CrossMark

Filter-based DIRECT method for constrained global optimization

M. Fernanda P. Costa[1] · **Ana Maria A. C. Rocha[2]** ·
Edite M. G. P. Fernandes[3]

Abstract This paper presents a DIRECT-type method that uses a filter methodology to assure convergence to a feasible and optimal solution of nonsmooth and nonconvex constrained global optimization problems. The filter methodology aims to give priority to the selection of hyperrectangles with feasible center points, followed by those with infeasible and non-dominated center points and finally by those that have infeasible and dominated center points. The convergence properties of the algorithm are analyzed. Preliminary numerical experiments show that the proposed filter-based DIRECT algorithm gives competitive results when compared with other DIRECT-type methods.

1 Introduction

In this paper, constrained global optimization (CGO) problems are addressed by proposing the integration of a filter methodology, as outlined in [19], into a DIRECT-type method to globally solve nonsmooth and nonconvex constrained optimization problems. The mathematical formulation of the problem is:

✉ M. Fernanda P. Costa
  mfc@math.uminho.pt

  Ana Maria A. Rocha
  arocha@dps.uminho.pt

  Edite M. G. Fernandes
  emgpf@dps.uminho.pt

[1] Centre of Mathematics, Department of Mathematics and Applications, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

[2] Algoritmi Research Centre, Department of Production and Systems, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

[3] Algoritmi Research Centre, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

$$\min_{x \in \Omega} f(x)$$
$$\text{subject to} \quad h(x) = 0 \tag{1}$$
$$g(x) \leq 0$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are nonlinear continuous functions and $\Omega = \{x \in \mathbb{R}^n : -\infty < l_i \leq x_i \leq u_i < \infty, i = 1, \ldots, n\}$. The feasible set of points that satisfy all the constraints is denoted by $F = \Lambda \cap \Omega$ where $\Lambda = \{x \in \mathbb{R}^n : h(x) = 0 \text{ and } g(x) \leq 0\}$. Differentiability and convexity are not assumed and therefore many local minima may exist in F. The most used and popular framework to solve problem (1) considers a penalty term, which depends on a constraint violation measure and is combined with the objective function to give the so-called penalty function. The penalty term aims to penalize f when a candidate solution, $\bar{x}$, is infeasible, i.e., when $\bar{x} \notin F$. Penalty functions require the use of a positive penalty parameter that aims to balance function and constraint violation values. Setting an initial value for the penalty parameter and tuning its values throughout the iterative process are important issues that should be addressed in the algorithm, since the performance is greatly affected by their values. In some cases, the optimal solution of the problem is attained only when the penalty parameter approaches infinity. With exact penalty functions, there exists a positive threshold such that for all parameter values smaller than that threshold the CGO and an equivalent bound constrained (or unconstrained) problem have the same global minimizers. An exact penalty approach combined with the DIRECT algorithm (for solving the bound constrained global optimization subproblems) that uses the derivatives of the involved functions for the penalty parameter updating is proposed in [11]. Later, the authors in [12] use the exact penalty framework presented in [11] and enrich the global search activity of DIRECT by using a derivative-free local search algorithm, in a multistart framework, to speed the convergence, as well as, a derivative-free strategy for the penalty parameter updating. Augmented Lagrangian functions are penalty functions for which a finite penalty parameter value is in general sufficient to guarantee convergence to the solution of the constrained problem [2]. Augmented Lagrangians for CGO problems have been implemented and their convergence properties have been derived. For example, in [4], the exact α-branch-and-bound method is used to find approximate global solutions to the subproblems. In [5], the global convergence to a minimum is guaranteed using a nonmonotone penalty parameter tuning and gradient-based approaches for solving the bound constrained subproblems.

The use of the filter method to guarantee sufficient progress towards feasible and optimal solutions of nonlinear constrained optimization has its origin in [19]. The filter technique has been used since then in the context of a variety of problems, nonlinear optimization [38], complementarity problems [44], nonlinear equations [22], inexact line search [29], trust region [43], nonmonotone strategies [42,44], to name a few. Filter with stochastic methods are new ideas recently proposed [24,37,39]. The issue concerned with the use of the filter method in the context of derivative-free optimization has not been frequently addressed [1,7,8,15,28,36].

In this paper, we aim to integrate the filter methodology into a DIRECT-type method algorithm for solving CGO problems. The derivative-free and deterministic global optimizer DIRECT [27] is a partition-based algorithm [34] that identifies potential optimal hyperrectangles and divides them further to produce finer and finer partitions of the hyperrectangles [17,18,21,26,30,33]. It has been shown that the set of points sampled by the DIRECT algorithm is dense in the search space [27]. Consequently, if f is continuous in a neighborhood of the global minimum, the DIRECT algorithm can sample a point that is arbitrarily near the global minimum. This guaranteed convergence makes it a reliable and interesting derivative-

free global optimizer. In the present study, the use of the filter set aims to give priority to the selection (for further division) of hyperrectangles with feasible center points that are more promising relative to the objective function, followed by the selection of hyperrectangles with infeasible and non-dominated center points, promising with respect to the constraint violation measure. Finally, the most promising hyperrectangles with infeasible and dominated center points are selected by means of the constraint violation measure. The dimensions for further division are selected based on a particular order that favors feasible trial points, and infeasible and non-dominated trial points, against infeasible and dominated trial points.

The remainder of the paper proceeds as follows. Section 2 introduces the DIRECT method, the filter set methodology, presents the new filter-based DIRECT algorithm and its convergence properties. Section 3 contains the results of all the numerical experiments and the conclusions are summarized in Sect. 4.

