.1 and Proposition 2.1, the reported failures are necessarily due to narrow regions of attraction surrounding every global minimum.

In subsequent sections we shall propose new variants of DFO-DIRMIN-TL with the aim of improving the above results and, in particular, its efficiency and reliability.

3 A new derivative-free algorithm

Looking at the results in the previous section, it emerges the large number of function evaluations needed in order to get convergence. Drawing inspiration from [22], we modify DFO-DIRMIN by updating during the iterations a working set, of dimension $n_{wks} = 100n$, of "active" local minimizations that are carried out in a balanced way. More specifically, starting from each centroid of the potentially optimal hyperrectangles, Algorithm DF is executed with an adaptive tolerance that is updated during the iterations on the basis of the behaviour of the active minimizations and becomes tighter and tighter as the algorithm proceeds. The points produced by the DIRECT partitioning strategy are added to the working set, if there are positions available. Whenever a new partial minimization is performed and the working set is full, the point is added only if its objective function value is better than the worst one present in the current working set, which is

replaced. Hence, the above strategy implies that only a limited number of local minimizations are completely performed, thus focusing on the most "promising" ones.

Furthermore, at the end of every iteration, all the points in the working set are updated by means of a single iteration of Algorithm DF. Whenever the maximum stepsize of an active minimization falls below the threshold tol, that minimization is removed from the working set, leaving space for a new one. We report below the resulting algorithm BDF-DIRMIN.

Algorithm BDF-DIRMIN

$\mathcal{H}_1 = \{D\},\ c = \text{center of } D,\ f_{min} = f(c),\ X_{min} = \{c\},\ tol > 0, \alpha_{\max} > tol,$
$kmax \geq 1, \mathcal{W}_1 = \emptyset, n_{wks} \geq 1,\ k = 1$

repeat

identify the potentially optimal hyperrectangles $\mathcal{P}_k$ in $\mathcal{H}_k$

for all centroids c_i of hyperrectangles in $\mathcal{P}_k$ compute $(\hat{c}_i, \hat{\alpha}_i) = DF(c_i, \alpha(c_i), \alpha_{\max}, kmax)$

if $(|\mathcal{W}_k| < n_{wks})$

set $\mathcal{W}_k = \mathcal{W}_k \cup (\hat{c}_i, \hat{\alpha}_i)$,

elseif $f(\hat{c}_i) < f(c_j),\ c_j = \arg\max_{y \in \mathcal{W}_k}\{f(y)\}$, set $\mathcal{W}_k = \mathcal{W}_k \cup (\hat{c}_i, \hat{\alpha}_i) \setminus (c_j, \hat{\alpha}_j)$

end if

subdivide the potentially optimal hyperrectangles to build a new partition $\mathcal{H}_{k+1}$

evaluate f in the centers of the new hyperrectangles

For every pair $(y_i, \alpha_i) \in \mathcal{W}_k$ set $(\tilde{y}_i, \tilde{\alpha}_i) = DF(y_i, \alpha_i, tol, 1)$. Set $\mathcal{W}_k := \bigcup_{i=1}^{|\mathcal{W}_k|} (\tilde{y}_i, \tilde{\alpha}_i)$

compute $f(y_{\min}) = \min_{i \in \mathcal{W}_k} f(y_i)$ and $\alpha_{max} = \max_{i \in \mathcal{W}_k}\{\alpha_i\}$.

Remove from $\mathcal{W}_k$ all the (y, α) such that $\max_{j=1,\ldots,n} \alpha_j \leq tol$.

$f_{min} = \min\{f(c) : c \in C_k, f(y_{\min})\},\ X_{min} = \{x \in D : f(x) = f_{min}\},\ k = k + 1$
$(C_k = \{\text{centers of the hyperrectangles in } \mathcal{H}_k\})$

until (stopping criterion satisfied)

return f_{min}, X_{min}

The final algorithm is obtained by embedding again the new algorithm BDF-DIRMIN in the same general scheme as DFO-DIRMIN-TL, where the algorithm is "restarted" by solving problem (2) around promising points.

Algorithm BDF-DIRMIN-TL

$x_{min} = \tilde{x} = (1/2 \ldots 1/2)^\top$, $f_{min} = f(x_{min})$, $\mathcal{N} = \emptyset$, $\mathcal{O} = \emptyset$, $maxint \gg 0$

Repeat

(S.1) Apply BDF-DIRMIN to Problem (2) until $maxint$ hyperintervals have been generated and let $\hat{x}$ be the best point produced and $\mathcal{W}$ be the set of "promising" stationary points.

(S.2) If ($f_{\tilde{x}}(\hat{x}) < f_{min}$) then set $f_{min} = f_{\tilde{x}}(\hat{x})$, $x_{min} = \tilde{x} = \hat{x}$, $\mathcal{N} = \emptyset$, $\mathcal{O} = \emptyset$ and **go to** Step (S.1).

