ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Liuzzi, S. Lucidi, V. Piccialli

EXPLOITING DERIVATIVE-FREE LOCAL SEARCHES IN DIRECT-TYPE ALGORITHMS FOR GLOBAL OPTIMIZATION

R. 1, 2014

Giampaolo Liuzzi – Consiglio Nazionale delle Ricerche, Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Viale Manzoni 30, 00185 Rome, Italy. liuzzi@iasi.cnr.it.

Stefano Lucidi – "Sapienza" Università di Roma, Dipartimento di Ingegneria Informatica Automatica e Gestionale "A. Ruberti", Via Ariosto 25, 00185 Rome, Italy. lucidi@dis.uniroma1.it.

Veronica Piccialli – Università degli Studi di Roma "Tor Vergata", Dipartimento di Ingegneria Civile e Ingegneria Informatica, Viale del Politecnico 1, 00133 Roma, Italy. piccialli@disp.uniroma2.it.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

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 strong convergence properties and a good ability to locate promising regions of 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.

Key words: Global optimization, DIRECT-type algorithms, Local minimizations

2010 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]. It 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, 2, 3].

In [3] an algorithm named DIRMIN-TL has been proposed where some modifications of the DIRECT Algorithm are introduced to enhance its performances. In particular, DIRECT behavior can be considerably improved by taking advantage of local minimizations [4, 5, 3] and nonlinear transformations of the variables [3]. We consider in detail Algorithm DIRMIN-TL from reference [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 reference [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 minimization in a distributed way. Second, drawing inspiration from [6, 7, 8], we can use the derivative-free local minimization routine to modify the objective function.

In Section 2 we present a simple adaptation of Algorithm DIRMIN-TL, from reference [3], to derivative-free optimization and present its numerical performances on a benchmark of difficult global optimization problems. In Section 3 we propose a distributed version of algorithm DIRMIN-TL where the local minimization are carried out in a distributed fashion thus considerably improving the efficiency of DIRMIN-TL. In Section 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 Section 5 we present an application of the the latter algorithm to a protein structural alignment problem [9]. Finally, in Section 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

4.

$\mathcal{H}_1 = \{D\}$, $c =$ 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}

Figure 1: Sketch of the original `DIRECT` Algorithm.

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 k-th 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. The minimum of f over all the centers of the final partition, and the corresponding centers, provide an approximate solution to the problem. The structure of `DIRECT` is outlined in Figure 1.

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 or "divide-the-best" algorithms, we refere the interested reader to [10, 11, 12].

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 reference [3]) where local minimizations starting from the cetroids 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.
In particular, given a point $\tilde{x} \in (0,1)^n$, let $y = T_{\tilde{x}}(x)$ be defined by

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

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 stops producing an estimate $x_{\min} \in (0,1)^n$ of the global minimum point. If the global minimum estimate value $f(x_{\min})$ is not sufficiently close to the optimal value f^*, we propose to 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.

6.

Algorithm DIRMIN-TL

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

Repeat

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

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

(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 $\mathcal{N} = \mathcal{N} \cup \{x \in \mathcal{W} : x \text{ is "promising"}\}$ where a stationary point x is promising when $f(x)$ is "close to" $f(x_{\min})$ and 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 $\mathcal{O} = \mathcal{O} \cup \{\tilde{x}\}$.

2. A plain modification of algorithm DIRMIN-TL

In principle, Algorithm DIRMIN-TL (from reference [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.

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 reference [3] the following proposition concerning some minimal assumptions needed by an iterative algorithm to be attracted by a global minimum point.

Proposition 2.2 ([13]) *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 2.2, we can thus propose a straightforward modification of `DIRMIN-TL` for black-box optimization that consists in substituting derivative-based with derivative-free local minimizations. Examples of such algorithms are present in the literature [14, 15].
Here we represent such an algorithm as

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

where x_0 is the starting point of the minimization, α_0 represents an estimate of the stationarity measure [16] 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 A for the interested reader). We denote by $\hat{\alpha}_{\max} = \max_{i=1,\ldots,n} \hat{\alpha}_i$ the stationarity measure of $\hat{x}$ [16].