2 Filter-based DIRECT

The DIRECT (DIviding RECTangles) algorithm has been originally proposed to solve bound constrained global optimization (BCGO) problems,

$$\min_{x \in \Omega} f(x) \tag{2}$$

where f is assumed to be a continuous function, by producing finer and finer partitions of the hyperrectangles generated from Ω [27]. The algorithm is a modification of the standard Lipschitzian approach, in which f must satisfy the Lipschitz condition

$$|f(x_1) - f(x_2)| \le K \|x_1 - x_2\| \text{ for all } x_1, x_2 \in \Omega,$$

where the Lipschitz constant $K > 0$ is viewed as a weighting parameter that indicates how much emphasis to place on global versus local search. DIRECT is known as a global optimizer since it is able to explore potentially optimal regions to converge to the global optimum and to avoid being trapped in local optima. It is a deterministic method that does not require any derivative information or the value of the Lipschitz constant [23]. It searches (locally and globally) by dividing all hyperrectangles that satisfy the following criteria, defining a potentially optimal hyperrectangle (POH):

Definition 1 Given the partition $\{H^i : i \in I\}$ of Ω, let ϵ be a positive constant and let $f_{\min}$ be the current best function value. A hyperrectangle j is said to be potentially optimal if there exists some rate-of-change constant $\hat{K} > 0$ such that

$$\begin{aligned} f(c_j) - \hat{K} d_j &\le f(c_i) - \hat{K} d_i, \text{ for all } i \in I \\ f(c_j) - \hat{K} d_j &\le f_{\min} - \epsilon |f_{\min}| \end{aligned} \tag{3}$$

where c_j is the center and d_j is a measure of the size of the hyperrectangle j.

In [27], d_j is the distance from c_j to its vertices. The use of $\hat{K}$ intends to show that it is not the Lipschitz constant. The second condition in (3) aims to prevent the algorithm from searching locally a region where very small improvements are obtained. However, the results in the literature show that DIRECT is fairly insensitive to the setting of the balance parameter ϵ, and good results are obtained for values ranging from 10^{-3} to 10^{-7} [27]. The parameter ϵ also prevents the hyperrectangle with best function value from being a POH. Identifying the set of POH can be regarded as a problem of finding the extreme points on the lower right

convex hull of a set of points in the plane [27]. A 2D-plot can be used to identify the set of POH. The horizontal axis corresponds to the size of hyperrectangle and the vertical axis corresponds to the f value at the center of hyperrectangle. Further details concerning the DIRECT algorithm are available in [16,20,23]. In [27], the DIRECT algorithm for solving the BCGO problem is described by the six main steps presented in Algorithm 2.1.

Algorithm 2.1 DIRECT

---

1. **Initialization**. Normalize the search space Ω to be an n-dimensional unit hypercube H^0. Sample the center point c_1 of this hypercube and evaluate $f(c_1)$. Initialize $f_{min} = f(c_1)$, set (function evaluation counter) $FE = 1$ and (iteration counter) $k = 0$. Set $I_0 = \{0\}$. (*The algorithm works in this normalized space, referring to the original space only when objective function values are required.*)

2. **Selection**. Given the partition $\{H^i : i \in I_k\}$ of Ω, identify the set $O_k \subseteq I_k$ of POH that are subregions of H^k (see Definition 1).

3. **Sampling**. For any hyperrectangle $j \in O_k$, identify the set L_k^j of dimensions with the maximum size, s_M. (*Dividing the hyperrectangle only along its longest dimension(s) assures that the maximal size(s) of the hyperrectangle is(are) decreased.*) Let $|L_k^j|$ be the cardinal of L_k^j. Let $\delta = 1/3 s_M$. Sample the function at points $c_j \pm \delta e_i$, for all $i \in L_k^j$, where c_j is the center of the hyperrectangle and e_i is the ith unit vector.

4. **Division**. Divide the hyperrectangle j containing c_j into thirds along the dimensions in L_k^j, using the following "order". First divide along the dimension m such that $w_m = \min w_i$ (for all $i \in L_k^j$) where $w_i = \min\{f(c_j + \delta e_i), f(c_j - \delta e_i)\}$ and then continue dividing until the dimension with the highest w_i. Set $FE = FE + 2|L_k^j|$.

5. **Iteration**. Set $O_k = O_k \setminus \{j\}$. Set $I_k = I_k \bigcup_{i=1,...,2|L_k^j|+1}\{j_i\} \setminus \{j\}$ and update $f_{min} = \min_{i \in I_k} f(c_i)$. If $O_k \neq \emptyset$ go to **3**.

6. **Termination**. Set $k = k + 1$. If stopping criteria are met, stop; otherwise go to **2**.

2.1 DIRECT-type methods

Partition-based algorithms search for the global optima by using sequences of partitions of the region Ω [34]. At each iteration, some sets are chosen to be further partitioned according to different criteria. Some criteria rely on *a priori* knowledge of the objective function, and others use a selection strategy that does not require any *a priori* knowledge on the objective function [25]. The DIRECT algorithm and some of its variants are particular cases of a partition-based algorithm (PBA) for GO [32]. Given any accuracy, a PBA can guarantee to find a solution that is sufficiently near the global optimum of problem (2). It is efficient, in the sense that it finds the regions of local minima in a few iterations. However, it requires the sampling of a dense set of points to obtain a solution. Moreover, the computational effort greatly increases as the accuracy increases.

In [34], PBA are explored and two new strategies for selecting the hyperrectangles (for further partitioning) are presented. A partition strategy that does not require any knowledge about the objective function, although it exploits information on the objective function gathered during the iterative process is proposed. The main difference between the therein called partition-based on estimation (PBE) algorithm and the DIRECT methods [18,21,26,27] is that PBE iteratively estimates the global minimum value rather than the Lipschitz constant.