(S.3) Otherwise set

$$\mathcal{N} = \mathcal{N} \cup \left\{ y \in \mathcal{W} : f(y) - f_{min} \leq \epsilon_f \quad \text{and} \quad \min_{x \in \mathcal{N} \cup \{x_{min}\}} d(y, x) > \epsilon_d \right\}.$$

(S.4) choose $\bar{x} \in \mathcal{N} \setminus \mathcal{O}$, set $\mathcal{O} = \mathcal{O} \cup \{\bar{x}\}$, $\tilde{x} = \bar{x}$.

Until ($\mathcal{N} \setminus \mathcal{O} = \emptyset$)

Return f_{min}, x_{min}

Note that, since DIRECT is the algorithm constituting the skeleton of BDF-DIRMIN, the latter one inherits the DIRECT convergence property, as stated in Lemma 2.1. Further, the step size $\alpha(c^i)$, with c^i center of D_i, is given by the bounds defining hyperrectangle D_i divided by two, the parameter tol is set to 10^{-4}, and $kmax$ is equal to 5000.

In Table 2, we report the results of Algorithm BDF-DIRMIN-TL on the 58 difficult problems. Looking at the table it can be noted the smaller number of function evaluations used by Algorithm BDF-DIRMIN-TL as opposed to DFO-DIRMIN-TL.

In order to better evaluate the savings in terms of function evaluations, in Fig. 3 we plot the cumulative distribution function $\rho(\tau)$ defined as:

$$\rho(\tau) = \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{\text{nf}_{p,2}}{\text{nf}_{p,1}} \leq \tau \right\} \right|,$$

where $\mathcal{P}$ is the set of test problems, and $\text{nf}_{p,1}$ ($\text{nf}_{p,2}$) is the number of function evaluations required by DFO-DIRMIN-TL (BDF-DIRMIN-TL) to stop when solving problem $p \in \mathcal{P}$. Function $\rho(\tau)$ helps comparing the performances of the two algorithms in terms of overall computational burden. In particular, Figure 3 shows that BDF-DIRMIN-TL stops in less than half the number of function evaluations required by DFO-DIRMIN-TL on about 87% of the problems. Obviously, this greater efficiency has a price: indeed, Algorithm BDF-DIRMIN-TL fails on 20 problems (out of 58) whereas Algorithm DFO-DIRMIN-TL only fails on 17 problems.

4 A new algorithm using the plateau modification function

Now, we try to improve the reliability of DFO-DIRMIN-TL, i.e., its ability to locate the global optimum, without worrying to much about the efficiency, i.e., the required number of local minimizations and function evaluations. To this aim, we first define the following "plateau" modification of the objective function [15]:

Table 2 Results of BDF-DIRMIN-TL

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Test problems from Ref. [3]					
Schubert	2	−1.8672e+02	109	10	47
Schub. pen. 1	2	−1.8673e+02	235	31	105
Schub. pen. 2	2	−1.8673e+02	193	24	75
Quartic	2	−3.5200e−01	96	11	55
Shekel $m = 7$	4	−1.0402e+01	126	12	51
Shekel $m = 10$	4	−1.0536e+01	142	15	57
Griewank mod.	2	1.9831e−05	8072	953	3421
Griewank mod.	5	2.3800e−05	274,190	24,104	133,129
Griewank mod.	10	1.0188e−05	14,724	963	9495
Pinter	2	1.6568e−05	197	25	105
Pinter	5	9.8179e−05	1843	214	1255
Pinter	10	9.5931e−05	16,393	1310	12,939
Pinter	20	9.9644e−05	107,872	4667	97,053
Griewrot2	50	−1.7998e+02	8468	71	6491
Ackley	2	3.9968e−15	407	42	145
Dixon Price	10	8.9499e−05	12,464	679	9833
Dixon Price	25	**6.6667e−01**	5,492,616	134,409	5,001,820
Dixon Price	50	**6.6667e−01**	1,940,253	21,107	1,863,368
Easom	2	−1.0000e+00	11,043	2198	6625
Michalewicz	5	−4.6876e+00	9538	983	6523
Michalewicz	10	−9.6595e+00	849,039	67,724	663,164
Rastrigin	2	1.9443e−07	181	22	87
Rastrigin	10	9.7216e−07	1188	72	817
Rastrigin	30	2.9165e−06	13,542	346	12,217
Rastrigin	50	4.8608e−06	51,820	834	49,023
Test problems from Ref. [28]					
Colville	4	7.3521e−05	1335	167	883
Perm1	5	**7.6076e−04**	36,624,683	3,647,938	25,003,738
Perm2	5	8.5173e−05	5275	597	3611
Schwefel	2	7.4746e−05	515	47	165
Schwefel	5	7.7473e−05	11,670	768	6177
Schwefel	10	2.5060e−05	822,302	45,537	500,522
Schwefel	20	4.4414e−05	12,217,489	496,164	10,408,019
Test problems from Ref. [29]					
Rot. Elliptic	10	−1.3000e+03	581,930,281	4,448,993	50,013,308
Rot. Discus	10	−1.2000e+03	18,501	481	4751
Rot. Bent Cigar	10	**−1.0907e+03**	122,143,170	166,257	2,500,131
Rot. Rosenbrock	10	−9.0000e+02	96,883	144	1505
Rot. Schaffers (F7)	10	−7.9992e+02	12,735,661	61,068	668,590
Rot. Ackley	10	**−6.8002e+02**	15,935,575	350,139	4,500,239