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 [17] 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 an 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, removing and keeping columns and boxes ordered can be done very efficiently. In particular, set I_k^w is computed by applying the Jarvis's march [18] just to the top elements of the list of boxes of each column (see Figure 2), which is of limited cardinality.

2.2. Numerical results with `DIRMIN-TL`

We applied this simple modification of Algorithm `DIRMIN-TL` to a set of global optimization problems from references [3, 19, 20] (see Table 5 in Appendix B for problem dimensions and optimal values). More precisely, inside `DIRMIN-TL` we allow the generation of at most $50000 \times n$

hyperrectangles by algorithm DIRMIN and we set the maximum number of restarts to 100. Furthermore, we use the following stopping criterion

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

where f^* is the 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
- 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.

Table 1: Results of DIRMIN-TL

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Test problems from reference [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

continued on next page

Table 1 – continued from previous page

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
$15n$ 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	78663	858	3089
Griewank mod.	5	6.2154e-10	490515	1896	11553
Griewank mod.	10	9.2333e-10	411178	753	7359
Griewank mod.	20	9.4514e-10	942	1	41
Pinter	2	3.3940e-09	170	2	7
Pinter	5	1.5414e-06	25618	101	629
Pinter	10	6.0700e-05	444607	1048	10259
Pinter	20	7.1553e-05	42682351	55392	1000478
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	10205	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	178195	503	7531
Dixon Price	25	7.6712e-08	283077766	339219	12523913
Dixon Price	50	**6.6667e-01**	36182567	21107	1863678
Easom	2	-1.0000e+00	131165	2190	6579
Michalewics	2	-1.8013e+00	69	1	5
Michalewics	5	-4.6877e+00	130058	920	6137
Michalewics	10	-9.6601e+00	21699660	67701	662976
Rastrigin	2	1.9443e-07	336	4	13
Rastrigin	10	9.7216e-07	12751	30	383
Rastrigin	30	2.9165e-06	280683	221	8491
Rastrigin	50	4.8608e-06	1265672	601	37703
Test problems from reference [19]					
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	225066	62	387
perm1	2	1.0801e-06	272	1	5
perm1	5	8.9149e-05	1568231384	582059	4009631
perm2	2	0.0000e+00	62	1	5
perm2	5	6.3056e-07	57766	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	201763	743	6013

continued on next page

Table 1 – continued from previous page

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
schwefel	10	2.7574e-07	24909898	45504	500094
schwefel	20	5.5133e-07	431426380	496122	10407001
Test problems form reference [20]					
Sphere	10	-1.4000e+03	573	1	21
Rot. Elliptic	10	-1.2999e+03	5642213969	1261645	14213501
Rot. Discus	10	-1.2000e+03	47680	31	397
Rot. Bent Cigar	10	**-4.1537e+02**	10584086678	2109744	31004082
Different Powers	10	-1.0000e+03	589	1	21
Rot. Rosenbrock	10	-8.9997e+02	36686	9	145
Rot. Schaffers (F7)	10	-7.9996e+02	374242440	91928	1014337
Rot. Ackley	10	**-6.8008e+02**	943365696	2134932	27501329
Rot. Weierstrass	10	**-5.9914e+02**	359339196	871942	9000562
Rot. Griewank	10	-4.9999e+02	14487	16	237
Rastrigin	10	-4.0000e+02	52620517	116714	1213659
Rot. Rastrigin	10	**-2.9801e+02**	3208062726	4780133	50004016
Non-Continuous Rot.	10	**-1.9801e+02**	1939249826	4921080	50004772
Schwefel	10	**-9.3295e+01**	1597692027	4458022	50004906
Rot. Schwefel	10	**1.1512e+02**	189035848	491308	5000954
Rot. Katsuura	10	**2.0004e+02**	204960561	570002	7000330
Lunacek Bi-Rastrigin	10	3.0003e+02	87808133	205922	2009957
Rot. Lunacek Bi-Rast	10	**4.0202e+02**	5511932601	5700552	50003148
Expanded Griewank +	10	5.0005e+02	12087229	3048	28155
Expanded Schaffer (F	10	**6.0154e+02**	2396769984	3454923	50000962
Comp. Function 1	10	7.0000e+02	104003	157	1645
Comp. Function 2	10	**9.0284e+02**	1655305428	4450965	50006510
Comp. Function 3	10	**1.0268e+03**	265301785	645319	6500803
Comp. Function 4	10	**1.0435e+03**	363888151	704658	7500899
Comp. Function 5	10	**1.2019e+03**	2518222053	4926281	50003832
Comp. Function 6	10	**1.2288e+03**	1265561798	936028	10501167
Comp. Function 7	10	**1.4717e+03**	118906294	282856	3000676
Comp. Function 8	10	1.4000e+03	33824697	49859	412627