The robust DIRECT (RDIRECT) algorithm presented in [31] uses a bilevel strategy into a modified DIRECT algorithm to overcome the difficulty related to the slow convergence

to achieve a high degree of accuracy. Since the second condition in (3) causes the original DIRECT to be sensitive to the additive scaling of the objective function (as shown in [18]), the robust version of DIRECT, meaning that it is insensitive to linear scaling of the objective function, relies on the following definition of POH:

Definition 2 [31] Given $\epsilon > 0$ and the index set of the hyperrectangles I. Let $f_{\min}$ be the known minimal function value, and c_i, d_i be the center and the size of the ith hyperrectangle. Denote f_{cc} as any convex combination of the objective function values gathered during previous iterations by RDIRECT. If there is a constant $\hat{K} > 0$, such that the following inequalities hold,

$$f(c_j) - \hat{K}d_j \le f(c_i) - \hat{K}d_i, \quad \text{for all } i \in I$$
$$f(c_j) - \hat{K}d_j \le f_{\min} - \epsilon|f_{\min} - f_{cc}|, \tag{4}$$

then the jth hyperrectangle is a POH.

The convex combination is given by

$$f_{cc} = \sum_{k=1}^{k_f} \alpha_k f(c_k) \text{ with } \alpha_k \in [0, 1] \text{ and } \sum_{k=1}^{k_f} \alpha_k = 1$$

where $\{f(c_k)\}_{k=1}^{k_f}$ are the gathered f values. Possible choices for f_{cc} include the maximal value, the minimal value, the average or the median value of the gathered f values [31]. To improve the inefficient behavior of the PBA, the authors in [32] combine the PBA with a multigrid algorithm. The work in [33] applies a multigrid approach to the robust version of DIRECT algorithm [31], obtaining a multilevel robust DIRECT (MrDIRECT) algorithm. Although additional parameters are required by this new version, the authors show that the MrDIRECT is insensitive to the parameters.

Efficient and robust global optimization algorithms balance global and local search capabilities. A DIRECT-type algorithm that is strongly biased towards local search is presented in [21]. The authors state that the algorithm does well for small problems with a single global minimizer and only a few local minimizers. It is common knowledge that DIRECT gets quickly close to the global optimum but requires a lot more of computational effort to reach a high accuracy solution. To overcome this inconvenience, Jones in [26] suggests the use of a local optimizer by alternating between running DIRECT in the usual way and using a local search algorithm starting from the solution produced by DIRECT. In fact, DIRECT can be considered a good technique to select starting points for a local search method. The local optimizer aims to reduce the value of $f_{\min}$ which, by affecting the selection of hyperrectangles, improves the global search capability. At the same time, choosing a larger value of ϵ will directly improve the global search [33].

The DIRECT version presented in [26] has been extended to handle nonlinear inequality constraints by using an auxiliary function that combines information on the objective and constraint functions. The value of the auxiliary function is zero at the global minimizer and is positive at any point that has $f > f^*$ or is infeasible, where f^* is the global minimum. The main idea relies on modifying the method for the selection of the hyperrectangles. The paper also addresses the use of integer variables.

Recently, two DIRECT-type algorithms with interesting ideas about the partition and selection of hyperrectangles appear for the scientific community [10,35]. The authors in [10] propose a DIRECT-type algorithm for BCGO problems and a DIRECT-type bilevel approach for CGO. Their convergence properties are established. In theory, the lower level feasibility problem, by minimizing a penalty function, is solved and then at the upper level, the minimum

of f is obtained among all the solutions of the lower level problem. At each iteration, the algorithm makes use of a derivative-free algorithm for non-smooth constrained local optimization (DFN$_{con}$ in [14]) starting from the center points of all identified POH for f which are a subset of the POH for θ (the penalty function of the general constraints), with the aim of improving f_{min}. Since DIRECT is good at locating promising regions of the search region Ω, although its efficiency deteriorates as the dimension and the ill-conditioning of the objective function increase [35], the authors in [35] 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.

2.2 The filter methodology

Given an approximation to the solution of the CGO problem (1), $x^{(k)}$, at iteration k, equality and inequality constraint violation is measured by the non-negative function $\theta : \mathbb{R}^n \to \mathbb{R}_+$,

$$\theta(x^{(k)}) = \|h(x^{(k)})\|_1 + \|g(x^{(k)})_+\|_1 \tag{5}$$

where $g_+ \in \mathbb{R}^p$ is defined componentwise by $\max\{0, g_i\}$. The filter paradigm for solving a constrained optimization problem reformulates the problem (1) into the bound constrained bi-objective optimization problem

$$\min_{x \in \Omega} (\theta(x), f(x)). \tag{6}$$

The filter method is an efficient methodology that builds a region of prohibited points, known as dominated points, while minimizing the constraint violation, θ, and the objective function, f. The concept of dominance arises from the multi-objective optimization area. According to [19], the used definition is the following.

Definition 3 A point x, or the corresponding pair $(\theta(x), f(x))$, is said to dominate y, or the corresponding pair $(\theta(y), f(y))$, denoted by $x \prec y$, if and only if

$$\theta(x) \leq \theta(y) \text{ and } f(x) \leq f(y),$$

with at least one inequality being strict.

A filter $\mathscr{F}$ is a finite set of pairs $(\theta(x), f(x))$, none of which is dominated by any of the others, and the corresponding points x are said to be non-dominated points.

   The main ideas addressed by a filter methodology are the following.