Table 2 continued

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Rot. Weierstrass	10	**−5.9919e+02**	12,565,878	657,155	6,500,311
Rot. Griewank	10	−4.9995e+02	16,384	533	4865
Rastrigin	10	−4.0000e+02	22,445,131	146,194	1,531,248
Rot. Rastrigin	10	**−2.9801e+02**	87,462,231	4,774,990	50,003,974
Non-continuous Rot. Rastrigin	10	**−1.9801e+02**	83,304,804	4,800,436	50,004,334
Schwefel	10	**−9.3233e+01**	353,592,387	4,433,733	50,004,226
Rot. Schwefel	10	**1.5011e+02**	5,107,995	363,772	3,500,411
Rot. Katsuura	10	**2.0003e+02**	21,832,018	490,174	6,000,264
Lunacek Bi-Rastrigin	10	3.0003e+02	45,057,438	205,910	2,009,909
Rot. Lunacek Bi-Rastrigin	10	**4.0222e+02**	391,928,534	5,702,225	50,003,450
Expanded Griewank	10	5.0005e+02	1,149,771	3995	36901
Expanded Schaffer	10	**6.0185e+02**	685,417,439	3,499,448	50,001,068
Comp. function 1	10	7.0000e+02	29,144	189	1933
Comp. function 2	10	**9.0284e+02**	362,203,872	4,455,047	50,006,616
Comp. function 3	10	**1.0323e+03**	4,701,450	293,292	3,000,406
Comp. function 4	10	**1.0291e+03**	6,281,748	389,132	4,500,485
Comp. function 5	10	**1.2025e+03**	85,773,668	4,966,625	50,003,834
Comp. function 6	10	**1.2344e+03**	21,781,707	286,120	3,000,222
Comp. function 7	10	**1.5882e+03**	717,640	45,111	500,041
Comp. function 8	10	**1.5000e+03**	2,231,849	161,795	1,500,363

$$\tilde{f}(x) = f(\hat{x}), \quad \text{where } (\hat{x}, \hat{\alpha}) = DF(x, \alpha_0, tol, kmax).$$

In particular, we substitute to the original objective function, the function value of the stationary point obtained by algorithm DF starting from the point x. The resulting function is a piecewise constant function (the so called "plateau" function, see, e.g., [15–17]) which, under the stated assumptions, is bounded from below. Then, in place of Problem (2), we look for a solution of the following Problem.

$$\min_{y \in [0,1]^n} \tilde{f}_{\bar{x}}(y). \tag{3}$$

We define a new algorithm, that we call `DIRFOB-TL`, that, roughly speaking, consists in repeatedly applying algorithm `DIRECT` to the find a global minimizer of Problem (3), i.e. a global minimizer of the "plateau" function $\tilde{f}$. In Algorithm `DIRFOB-TL` we maintain the restarting technique used in `DFO-DIRMIN-TL`, by means of the same nonlinear transformation applied on a set of "promising" points.

Note that, even though the plateau modification function is not Lipschitz continuous, the everywhere convergence property of `DIRECT` is still valid. Indeed, as showed in [20], this property follows from

$$I_k^* \cap \{i \in I_k : \|u^i - l^i\| = d_k^{\max}\} \neq \emptyset,$$

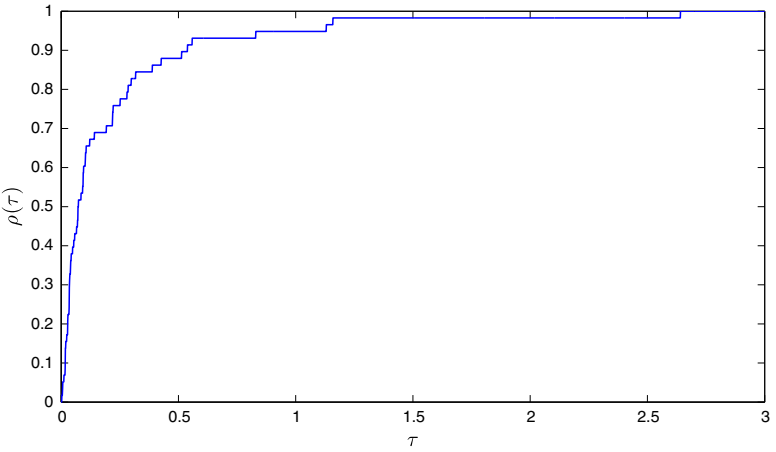


Fig. 3 Comparison of DFO-DIRMIN-TL and BDF-DIRMIN-TL by means of the cumulative distribution function $\rho(\tau)$

where $d_k^{\max} = \max_{i \in I_k} \|u^i - l^i\|$, which is true independently from the continuity of the objective function.