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 [19]. Finally, the third part of the table contains the test problems recently proposed in [20] 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:

- the derivative-free version of Algrithm `DIRMIN-TL` fails only on one problem from reference [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., [13] for smooth problems and [21] for nonsmooth optimization).

- `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. Hence, in our further experimentations we will not consider these "easy" problems.

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.

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

3. A new distributed 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 [13], we update during the iterations a working set, of dimension $n_{wks} = 100n$, of "open" local minimizations that are carried out in a distributed fashion. The idea is to perform until the end only a limited number of local minimizations, focusing on the most "promising" ones. In particular, 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, that is replaced.

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.

12.

Algorithm DDF-DIRMIN

$\mathcal{H}_1 = \{D\}$, $c =$ center of D, $f_{min} = f(c)$, $X_{min} = \{c\}$, $tol > 0$, $\alpha_{\max} > tol$, $kmax \geq 1$, $\mathcal{W}_1 = \emptyset$, $\alpha_1 \in \Re^n$, $n_{wks} \geq 1$, $k = 1$

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 $(\hat{c}_i, \hat{\alpha}_i) = DF(c_i, d_i, \alpha_{\max}, kmax)$

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 $\max_{j=1,\ldots,n} \alpha_j \leq tol$.

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

until (stopping criterion satisfied)

return f_{min}, X_{min}

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

Table 2: Results of DDF-DIRMIN

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Test problems for reference [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	274190	24104	133129
Griewank mod.	10	1.0188e-05	14724	963	9495
Pinter	2	1.6568e-05	197	25	105
Pinter	5	9.8179e-05	1843	214	1255