 (i) The filter is initialized by $\mathscr{F} = \{(\theta^U, -\infty)\}$ where θ^U is an upper bound on the acceptable constraint violation.
 (ii) Slightly stronger convergence properties are obtained [6] when an envelope is added around the filter, so that a point $x^{(k)}$, or the corresponding pair $(\theta(x^{(k)}), f(x^{(k)}))$, at iteration k, is acceptable to the filter if, for all $x^{(l)}$ such that $(\theta(x^{(l)}), f(x^{(l)})) \in \mathscr{F}^{(k)}$,

$$\theta(x^{(k)}) \leq \beta\theta(x^{(l)}) \text{ or } f(x^{(k)}) \leq f(x^{(l)}) - \gamma\theta(x^{(k)}) \tag{7}$$

   where $\beta, \gamma > 0$ are constants.
 (iii) When a point is added to the filter, all the dominated points are removed from the filter.
 (iv) When the constraint violation becomes sufficiently small, only a sufficient reduction is enforced on f,

$$f(x^{(k)}) \leq f(x^{(k-1)}) - \alpha\Delta^{(k)}$$

   where $\alpha \in (0, 1)$ and $\Delta^{(k)} > 0$ tends to zero as k increases.

As described in the next section, our algorithm implements some specific ideas from the filter methodology.

2.3 Filter-based DIRECT algorithm

For solving the CGO (1), the authors in [10] define two problems. The feasibility problem aims to minimize a penalty function subject to the bound constraints and the optimality problem minimizes the objective function on the set of all the solutions to the feasibility problem. While in [10], a bilevel approach is used, where the lower level problem is the feasibility problem and the upper level is the optimality problem, our proposal minimizes constraint violation θ and objective function f simultaneously, at the same level, using a filter set methodology.

In the context of the bi-objective problem (6), to extend the concept of the "Selection" step, where the indices of the hyperrectangles which are promising are identified, in the sense that both θ and f are to be minimized, a definition of POH with respect to the function θ is required. The definition presented in [10] is adopted.

Definition 4 Given the partition $\{H^i : i \in I_k\}$ of Ω, let ϵ be a positive constant. A hyper-rectangle j is said to be potentially optimal w.r.t. the function θ if there exists some constant $\hat{K}^j > 0$ such that

$$
\begin{aligned}
\theta(c_j) - \frac{\hat{K}^j}{2}\|u^j - l^j\| &\leq \theta(c_i) - \frac{\hat{K}^j}{2}\|u^i - l^i\|, \quad \text{for all } i \in I_k \\
\theta(c_j) - \frac{\hat{K}^j}{2}\|u^j - l^j\| &\leq \theta_{\min} - \epsilon\theta_{\min}
\end{aligned}
\tag{8}
$$

where c_j is the center of the hyperrectangle j and $\theta_{\min} = \min_{i \in I_k} \theta(c_i)$.

The filter-based DIRECT algorithm for solving the CGO problem is described in Algorithm 2.2.

In the herein presented filter method, the identification of dominated and non-dominated points is crucial when they are infeasible. To identify the non-dominated points we modify the sufficient decrease conditions shown in (7) and require only improvement in either θ or f for a point to be acceptable to the filter, i.e., we use only the dominance conditions.

Furthermore, another specific property of our algorithm is related to the feasible points not being added to the filter [1]. In Algorithm 2.2, points are added to $\mathscr{F}$ as they are generated. Each time a trial point is generated and the corresponding f and θ values are evaluated, the algorithm updates the filter. Whether a point is acceptable to the filter will depend on when it is generated. This temporal property can cause the so called 'blocking entries' (in particular, of infeasible points that are very near to feasibility). If the feasible points were added to the filter, any infeasible point x with $\theta(x) < \theta(x^{(l)})$ (for all $x^{(l)}$ such that $(\theta(x^{(l)}), f(x^{(l)})) \in \mathscr{F}^{(k)}$ and $\theta(x^{(l)}) > 0$) and $f(x) > f(x^F)$, (where $(\theta(x^F), f(x^F)) \in \mathscr{F}^{(k)}$ and $\theta(x^F) = 0$), would not be acceptable to the filter. To avoid the problem of 'blocking entries', our filter contains only infeasible points. This strategy aims to make the filter less conservative. However, the feasible point with the least function value found so far is separately maintained and will only be used to filter other feasible points.

Thus, using this filter methodology, the "Selection" step of the algorithm defines three separate sets of indices in the partition of H^k:

– the set I_k^F which corresponds to hyperrectangles with feasible center points, relative to the general equality and inequality constraints, i.e., the center points that in practice satisfy $\theta(c_j) \leq \theta_{\text{feasible}}$, for a sufficiently small positive tolerance θ_{feasible};

Algorithm 2.2 Filter-based DIRECT

1. **Initialization**. Normalize the search space Ω to be an n-dimensional unit hypercube H^0. Sample the center point c_1 of this hypercube and evaluate $f(c_1)$ and $\theta(c_1)$. Initialize $f_{\min} = f(c_1)$, $\theta_{\min} = \theta(c_1)$, set ($f/\theta$ evaluation counter) $FE = 1$ and (iteration counter) $k = 0$. Set $I_0 = \{0\}$. Initialize $\mathscr{F}$. Identify the sets I_k^F, I_k^{I-ND} and I_k^{I-D}.

2. **Selection**. Given the partition $\{H^i : i \in I_k$ where $I_k = I_k^F \cup I_k^{I-ND} \cup I_k^{I-D}\}$ of Ω:

   2.1 If $I_k^F \neq \emptyset$, find $O_k^F \subseteq I_k^F$ of POH w.r.t. f.
   2.2 If $I_k^{I-ND} \neq \emptyset$, find $O_k^{I-ND} \subseteq I_k^{I-ND}$ of POH w.r.t. θ.
   2.3 If $I_k^{I-D} \neq \emptyset$, find $O_k^{I-D} \subseteq I_k^{I-D}$ of POH w.r.t. θ.

3. **Sampling**.

   3.1 For any hyperrectangle $j_1 \in O_k^F$, identify the set L_k^{j1} of dimensions with the maximum size, s_F. Let $|L_k^{j1}|$ be the cardinal of L_k^{j1}. Set $\delta_F = 1/3s_F$. Sample f and θ at points $c_{j_1} \pm \delta_F e_i$, for all $i \in L_k^{j1}$. Update $\mathscr{F}$ with the trials points when appropriate using Definition 3.