Algorithm DIRFOB-TL

$x_{min} = \tilde{x} = (1/2 \ldots 1/2)^\top$, $f_{min} = \tilde{f}(x_{min})$, $\mathcal{N} = \emptyset$, $\mathcal{O} = \emptyset$, $maxint \gg 0$, $k = 1$, $\epsilon_f, \epsilon_d > 0$.

Repeat

(S.1) Apply DIRECT to Problem (3) until $maxint$ hyperintervals have been generated and let $\hat{x}$ be the best point produced and $\mathcal{W}$ be the set of "promising" stationary points.

(S.2) If ($\tilde{f}_{\tilde{x}}(\hat{x}) < f_{min}$) then set $f_{min} = \tilde{f}_{\tilde{x}}(\hat{x})$, $x_{min} = \tilde{x} = \hat{x}$, $\mathcal{N} = \emptyset$, $\mathcal{O} = \emptyset$ and **go to** Step (S.1).

(S.3) Otherwise set

$$\mathcal{N} = \mathcal{N} \cup \left\{ y \in \mathcal{W} : \tilde{f}(y) - f_{min} \leq \epsilon_f \quad \text{and} \quad \min_{x \in \mathcal{N} \cup \{x_{min}\}} d(y, x) > \epsilon_d \right\}.$$

(S.4) choose $\bar{x} \in \mathcal{N} \setminus \mathcal{O}$, set $\mathcal{O} = \mathcal{O} \cup \{\bar{x}\}$, $\tilde{x} = \bar{x}$.

Until ($\mathcal{N} \setminus \mathcal{O} = \emptyset$)

Return f_{min}, x_{min}

In Table 3 we report the results obtained by Algorithm DIRFOB-TL on the set of 58 difficult test problems.

As it can be seen, the reliability of Algorithm DIRFOB-TL is significantly improved with respect to DFO-DIRMIN-TL. Indeed, DIRFOB-TL only fails on 4 problems out of 58. Not surprisingly Algorithm DIRFOB-TL is generally more expensive than DFO-DIRMIN-TL (and hence of BDF-DIRMIN).

However, this is not always the case as it emerges from Fig. 4 where we plot function $\rho(\tau)$ for the comparison among DFO-DIRMIN-TL and DIRFOB-TL. In particular, we plot $\rho(\tau)$ for $\tau \in [0, 1]$ (left side of Fig. 4) and for $\tau \in [1, 60]$ (right side of Fig. 4).

Table 3 Results of DIRFOB-TL

Problem	n	$f(\bar{x})$	n.f.	n.int.
Test problems from Ref. [3]				
Schubert	2	−1.8673e+02	2315	205
Schub. pen. 1	2	−1.8673e+02	6,316	207
Schub. pen. 2	2	−1.8673e+02	3,868	203
Quartic	2	−3.5239e−01	84	19
Shekel $m = 7$	4	−1.0403e+01	164	23
Shekel $m = 10$	4	−1.0536e+01	160	23
Griewank mod.	2	1.7036e−09	14,746	233
Griewank mod.	5	4.3145e−10	154,689	845
Griewank mod.	10	1.4426e−09	182,128	1009
Pinter	2	3.3940e−09	90	5
Pinter	5	1.5150e−06	10,547	521
Pinter	10	1.5090e−06	24,151,879	89,033
Pinter	20	1.2913e−05	822,928,421	1,569,822
Griewrot2	50	−1.7998e+02	5104	101
Ackley	2	5.6461e−05	466	151
Dixon Price	10	9.3893e−08	137,441	1087
Dixon Price	25	1.3575e−08	5,731,881	5419
Dixon Price	50	4.8111e−08	1,269,313,104	460,873
Easom	2	−1.0000e+00	318,161	8829
Michalewicz	5	−4.6877e+00	23,308	533
Michalewicz	10	−9.6601e+00	30,065,636	178,221
Rastrigin	2	7.4494e−07	1856	209
Rastrigin	10	4.9073e−08	12,010	1103
Rastrigin	30	2.7650e−06	6145	3223
Rastrigin	50	4.4339e−06	10,277	5355
Test problems from Ref. [28]				
Colville	4	2.8539e−06	13,120,363	60,281
Perm1	5	**9.3933e−04**	7,279,772,213	25,003,136
Perm2	5	3.6115e−05	168,322,925	560,003
Schwefel	2	5.5145e−08	525	121
Schwefel	5	1.3907e−07	91,223	527
Schwefel	10	2.7814e−07	483,319	1237
Schwefel	20	5.5629e−07	4,295,690	5361
Test problems from Ref. [29]				
Rot. Elliptic	10	−1.3000e+03	121,196,868,750	34,503,835
Rot. Discus	10	−1.2000e+03	2,611,405	1925
Rot. Bent Cigar	10	−1.1000e+03	4,059,627,002	1,017,617
Rot. Rosenbrock	10	−9.0000e+02	110,939	1,067
Rot. Schaffers (F7)	10	−7.9993e+02	425,933,887	527,362
Rot. Ackley	10	−6.9997e+02	650,470,093	3,501,090