continued on next page

Table 2 – continued from previous page

Problem	n	$f(\bar{x})$	n.f.	n.loc.	n.int.
Pinter	10	9.5931e-05	16393	1310	12939
Pinter	20	9.9644e-05	107872	4667	97053
Griewrot2	50	-1.7998e+02	8468	71	6491
Ackley	2	3.9968e-15	407	42	145
Dixon Price	10	8.9499e-05	12464	679	9833
Dixon Price	25	**6.6667e-01**	5492616	134409	5001820
Dixon Price	50	**6.6667e-01**	1940253	21107	1863368
Easom	2	-1.0000e+00	11043	2198	6625
Michalewics	5	-4.6876e+00	9538	983	6523
Michalewics	10	-9.6595e+00	849039	67724	663164
Rastrigin	2	1.9443e-07	181	22	87
Rastrigin	10	9.7216e-07	1188	72	817
Rastrigin	30	2.9165e-06	13542	346	12217
Rastrigin	50	4.8608e-06	51820	834	49023
Test problems for reference [19]					
Colville	4	7.3521e-05	1335	167	883
perm1	5	**7.6076e-04**	36624683	3647938	25003738
perm2	5	8.5173e-05	5275	597	3611
schwefel	2	7.4746e-05	515	47	165
schwefel	5	7.7473e-05	11670	768	6177
schwefel	10	2.5060e-05	822302	45537	500522
schwefel	20	4.4414e-05	12217489	496164	10408019
Test problems for reference [20]					
Rot. Elliptic	10	-1.3000e+03	581930281	4448993	50013308
Rot. Discus	10	-1.2000e+03	18501	481	4751
Rot. Bent Cigar	10	**-1.0907e+03**	122143170	166257	2500131
Rot. Rosenbrock	10	-9.0000e+02	96883	144	1505
Rot. Schaffers (F7)	10	-7.9992e+02	12735661	61068	668590
Rot. Ackley	10	**-6.8002e+02**	15935575	350139	4500239
Rot. Weierstrass	10	**-5.9919e+02**	12565878	657155	6500311
Rot. Griewank	10	-4.9995e+02	16384	533	4865
Rastrigin	10	-4.0000e+02	22445131	146194	1531248
Rot. Rastrigin	10	**-2.9801e+02**	87462231	4774990	50003974
Non-Continuous Rot.	10	**-1.9801e+02**	83304804	4800436	50004334
Schwefel	10	**-9.3233e+01**	353592387	4433733	50004226
Rot. Schwefel	10	**1.5011e+02**	5107995	363772	3500411
Rot. Katsuura	10	**2.0003e+02**	21832018	490174	6000264
Lunacek Bi-Rastrigin	10	3.0003e+02	45057438	205910	2009909
Rot. Lunacek Bi-Rast	10	**4.0222e+02**	391928534	5702225	50003450
Expanded Griewank +	10	5.0005e+02	1149771	3995	36901
Expanded Schaffer (F	10	**6.0185e+02**	685417439	3499448	50001068
Comp. Function 1	10	7.0000e+02	29144	189	1933
Comp. Function 2	10	**9.0284e+02**	362203872	4455047	50006616
Comp. Function 3	10	**1.0323e+03**	4701450	293292	3000406
Comp. Function 4	10	**1.0291e+03**	6281748	389132	4500485
Comp. Function 5	10	**1.2025e+03**	85773668	4966625	50003834
Comp. Function 6	10	**1.2344e+03**	21781707	286120	3000222
Comp. Function 7	10	**1.5882e+03**	717640	45111	500041
Comp. Function 8	10	**1.5000e+03**	2231849	161795	1500363

In order to better evaluate the savings in terms of function evaluations, in Figure 3 we plot

the cumulative distribution function $\rho(\tau)$ defined as:

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

where $\mathcal{P}$ is the set of test problems, and $\mathrm{nf}_{p,1}$ ($\mathrm{nf}_{p,2}$) is the number of function evaluations required by `DIRMIN-TL` (`DDF-DIRMIN`) 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 `DDF-DIRMIN` stops in less than half the number of function evaluations required by `DIRMIN-TL` on about 87% of the problems. Obviously, this greater efficiency has a price: indeed, Algorithm `DDF-DIRMIN` fails on 20 problems (out of 58) whereas Algorithm `DIRMIN-TL` only fails on 17 problems.

4. A new algorithm using the plateau modification function

Now, we try to improve the reliability of `DIRMIN-TL`, i.e. its ability to locate the global optimum, without worrying to much about the efficiency. To this aim, we first define the following "plateau" modification of the objective function [6]:

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

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. [6, 7, 8]) which, under the stated assumptions, is bounded from below. We define a new algorithm, that we call `DIRFOB`, that, roughly speaking, consists in applying algorithm `DIRECT` to the global minimization of the "plateau" function $\tilde{f}$. In Algorithm `DIRFOB` we maintain the restarting technique used in `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 [12], this property follows from

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

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

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

Repeat

(S.1) Apply DIRECT to $\min_{y \in [0,1]^n} \tilde{f}_{\tilde{x}}(y)$ until $|\mathcal{H}_k| \leq maxint$ 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 **cycle**.