   3.2 For any hyperrectangle $j_2 \in O_k^{I-ND}$, identify the set L_k^{j2} of dimensions with the maximum size, s_{I-ND}. Let $|L_k^{j2}|$ be the cardinal of L_k^{j2}. Set $\delta_{I-ND} = 1/3s_{I-ND}$. Sample f and θ at points $c_{j_2} \pm \delta_{I-ND} e_i$, for all $i \in L_k^{j2}$. Update $\mathscr{F}$ with the trials points when appropriate using Definition 3.

   3.3 For any hyperrectangle $j_3 \in O_k^{I-D}$, identify the set L_k^{j3} of dimensions with the maximum size, s_{I-D}. Let $|L_k^{j3}|$ be the cardinal of L_k^{j3}. Set $\delta_{I-D} = 1/3s_{I-D}$. Sample f and θ at points $c_{j_3} \pm \delta_{I-D} e_i$, for all $i \in L_k^{j3}$. Update $\mathscr{F}$ with the trials points when appropriate using Definition 3.

4. **Division**.

   4.1 Divide the hyperrectangle $j_1 \in O_k^F$ containing c_{j_1} into thirds along the dimensions in L_k^{j1}, using the rules of "preference point" and "preference order" described in Definition 5. Set $O_k^F = O_k^F \setminus \{j_1\}$. Set $I_k = I_k \bigcup_{i=1,\ldots,2|L_k^{j1}|+1} \{j_i\} \setminus \{j_1\}$.

   4.2 Divide the hyperrectangle $j_2 \in O_k^{I-ND}$ containing c_{j_2} into thirds along the dimensions in L_k^{j2} using the rules of "preference point" and "preference order" described in Definition 5. Set $O_k^{I-ND} = O_k^{I-ND} \setminus \{j_2\}$. Set $I_k = I_k \bigcup_{i=1,\ldots,2|L_k^{j2}|+1} \{j_i\} \setminus \{j_2\}$.

   4.3 Divide the hyperrectangle $j_3 \in O_k^{I-D}$ containing c_{j_3} into thirds along the dimensions in L_k^{j3} using the rules of "preference point" and "preference order" described in Definition 5. Set $O_k^{I-D} = O_k^{I-D} \setminus \{j_3\}$. Set $I_k = I_k \bigcup_{i=1,\ldots,2|L_k^{j3}|+1} \{j_i\} \setminus \{j_3\}$.

   Set $FE = FE + 2(|L_k^{j1}| + |L_k^{j1}| + |L_k^{j3}|)$.

5. **Iteration**. Let $O_k = O_k^F \cup O_k^{I-ND} \cup O_k^{I-D}$. If $O_k \neq \emptyset$ go to **3**; otherwise set $k = k + 1$; update $f_{\min}$ and $\theta_{\min}$ as follows:

   5.1 Identify the sets I_k^F, I_k^{I-ND} and I_k^{I-D}.
   5.2 If $I_k^F \neq \emptyset$, set $f_{min} = f(c_m)$ and $\theta_{\min} = \theta(c_m)$ such that $m = \arg\min_{i \in I_k^F} f(c_i)$; otherwise $\theta_{\min} = \arg\min_{i \in I_k^{I-ND} \cup I_k^{I-D}} \theta(c_i)$.

6. **Termination**. If stopping criteria are met, stop; otherwise go to **2**.

- the set I_k^{I-ND} which corresponds to hyperrectangles with center points that are infeasible and non-dominated points;
- the set I_k^{I-D}, corresponding to hyperrectangles that have infeasible and dominated center points,

where $I_k = I_k^F \cup I_k^{I-ND} \cup I_k^{I-D}$ and $I_k^F \cap I_k^{I-ND} = I_k^F \cap I_k^{I-D} = I_k^{I-ND} \cap I_k^{I-D} = \emptyset$. We note that, at iteration k, at least one of these sets is non-empty. Namely, one of the following cases can occur: $I_k^F \neq \emptyset$ and $I_k^{I-ND} = I_k^{I-D} = \emptyset$; $I_k^{I-ND} \neq \emptyset$ and $I_k^F = I_k^{I-D} = \emptyset$; or $I_k^{I-ND} \neq I_k^{I-D} \neq \emptyset$ and $I_k^F = \emptyset$; or $I_k^F \neq I_k^{I-ND} \neq I_k^{I-D} \neq \emptyset$.

To identify the set of POH, the three sets are handled separately. Our strategy is the following. If there are hyperrectangles with feasible center points, i.e., $I_k^F \neq \emptyset$, the objective function values at the center points are used to define the convex hull and POH w.r.t. f are identified. On the other hand, if there are hyperrectangles with infeasible and non-dominated center points ($I_k^{I-ND} \neq \emptyset$) then the constraint violation function θ is used instead, and POH w.r.t. θ are identified. Further, if $I_k^{I-D} \neq \emptyset$, meaning that hyperrectangles with infeasible and dominated center points do exist, then the constraint violation function θ is used and POH w.r.t. θ are identified. We note that handling I_k^{I-ND} and I_k^{I-D} separately to construct the convex hull, POH with only non-dominated center points are identified when handling I_k^{I-ND}, while POH with only dominated center points are found when using I_k^{I-D}.