Table 3 continued

Problem	n	$f(\bar{x})$	n.f.	n.int.
Rot. Weierstrass	10	−5.9997e+02	263,655,451	1,018,829
Rot. Griewank	10	−4.9998e+02	22,073	1047
Rastrigin	10	−4.0000e+02	202,714,074	511,630
Rot. Rastrigin	10	−3.0000e+02	8,398,182	20,159
Non-continuous rot. rastrigin	10	−2.0000e+02	3,864,394	13,701
Schwefel	10	−1.0000e+02	186,574,187	522,190
Rot. Schwefel	10	1.0000e+02	546,425,335	1,501,278
Rot. Katsuura	10	**2.0003e+02**	610,000,645	3,000,142
Lunacek Bi-Rastrigin	10	3.0000e+02	441,877,163	1,291,487
Rot. Lunacek Bi-Rastrigin	10	4.0000e+02	11,556,220	28,031
Expanded Griewank + Rosenbrock	10	5.0004e+02	31,844,643	49,393
Expanded Schaffer (F6)	10	6.0006e+02	54,060,331	138,325
Comp. Function 1	10	7.0000e+02	196,612	1041
Comp. Function 2	10	8.0000e+02	189,617,796	519,686
Comp. Function 3	10	9.0000e+02	97,308,583	265,987
Comp. Function 4	10	1.0000e+03	553,844,533	1,501,776
Comp. Function 5	10	**1.2000e+03**	2,008,600,802	5,500,523
Comp. Function 6	10	1.2000e+03	1,580,140,782	2,017,773
Comp. Function 7	10	**1.4000e+03**	3,127,930,717	8,000,486
Comp. Function 8	10	1.4000e+03	1,541,217	3035

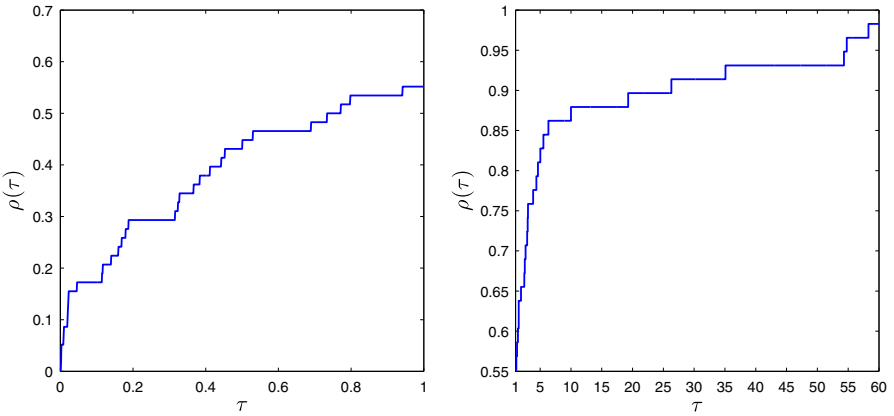


Fig. 4 Comparison of DFO-DIRMIN-TL and DIRFOB-TL by means of the cumulative distribution function $\rho(\tau)$

It can be seen that DIRFOB-TL requires a number of function evaluation not greater than that required by DFO-DIRMIN-TL on approximately half of the test problems (see, e.g., left side of Fig. 4).

5 An application to a protein structural alignment problem

Given two protein structures $\mathcal{P}$ and $\mathcal{Q}$, let us denote by P and Q the two finite sets of points corresponding to the atoms of the active sites of the two structures $\mathcal{P}$ and $\mathcal{Q}$, respectively. We let $n = |P|$ and $m = |Q|$ and assume, without loss of generality, that $n \leq m$. The set P is conventionally representative of a query shape while Q defines a reference model shape.

An isometric transformation in three-dimensional space can be defined by a unit quaternion $a_r = (a_0, a_1, a_2, a_3)^\top \in \Re^4$ ($\|a_r\| = 1$) and by a translation vector $a_t \in \Re^3$. Let $a^\top = (a_r^\top \; a_t^\top)$ be the transformation defining vector and denote by T_a the corresponding transformation, so that

$$y = T_a(x) = R(a_r)x + a_t$$

for every $x \in \Re^3$, where $R(a_r)$ is the rotation matrix defined by the unit quaternion a_r.

Let $\Theta \subset \Re^7$ be the set of all vectors $a \in \Re^7$ defining an isometric transformation in $\Re^3$. Given a transformation vector $a \in \Theta$, let $T_a(P) = P_a$ denote the set of points obtained by applying the transformation T_a to every point of P, that is

$$T_a(P) = P_a = \{y : y = R(a_r)p + a_t, \; \forall \; p \in P\}.$$

Let $\psi : P \to Q$ denote a point to point mapping that associates to every point of P a point of Q. Since, as assumed above, P and Q are finite sets, the class Ψ of all mappings ψ has finite cardinality given by $|\Psi| = m^n$.