(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 on the set of 58 difficult test problems.

Table 3: Results of DIRFOB

Problem	n	$f(\bar{x})$	n.f.	n.int.
Test problems for reference [3]				
Schubert	2	-1.8673e+02	2315	205
Schub. pen. 1	2	-1.8673e+02	6316	207
Schub. pen. 2	2	-1.8673e+02	3868	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	14746	233
Griewank mod.	5	4.3145e-10	154689	845
Griewank mod.	10	1.4426e-09	182128	1009
Pinter	2	3.3940e-09	90	5
Pinter	5	1.5150e-06	10547	521
Pinter	10	1.5090e-06	24151879	89033
Pinter	20	1.2913e-05	822928421	1569822
Griewrot2	50	-1.7998e+02	5104	101
Ackley	2	5.6461e-05	466	151
Dixon Price	10	9.3893e-08	137441	1087
Dixon Price	25	1.3575e-08	5731881	5419
Dixon Price	50	4.8111e-08	1269313104	460873
Easom	2	-1.0000e+00	318161	8829
Michalewics	5	-4.6877e+00	23308	533
Michalewics	10	-9.6601e+00	30065636	178221
Rastrigin	2	7.4494e-07	1856	209
Rastrigin	10	4.9073e-08	12010	1103
Rastrigin	30	2.7650e-06	6145	3223
Rastrigin	50	4.4339e-06	10277	5355

continued on next page

Table 3 – continued from previous page

Problem	n	$f(\bar{x})$	n.f.	n.int.
Test problems for reference [19]				
Colville	4	2.8539e-06	13120363	60281
perm1	5	**9.3933e-04**	7279772213	25003136
perm2	5	3.6115e-05	168322925	560003
schwefel	2	5.5145e-08	525	121
schwefel	5	1.3907e-07	91223	527
schwefel	10	2.7814e-07	483319	1237
schwefel	20	5.5629e-07	4295690	5361
Test problems for reference [20]				
Rot. Elliptic	10	-1.3000e+03	121196868750	34503835
Rot. Discus	10	-1.2000e+03	2611405	1925
Rot. Bent Cigar	10	-1.1000e+03	4059627002	1017617
Rot. Rosenbrock	10	-9.0000e+02	110939	1067
Rot. Schaffers (F7)	10	-7.9993e+02	425933887	527362
Rot. Ackley	10	-6.9997e+02	650470093	3501090
Rot. Weierstrass	10	-5.9997e+02	263655451	1018829
Rot. Griewank	10	-4.9998e+02	22073	1047
Rastrigin	10	-4.0000e+02	202714074	511630
Rot. Rastrigin	10	-3.0000e+02	8398182	20159
Non-Continuous Rot.	10	-2.0000e+02	3864394	13701
Schwefel	10	-1.0000e+02	186574187	522190
Rot. Schwefel	10	1.0000e+02	546425335	1501278
Rot. Katsuura	10	**2.0003e+02**	610000645	3000142
Lunacek Bi-Rastrigin	10	3.0000e+02	441877163	1291487
Rot. Lunacek Bi-Rast	10	4.0000e+02	11556220	28031
Expanded Griewank +	10	5.0004e+02	31844643	49393
Expanded Schaffer (F	10	6.0006e+02	54060331	138325
Comp. Function 1	10	7.0000e+02	196612	1041
Comp. Function 2	10	8.0000e+02	189617796	519686
Comp. Function 3	10	9.0000e+02	97308583	265987
Comp. Function 4	10	1.0000e+03	553844533	1501776
Comp. Function 5	10	**1.2000e+03**	2008600802	5500523
Comp. Function 6	10	1.2000e+03	1580140782	2017773
Comp. Function 7	10	**1.4000e+03**	3127930717	8000486
Comp. Function 8	10	1.4000e+03	1541217	3035