If this separation of hyperrectangles with infeasible center points was not made, meaning that the sets I_k^{I-ND} and I_k^{I-D} of hyperrectangles with infeasible center points are combined into one set, only one set of POH w.r.t θ is identified. At a certain iteration, POH with infeasible non-dominated and dominated center points will probably exist in the convex hull w.r.t. θ. This is illustrated in Fig. 1a, b, c and d which contain the sample points produced on iterations 7, 9, 15 and 34 (the last one) respectively, when solving Problem 5 available in [3]. The horizontal axis shows the sizes of the hyperrectangles and the vertical axis corresponds to the θ values of their centers. Infeasible non-dominated points are marked with '+' (blue) and infeasible dominated points are marked with '×' (red). The selected POH of the convex hull are identified by a black circle. POH with non-dominated center points are the ones with smallest sizes mostly.

Since the separation of the sets I_k^{I-ND} and I_k^{I-D} aimed to promote the exploration of the non-dominated infeasible region in an expeditious manner, we show in the next section that keeping these sets separate gives better results than using them combined into a single one.

When dividing the hyperrectangles in the "Division" step, the filter-based DIRECT algorithm uses the following definition of "preference point" by dimension and "preference order".

Definition 5 The rules to define "preference point" by dimension for dividing hyperrectangles are the following. A pool is created according to the properties feasibility/infeasibility and non-dominance/dominance of each trial point generated by dimension, t_i for $i = 1, 2$:

- if both trial points are feasible, the one with smallest f value is selected for the pool;
- if one trial point is feasible and the other is infeasible, the feasible point is selected for the pool;
- if both trial points are infeasible and non-dominated (or both dominated), the one with smallest θ value is selected for the pool;
- if both trial points are infeasible and only one is non-dominated, the non-dominated point is selected for the pool, whatever its θ value.

After the rules for a "preference point" by dimension being established, the "preference order" for division of the hyperrectangles is as follows (favoring dimensions with feasible "preference points" against infeasible ones). Starting with the dimensions that were selected by the feasibility property, first divide along the dimension m such that $f_m = \min f(t_i)$ and then continue dividing until the dimension with the highest $f(t_i)$. Then, using the dimensions that were selected by the infeasibility property, first divide along the dimension m such that $\theta_m = \min \theta(t_i)$ and then continue dividing until the dimension with the highest $\theta(t_i)$.
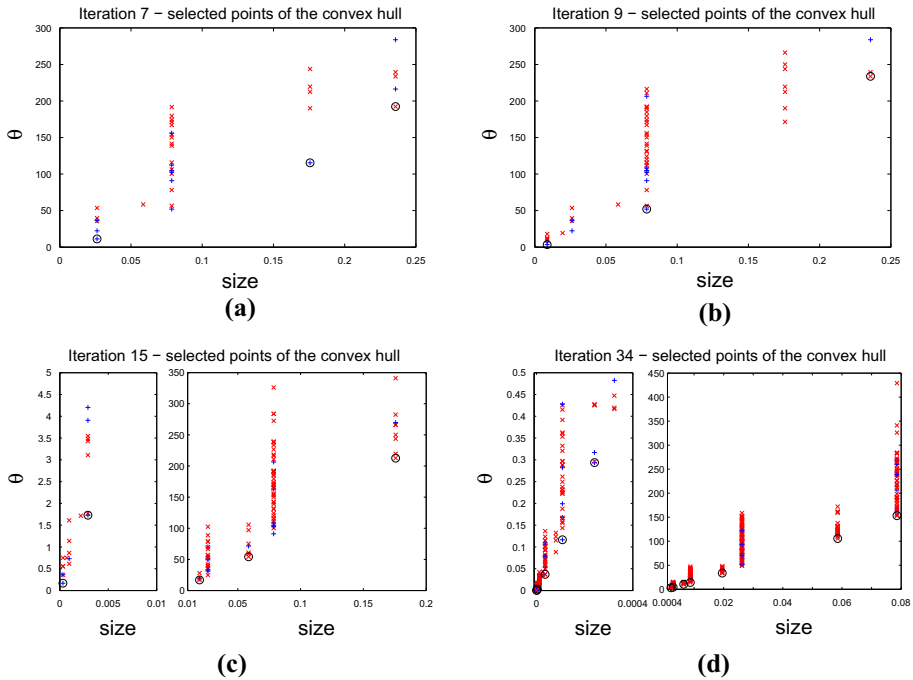
Fig. 1 Sets I_k^{I-ND} and I_k^{I-D} combined into one set -'+' (blue): non-dominated point; '×' (red): dominated point. **a** Based on $I_k^{I-ND} \cup I_k^{I-D}$, POH on iteration 7. **b** Based on $I_k^{I-ND} \cup I_k^{I-D}$, POH on iteration 9. **c** Based on $I_k^{I-ND} \cup I_k^{I-D}$, POH on iteration 15. **d** Based on $I_k^{I-ND} \cup I_k^{I-D}$, POH on iteration 34. (Color figure online)

For the purpose of evaluating the algorithm's performance on test problems, we will note the number of function evaluations when the algorithm finds a feasible point that is sufficiently close to the known globally optimal function value. In particular, the algorithm stops at iteration k when an approximate solution $x^{(k)}$ satisfying

$$\theta(x^{(k)}) \leq \eta_1 \text{ and } \frac{\left|f(x^{(k)}) - f^*\right|}{\max\{1, |f^*|\}} \leq \eta_2 \tag{9}$$

is found, for sufficiently small tolerances $\eta_1, \eta_2 > 0$, where f^* is the best known solution to the problem. However, if conditions (9) are not satisfied, the algorithm is allowed to run until a target value of number of iterations, $NI_{\max}$, (or a target value of function evaluations) is reached.

2.4 Convergence properties

Borrowing the ideas presented in [10], we show that the filter-based DIRECT algorithm, as described by Algorithm 2.2, satisfies the everywhere dense convergence property. We present the proposition for the more general case, i.e., all three types of hyperrectangles exist in the partition H^k of Ω, $k > 1$, since the others are particular cases.