Let $\psi \in \Psi$ be a given mapping and a be a vector defining an isometric transformation, then the mean square error function between P and Q is the following

$$f(\psi, a) = \frac{1}{n} \sum_{p \in P} \|\psi(p) - R(a_r)p - a_t\|^2.$$

Let us denote by $\psi(a) = \arg\min_{\psi \in \Psi} f(\psi, a)$ the closest point mapping [31] and $g(a) = f(\psi(a), a)$. Then, the surface alignment problem can be posed as

$$\min_{a \in \Theta} g(a). \tag{4}$$

Every global solution a^* of (4) is, by definition, a solution such that $f(\psi(a^*), a^*) \leq f(\psi(a), a)$, for all $a \in \Theta$. Problem (4) is a global optimization problem with a black-box objective function, a feasible set Θ described by box constraints and some "easy" constraints (i.e., $\|a_r\| = 1$). Furthermore, numerical experiments show that the problem has many local minima and a global minimum exists with reasonably large basin of attraction.

Since, among the proposed algorithms `DIRFOB-TL` is the more robust one, this is the code that we employ to find correct alignments on the set of 19 proteins used in [19]. The proteins all bind ligand ATP and are from different families according to the structural classification SCOP [32].

Table 4 Results obtained by DIRFOB-TL for the protein structural alignment problem

Protein pair	DIRFOB-TL		CO	
	N. align. atoms	RMSD	N. corresp. atoms	RMSD
1atpE-1phk	66	0.90	57	0.91
1atpE-1csn	64	0.99	50	1.18
1atpE-1hck	61	1.50	62	1.20
1atpE-1ayl	29	1.91	12	1.21
1atpE-1yag	28	2.05	20	1.92
1atpE-1nsf	28	2.15	34	2.11
1atpE-1j7k	25	2.09	25	1.81
1atpE-1a82	24	1.81	19	2.02
1atpE-1mjhA	23	2.22	16	2.28
1atpE-1kp2A	22	1.92	13	1.51
1atpE-1kay	21	2.15	20	1.90
1atpE-1jjv	19	2.02	18	1.76
1atpE-1e2q	18	2.07	15	1.39
1atpE-1gn8A	16	2.11	17	2.37
1atpE-1b8aA	12	2.08	16	2.05
1atpE-1f9aC	11	2.35	21	2.17
1atpE-1e8xA	9	1.92	24	1.74
1atpE-1g5t	8	1.77	7	2.26

We performed pairwise comparisons of the active site of the catalytic subunit of cAMP-dependent Protein-Kinase (pdb code 1atp, chain E) with each of the remaining proteins of the input data set. Of the set of proteins only three belong to the same SCOP family as 1atp, namely 1phk, 1csn and 1hck. In Table 4 for each comparison we report the number of aligned atoms along with the Root Mean Square Distance (RMSD) obtained by DIRFOB-TL and CO (i.e., the algorithm proposed in [19]), respectively.

We observe that both methods correctly rank at the top three positions (with respect to the number of aligned atoms) proteins in the same family as 1atp, that is 1phk, 1hck and 1csn. It can also be noted that DIRFOB-TL better separates proteins in the same SCOP family as 1atp, from the others. Indeed, DIRFOB-TL aligns 29 atoms for the protein pair 1atp-1ayl, whereas CO aligns 34 atoms for the protein pair 1atp-1nsf. Hence, the gaps obtained by DIRFOB-TL and CO between different SCOP families are 32 and 28 atoms, respectively.

6 Conclusions

In the paper we focus on the definition of new deterministic algorithms for the solution of hard box-constrained global optimization problems when derivatives of the objective function are unavailable. We concentrate on how to make efficient use of local minimization within the framework of DIRECT-type algorithms and we propose three different DIRECT-type algorithms making use of derivative-free local searches combined with nonlinear transformations of the feasible domain and, possibly, of the objective function. It should be noted that all of the proposed modifications are also

suitable to other more sophisticated partition-based algorithms, like for instance those based on simplicial or diagonal partitioning.

Our starting point is algorithm DIRMIN-TL, which has been recently proposed by exploiting an efficient Newton-type local minimization routine. The first algorithm that we propose, DFO-DIRMIN-TL, is indeed a simple adaptation of DIRMIN-TL to the derivative-free context. The use of a derivative-free local search routine, in place of the more efficient Newton-type one, still gives us a code with a fairly good reliability (ability to locate the global minimum). This is most probably because both the local search engines are attracted by any global minimum point. Then, we devise two more algorithms trying to improve both the efficiency and reliability of DFO-DIRMIN-TL. More precisely, we show that algorithm BDF-DIRMIN-TL is far more efficient than DFO-DIRMIN-TL in terms of required function evaluations at the expense of a reduced reliability. Then, we try to improve on the reliability and come up with algorithm DIRFOB-TL which is indeed far more reliable than both DFO-DIRMIN-TL and BDF-DIRMIN-TL though it is generally more expensive than the first two codes. Finally, we report the results obtained by Algorithm DIRFOB-TL on a difficult protein structural alignment problem and show that it performs better than a recently proposed random search method tailored to the specific application.

Acknowledgments We thank two anonymous Reviewers whose helpful comments and suggestions helped up to improve the paper.