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

However, this is not always the case as it emerges from Figure 4 where we plot function $\rho(\tau)$ for the comparison among `DIRMIN-TL` and `DIRFOB`. In particular, we plot $\rho(\tau)$ for $\tau \in [0, 1]$ (left side of Figure 4) and for $\tau \in [1, 60]$ (right side of Figure 4). It can be seen that `DIRFOB` requires a number of function evaluation not greater than that required by `DIRMIN-TL` on approximately half of the test problems (see, e.g., left side of Figure 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 $n = |P| \quad \text{and} \quad m = |Q|$ 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 [22] and $g(a) = f(\psi(a), a)$. Then, the surface alignment problem can be posed as

$$\min_{a \in \Theta} g(a). \tag{3}$$

Every global solution a^* of (3) is, by definition, a solution such that $f(\psi(a^*), a^*) \leq f(\psi(a), a)$, for all $a \in \Theta$. Problem (3) 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` is the more robust one, this is the code that we employ to find correct alignments on the set of 19 proteins used in [9]. The proteins all bind ligand ATP and are from different families according to the structural classification SCOP [23].
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` and CO (i.e. the algorithm proposed in [9]), respectively.

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

	DIRFOB		CO	
Protein Pair	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 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 better separates proteins in the same SCOP family as 1atp, from the others. Indeed, DIRFOB 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 and CO between different SCOP families are 32 and 28 atoms, respectively.

6. Conclusions

In the paper we focused 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. In particular, we proposed three different DIRECT-type algorithms which makes efficient use of derivative-free local searches combined with nonlinear transformations of the feasible domain and, possibly, of the objective function. 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 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 devised two more algorithms trying to improve both the efficiency and reliability of DIRMIN-TL. More precisely, we showed that algorithm DDF-DIRMIN is far more efficient than DIRMIN-TL in terms of required function evaluations at the expense of a reduced reliability. Then, we tried to improve on the reliability

and came up with algorithm DIRFOB which is indeed far more reliable than both DIRMIN-TL and DDF-DIRMIN though it is generally more expensive than the first two codes.
Finally, we reported the results obtained by Algorithm DIRFOB on a difficult protein structural alignment problem and show that it performs better than a method recently proposed in the literature.

Appendix

A. The derivative-free local algorithm

In this section we report the sketch of a derivative-free procedure for unconstrained local minimization [14].

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

In particular, the actual implementation of Algorithm DF that we use is based on the one proposed in [14]. As for the parameters, the initial stepsizes $\alpha(c^i)$, with c^i center of D_i, are equal to the boxes of the hyperrectangles D_i divided by two, the parameter tol is set to 10^{-4}, and $kmax$ is equal to 5000.

B. 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.

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
Michalewics	2	-1.8013E+00
Michalewics	5	-4.6876E+00
Michalewics	10	-9.6602E+00
Rastrigin	2,10,30,50	0.0000E+00
Problems from [19]		
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
schwefel	2,5,10,20	0.0000E+00
Problems from [20]		
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

continued on next page

Table 5 – continued from previous page

Problem	n	f^*
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.	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-Rast	10	4.0000E+02
Expanded Griewank +	10	5.0000E+02
Expanded Schaffer (F	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

References

[1] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[2] D. R. Jones. DIRECT global optimization. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of optimization*, pages 725–735. Springer, 2009.

[3] G. Liuzzi, S. Lucidi, and V. Piccialli. A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Computational Optimization and Applications*, 45:353–375, 2010.

[4] T.D. Panning, L.T. Watson, N.A. Allen, C.A. Shaffer, and J.J. Tyson. Deterministic global parameter estimation for a budding yeast model. In J.A. Hamilton Jr., R. MacDonald, and M.J. Chinni, editors, *Proc. 2006 Spring Simulation Multiconf., High Performance Computing Symp.*, pages 195–201. Soc. for Modeling and Simulation Internat., San Diego, CA, 2006.