Proposition 1 *Consider the filter-based DIRECT algorithm, as described by Algorithm 2.2, which selects*

$$O_k^F = \{i \in I_k^F : H^i \text{ is POH w.r.t. } f\} \tag{10a}$$

$$O_k^{I-ND} = \{i \in I_k^{I-ND} : H^i \text{ is POH w.r.t. } \theta\} \tag{10b}$$

$$O_k^{I-D} = \{i \in I_k^{I-D} : H^i \text{ is POH w.r.t. } \theta\} \tag{10c}$$

where $I_k = I_k^F \cup I_k^{I-ND} \cup I_k^{I-D}$ and $I_k^F \cap I_k^{I-ND} = I_k^F \cap I_k^{I-D} = I_k^{I-ND} \cap I_k^{I-D} = \emptyset$, then

(i) *All the sequences of sets/hyperrectangles $\{H^{i_k}\}$ produced by the algorithm are strictly nested, namely for every $\{H^{i_k}\}$ there exists a point $\bar{x} \in \Omega$ such that*

$$\bigcap_{k=0}^{\infty} H^{i_k} = \{\bar{x}\}.$$

(ii) *For every $\bar{x} \in \Omega$, the algorithm produces a strictly nested sequence of sets/hyperrectangles $\{H^{i_k}\}$ such that*

$$\bigcap_{k=0}^{\infty} H^{i_k} = \{\bar{x}\}.$$

Proof Let $s^i = \|u^i - l^i\|$ be the diagonal length of the hyperrectangle H^i for all $i \in I_k$ and let

$$s_F^{\max} = \max_{i \in I_k^F} s^i, \quad s_{I-ND}^{\max} = \max_{i \in I_k^{I-ND}} s^i \text{ and } s_{I-D}^{\max} = \max_{i \in I_k^{I-D}} s^i$$

be the three largest diagonal lengths in the three sets of the partition. Denote by $I_{\max}^F = \{i \in I_k^F : s^i = s_F^{\max}\}$, $I_{\max}^{I-ND} = \{i \in I_k^{I-ND} : s^i = s_{I-ND}^{\max}\}$ and $I_{\max}^{I-D} = \{i \in I_k^{I-D} : s^i = s_{I-D}^{\max}\}$ the subsets of indices of hyperrectangles with the largest diagonal lengths in the three sets of the partition. Let $O_k = O_k^F \cup O_k^{I-ND} \cup O_k^{I-D}$ and $I_{k,\max} = I_{\max}^F \cup I_{\max}^{I-ND} \cup I_{\max}^{I-D}$.

According to the Proposition 2 in [34], the convergence properties (i) and (ii) of our algorithm follow by showing that for all k, $I_{\max}^F \cap O_k^F \neq \emptyset$, $I_{\max}^{I-ND} \cap O_k^{I-ND} \neq \emptyset$ and $I_{\max}^{I-D} \cap O_k^{I-D} \neq \emptyset$.

Let the three sets of indices $\bar{I}_{\max}^F, \bar{I}_{\max}^{I-ND}, \bar{I}_{\max}^{I-D}$ be defined by

$$\bar{I}_{\max}^F = \{j \in I_{\max}^F : f(c_j) = \min_{i \in I_{\max}^F} f(c_i)\}$$

$$\bar{I}_{\max}^{I-ND} = \{j \in I_{\max}^{I-ND} : \theta(c_j) = \min_{i \in I_{\max}^{I-ND}} \theta(c_i)\}$$

$$\bar{I}_{\max}^{I-D} = \{j \in I_{\max}^{I-D} : \theta(c_j) = \min_{i \in I_{\max}^{I-D}} \theta(c_i)\}$$

For any $l \in \bar{I}_{\max}^F$, it is sufficient to choose a positive $\hat{K}_1^l$ such that

$$\hat{K}_1^l > 2 \max \left\{ \frac{f(c_l) - f_{\min} + \epsilon|f_{\min}|}{s^l}, \max_{j \in I_k^F \setminus I_{\max}^F} \frac{f(c_l) - f(c_j)}{s^l - s^j} \right\}$$

to obtain $l \in O_k^F$ (as defined in (10a)) so that $\bar{I}_{\max}^F = O_k^F \cap I_{\max}^F \neq \emptyset$.

Similarly, let $l \in \bar{I}_{\max}^{I-ND}$ be such that $\theta(c_l) \leq \theta(c_i)$ for all $i \in \bar{I}_{\max}^{I-ND}$. By choosing a positive $\hat{K}_2^l$ such that

$$\hat{K}_2^l > 2 \max \left\{ \frac{\theta(c_l) - \theta_{\min} + \epsilon\theta_{\min}}{s^l}, \max_{j \in I_k^{I-ND} \setminus I_{\max}^{I-ND}} \frac{\theta(c_l) - \theta(c_j)}{s^l - s^j} \right\}$$

to obtain $l \in O_k^{I-ND}$ (as defined in (10b)) so that $\bar{I}_{\max}^{I-ND} = O_k^{I-ND} \cap I_{\max}^{I-ND} \neq \emptyset$.

The same is true for $l \in \bar{I}_{\max}^{I-D}$ such that $\theta(c_l) \leq \theta(c_i)$ for all $i \in \bar{I}_{\max}^{I-D}$, where a positive $\hat{K}_3^l$ is chosen such that

$$\hat{K}_3^l > 2 \max \left\{ \frac{\theta(c_l) - \theta_{\min} + \epsilon\theta_{\min}}{s^l}, \max_{j \in I_k^{I-D} \setminus I_{\max}^{I-D}} \frac{\theta(c_l) - \theta(c_j)}{s^l - s^j} \right\}$$

to obtain $l \in O_k^{I-D}$ (as defined in (10c)) so that $\bar{I}_{\max}^{I-D} = O_k^{I-D} \cap I_{\max}^{I-D} \neq \emptyset$.