Appendix

The derivative-free local algorithm

In this section we report the sketch of a derivative-free procedure for unconstrained local minimization [23].

Algorithm DF $(\hat{x}, \hat{\alpha}) = DF(x_0, \alpha_0, tol, kmax)$

Data $d^1, \ldots, d^n \in \Re^n$.

Set $\alpha_{\max} = \max_{i=1,\ldots,n} \alpha_0^i$, $k = 0$

Repeat

For $i = 1, \ldots, n$

starting from α_k perform a derivative free linesearch along d^i producing α_{k+1}^i

End For

Set $x_{k+1} = x_k + \sum_{i=1}^{n} \alpha_{k+1}^i d^i$

Set $\alpha_{\max} = \max_{i=1,\ldots,n} \alpha_{k+1}^i$, $k = k + 1$.

Until $((\alpha_{\max} < tol)$ and $(k = kmax))$

Return $(x_k, \alpha_k, \alpha_{\max})$

Table 5 Test problems

Problem	n	f^*
Problems from [3]		
Schubert	2	−1.8673e+02
Schub. pen. 1	2	−1.8673e+02
Schub. pen. 2	2	−1.8673e+02
S-H. Camel B.	2	−1.0316e+00
Goldstein-Price	2	3.0000e+00
Treccani mod.	2	0.0000e+00
Quartic	2	−3.5200e−01
Shekel $m = 5$	4	−1.0153e+01
Shekel $m = 7$	4	−1.0403e+01
Shekel $m = 10$	4	-1.0536e+01
Espon. mod.	2	−1.0000e+00
Espon. mod.	4	−1.0000e+00
Cos-mix mod.	2	−2.0000e - 01
Cos-mix mod.	4	−4.0000e - 01
Hartman	3	−3.8627e+00
Hartman	6	−3.3223e+00
$5n$ loc-min	2,5,10,20	0.0000e+00
$10n$ loc-min	2,5,10,20	0.0000e+00
$15n$ loc-min	2,5,10,20	0.0000e+00
Griewank mod.	2,5,10,20	0.0000e+00
Pinter	2,5,10,20	0.0000e+00
Griewrot2	2,10,30,50	−1.8000e+02
Ackley	2,10,30,50	0.0000e+00
Dixon Price	2,10,25,50	0.0000e+00
Easom	2	−1.0000e+00
Michalewicz	2	−1.8013e+00
Michalewicz	5	−4.6876e+00
Michalewicz	10	−9.6602e+00
Rastrigin	2,10,30,50	0.0000e+00
Problems from [28]		
Beale	2	0.0000e+00
Bohachevsky 1	2	0.0000e+00
Bohachevsky 2	2	0.0000e+00
Bohachevsky 3	2	0.0000e+00
Booth	2	0.0000e+00
Colville	4	0.0000e+00
Perm 1	2,5	0.0000e+00
Perm 2	2,5	0.0000e+00
Powell	4,8,16,24	0.0000e+00
Powersum	4	0.0000e+00

Table 5 continued

Problem	n	f^*
Schwefel	2,5,10,20	0.0000e+00
Problems from [29]		
Sphere	10	−1.4000e+03
Rot. Elliptic	10	−1.2999e+03
Rot. Discus	10	−1.2000e+03
Rot. Bent Cigar	10	−1.1000e+03
Different Powers	10	−1.0000e+03
Rot. Rosenbrock	10	−8.9997e+02
Rot. Schaffers (F7)	10	−8.0000e+02
Rot. Ackley	10	−7.0000e+02
Rot. Weierstrass	10	−6.0000e+02
Rot. Griewank	10	−4.9999e+02
Rastrigin	10	−4.0000e+02
Rot. Rastrigin	10	−3.0000e+02
Non-continuous Rot. Rastrigin	10	−2.0000e+02
Schwefel	10	−1.0000e+02
Rot. Schwefel	10	1.0000e+02
Rot. Katsuura	10	2.0000e+02
Lunacek Bi-Rastrigin	10	3.0000e+02
Rot. Lunacek Bi-Rastrigin	10	4.0000e+02
Expanded Griewank + Rosenbrock	10	5.0000e+02
Expanded Schaffer (F6)	10	6.0000e+02
Comp. Function 1	10	7.0000e+02
Comp. Function 2	10	8.0000e+02
Comp. Function 3	10	9.0000e+02
Comp. Function 4	10	1.0000e+03
Comp. Function 5	10	1.1000e+03
Comp. Function 6	10	1.2000e+03
Comp. Function 7	10	1.3000e+03
Comp. Function 8	10	1.4000e+03

In particular, the actual implementation of Algorithm DF that we use is based on the one proposed in [23] where $d^i = e^i$, $i = 1, \ldots, n$, where e^i denotes the i-th coordinate direction in $\Re^n$.

Test set description

In the following table, for each problem of our test set, we report its name, the adopted number of variables and the value of the known global minimum point.