[5] T.D. Panning, L.T. Watson, N.A. Allen, K.C. Chen, C.A. Shaffer, and J.J. Tyson. Deterministic parallel global parameter estimation for a model of the budding yeast cell cycle. *Journal of Global Optimization*, 40:719–738, 2008.

[6] D.J. Wales and J.P.K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[7] R.H. Leary. Global optimization on funneling landscapes. *Journal of Global Optimization*, 18(4):367–383, 2000.

[8] M. Locatelli and F. Schoen. Efficient algorithms for large scale global optimization: Lennard-jones clusters. *Computational Optimization and Applications*, 26(2):173–190, 2003.

[9] P. Bertolazzi, C. Guerra, and G. Liuzzi. A global optimization algorithm for protein surface alignment. *BMC Bioinformatics*, 11:488–498, 2010.

[10] Yaroslav D. Sergeyev. On convergence of "Divide the Best" global optimization algorithms. *Optimization*, 44(3):303–325, 1998.

[11] Yaroslav D. Sergeyev and Dmitri E. Kvasov. Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM J. Optim.*, 16(3):910–937, 2006.

[12] G. Liuzzi, S. Lucidi, and V. Piccialli. A partition-based global optimization algorithm. *Journal of Global Optimization*, 48:113–128, 2010.

[13] G. Liuzzi, S. Lucidi, V. Piccialli, and A. Sotgiu. A magnetic resonance device designed via global optimization techniques. *Mathematical Programming*, 101(2):339–364, 2004.

[14] S. Lucidi and M. Sciandrone. On the global convergence of derivative free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13:97–116, 2002.

[15] S. Lucidi and M. Sciandrone. A derivative-free algorithm for bound constrained optimization. *Computational Optimization and Applications*, 21(2):119–142, 2002.

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

[17] J. He, L.T. Watson, N. Ramakrishnan, C.A. Shaffer, A. Verstak, J. Jiang, K. Bae, and W.H. Tranter. Dynamic data structures for a direct search algorithm. *Computational Optimization and Applications*, 23:5–25, 2002.

[18] R.A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2:18–21, 1973.

[19] Abdel-Rahman Hedar. Test problems for unconstrained optimization. `http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm`.

[20] J. J. Liang, B-Y. Qu, P. N. Suganthan, and A. G. Hernndez-Daz. Problem definitions and evaluation criteria for the cec 2013 special session and competition on real-parameter optimization. Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, 2013.

[21] G. Di Pillo, S. Lucidi, and F. Rinaldi. A derivative-free algorithm for constrained global optimization based on exact penalty functions. *Journal of Optimization Theory and Applications*, 2013. Submitted.

[22] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Mach. Intelligence*, 14:239–255, 1992.