References

1. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**(1), 157–181 (1993)
2. Jones, D.R.: DIRECT global optimization. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 725–735. Springer, Berlin (2009)
3. Liuzzi, G., Lucidi, S., Piccialli, V.: A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. Comput. Optim. Appl. **45**, 353–375 (2010)
4. Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J.: Globally-biased disimpl algorithm for expensive global optimization. J. Glob. Optim. **59**(2–3), 545–567 (2014)
5. Lera, D., Sergeyev, Y.D.: Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants. Commun. Nonlinear Sci. Numer. Simul. **23**(1–3), 328–342 (2015)
6. Di Serafino, D., Liuzzi, G., Piccialli, V., Riccio, F., Toraldo, G.: A modified DIviding RECTangles algorithm for a problem in astrophysics. J. Optim. Theory Appl. **151**, 175–190 (2011)
7. Liu, Q., Cheng, W.: A modified direct algorithm with bilevel partition. J. Glob. Optim. **60**(3), 483–499 (2014)
8. Liu, Q., Zeng, J.: Global optimization by multilevel partition. J. Glob. Optim. **61**(1), 47–69 (2015)
9. Gablonsky, J.M., Kelley, C.T.: A locally-biased form of the DIRECT algorithm. J. Glob. Optim. **21**(1), 27–37 (2001)
10. Panning, T.D., Watson, L.T., Allen, N.A., Shaffer, C.A., Tyson, J.J.: Deterministic global parameter estimation for a budding yeast model. In: Hamilton Jr., J.A., MacDonald, R., and Chinni, M.J. (eds.) Proceedings of the 2006 Spring Simulation Multiconf., High Performance Computing Symposium, pp. 195–201. Soc. for Modeling and Simulation Internat., San Diego, CA, (2006)
11. Panning, T.D., Watson, L.T., Allen, N.A., Chen, K.C., Shaffer, C.A., Tyson, J.J.: Deterministic parallel global parameter estimation for a model of the budding yeast cell cycle. J. Glob. Optim. **40**, 719–738 (2008)
12. Sergeyev, Ya D.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. SIAM J. Optim **16**(3), 910–937 (2006)
13. Paulavičius, R., Žilinskas, J.: Simplicial Global Optimization. Springer, Berlin (2014)
14. Paulavičius, R., Žilinskas, J.: Simplicial Lipschitz optimization without the Lipschitz constant. J. Glob. Optim. **59**(1), 23–40 (2014)
15. Wales, D.J., Doye, J.P.K.: Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. J. Phys. Chem. A **101**(28), 5111–5116 (1997)
16. Leary, R.H.: Global optimization on funneling landscapes. J. Glob. Optim. **18**(4), 367–383 (2000)
17. Locatelli, M., Schoen, F.: Efficient algorithms for large scale global optimization: Lennard-Jones clusters. Comput. Optim. Appl. **26**(2), 173–190 (2003)
18. Kvasov, D.E., Sergeyev, Ya D.: Deterministic approaches for solving practical black-box global optimization problems. Adv. Eng. Softw. **80**, 58–66 (2015)
19. Bertolazzi, P., Guerra, C., Liuzzi, G.: A global optimization algorithm for protein surface alignment. BMC Bioinform. **11**, 488–498 (2010)
20. Liuzzi, G., Lucidi, S., Piccialli, V.: A partition-based global optimization algorithm. J. Glob. Optim. **48**, 113–128 (2010)
21. Sergeyev, Ya D.: On convergence of "Divide the Best" global optimization algorithms. Optimization **44**(3), 303–325 (1998)
22. Liuzzi, G., Lucidi, S., Piccialli, V., Sotgiu, A.: A magnetic resonance device designed via global optimization techniques. Math. Progr. **101**(2), 339–364 (2004)
23. Lucidi, S., Sciandrone, M.: On the global convergence of derivative free methods for unconstrained optimization. SIAM J. Optim. **13**, 97–116 (2002)
24. Lucidi, S., Sciandrone, M.: A derivative-free algorithm for bound constrained optimization. Comput. Optim. Appl. **21**(2), 119–142 (2002)
25. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
26. He, J., Watson, L.T., Ramakrishnan, N., Shaffer, C.A., Verstak, A., Jiang, J., Bae, K., Tranter, W.H.: Dynamic data structures for a direct search algorithm. Comput. Optim. Appl. **23**, 5–25 (2002)
27. Jarvis, R.A.: On the identification of the convex hull of a finite set of points in the plane. Inf. Process. Lett. **2**, 18–21 (1973)

28. Hedar, A.-R.: Test problems for unconstrained optimization. http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm
29. Liang, J.J., Qu, B.-Y., Suganthan, P.N., Hernndaz-Daz, A.G.: Problem definitions and evaluation criteria for the cec 2013 special session and competition on real-parameter optimization. In: Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, (2013)
30. Di Pillo, G., Lucidi, S., Rinaldi, F.: A derivative-free algorithm for constrained global optimization based on exact penalty functions. J. Optim. Theory Appl. **164**(3), 862–882 (2015). doi:10.1007/s10957-013-0487-1
31. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach Intell. **14**, 239–255 (1992)
32. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: Scop: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. **247**, 536–540 (1995)