[23] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

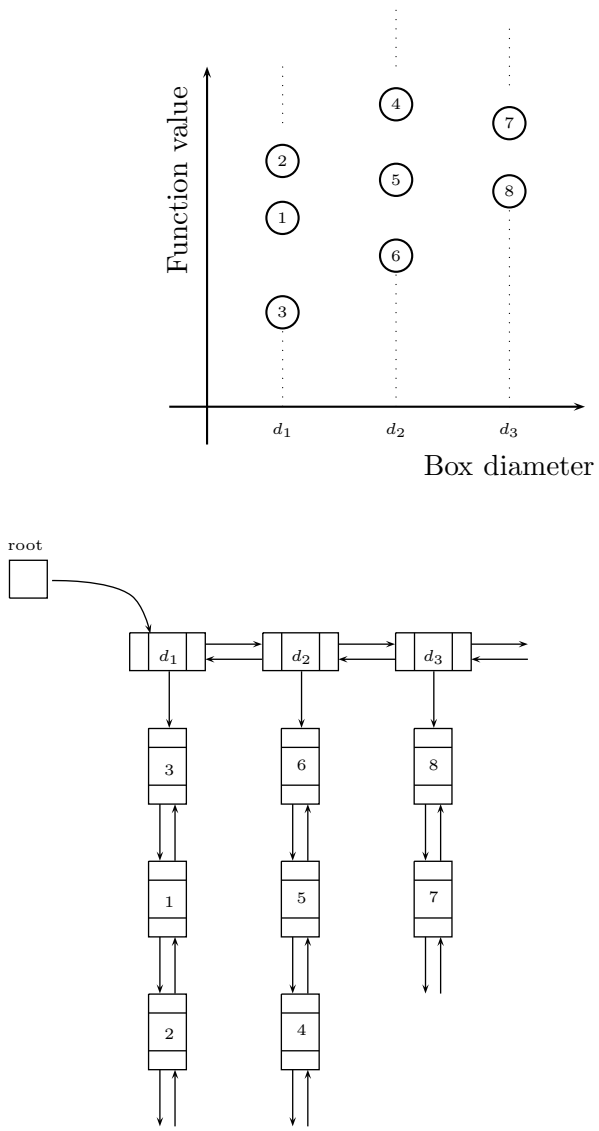

Figure 2: Potentially optimal hyperintervals

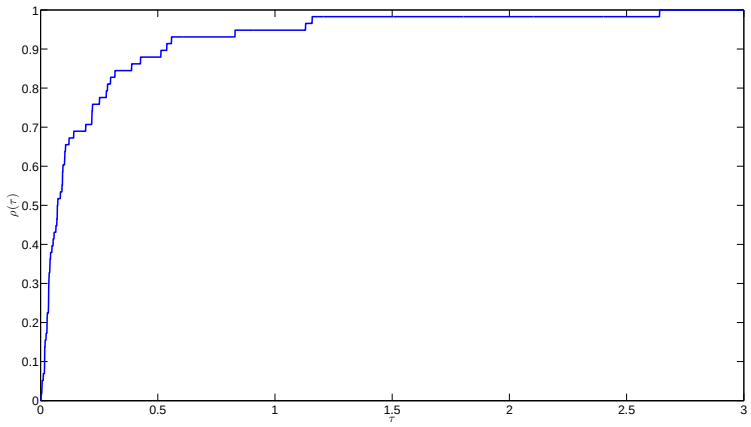

Figure 3: Comparison of `DIRMIN-TL` and `DDF-DIRMIN` by means of the cumulative distribution function $\rho(\tau)$

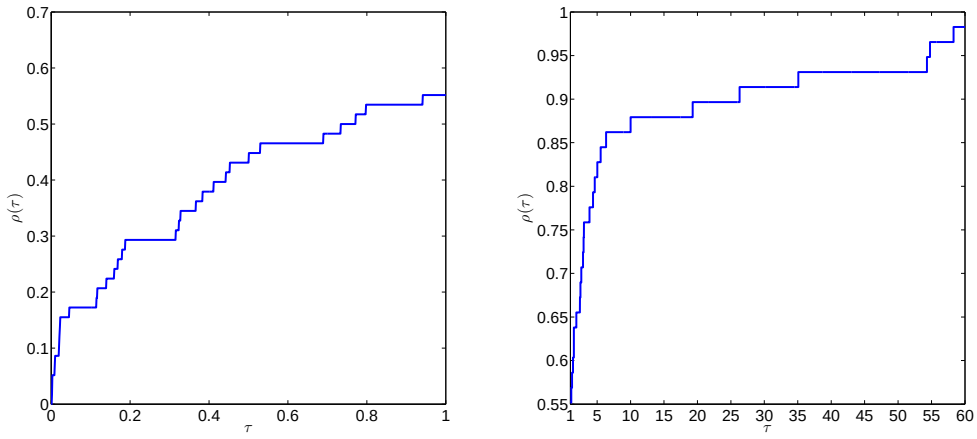

Figure 4: Comparison of `DIRMIN-TL` and `DIRFOB` by means of the cumulative distribution function $\rho(\tau)$