Therefore, for all k we have that $O_k \cap I_{k,\max} \neq \emptyset$, which concludes the proof. □

3 Numerical experiments

This section aims to present, analyze and compare the results produced by the proposed filter-based DIRECT algorithm with other DIRECT-based algorithms, implemented in the Matlab™ (Matlab is a registered trademark of the MathWorks, Inc.) programming language. Unless otherwise stated, the parameter values for the algorithm are set as follows: $\epsilon = 10^{-4}$, $\eta_1 = 10^{-4}$, $\eta_2 = 10^{-4}$ and $\theta_{\text{feasible}} = 10^{-4}$.

We illustrate the performance of our algorithm on two small problems. First, we use a two-variable problem, known as 'Gomez #3' problem (used in [26] to illustrate the performance of the therein proposed DIRECT version), with global optimum value $f^* = -0.9711$, occurring at $(0.109, -0.623)$ to show the effectiveness of Algorithm 2.2:

$$\min_{-1 \leq x_1, x_2 \leq 1} f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2) x_2^2$$

subject to $-\sin(4\pi x_1) + 2\sin^2(2\pi x_2) \leq 0$.

Our filter-based DIRECT algorithm finds a solution within 1% of the known optimum after 219 function evaluations (9 iterations) and a solution within 0.01% after 733 function evaluations (18 iterations). In [26], it is mentioned that the function evaluations required in both runs are 89 and 513 respectively. The center points sampled by our algorithm in both runs are shown in Figs. 2a and 3a respectively. The points marked with '×' (red) are infeasible and the points marked with '+' (blue) are feasible. The global minimum is displayed by a black circle. We show in Figs. 2b and 3b the behavior of $f_{\min}$ (the function value of the best solution to the problem found so far) as a function of function evaluations. The constraint violation value is zero along the iterative process.

We now take Problem 8 available in [3] and show in the Fig. 4a all the sampled center points of hyperrectangles generated by our algorithm when the sets I_k^{I-ND} and I_k^{I-D} are analyzed separately. Figure 4b shows the pairs (θ, f) corresponding to the center points of the hyperrectangles: dominated points are marked with '×' (red); non-dominated points (or filter points) are marked with '+' (blue). Based on the stopping conditions shown in (9), the algorithm converges in 23 iterations and 881 function evaluations to the global solution -118.700976 (global minimum displayed by a black circle). On the other hand, Fig. 5a, b
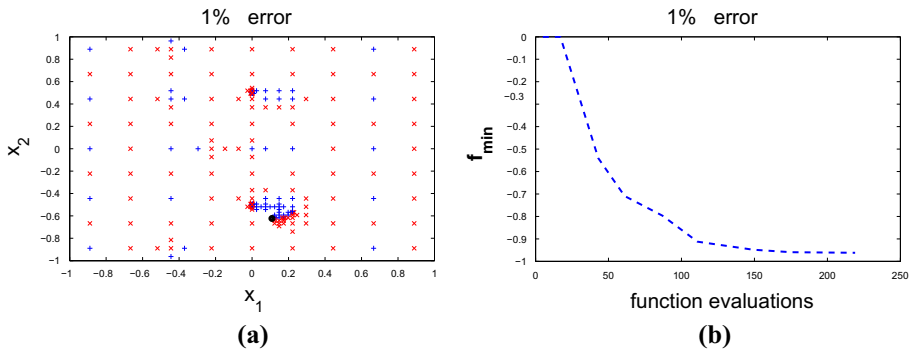
Fig. 2 Run stopping with a solution within 1% of f^*. **a** '+' (blue): feasible point; '×' (red): infeasible point. **b** Behavior of $f_{\min}$ as a function of function evaluations. (Color figure online)
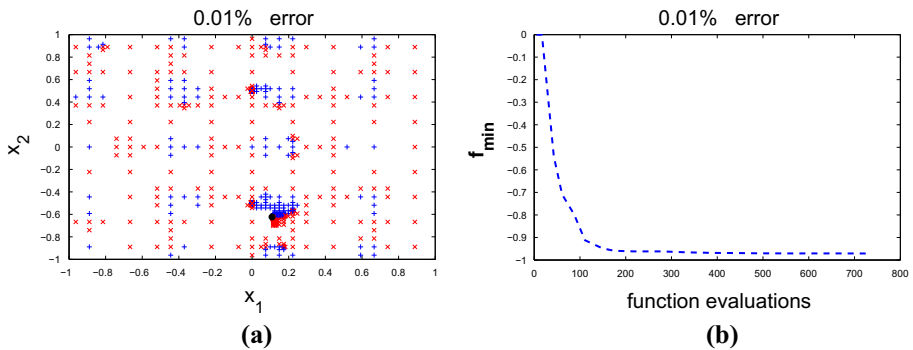
Fig. 3 Run stopping with a solution within 0.01% of f^*. **a** '+' (blue): feasible point; '×' (red): infeasible point. **b** Behavior of $f_{\min}$ as a function of function evaluations. (Color figure online)

show the center points of the hyperrectangles and the corresponding pairs (θ, f) of dominated and non-dominated points, when the sets I_k^{I-ND} and I_k^{I-D} are combined into a single set. The global solution -118.688470 is reached after 67 iterations and 3317 function evaluations. As it is illustrated in the figures, one advantage of using these separate infeasible sets is the gain in computational effort when exploring the infeasible region.

In the remaining numerical experiments, two sets of CGO problems are used. The first set contains 20 problems, with n ranging from 2 to 6 and $m + p$ ranging from 1 to 12 (details in [3]). The second set contains 82 problems, with $n \leq 10$, belonging to the GLOBALLib collection of the COCONUT project [41]. They are the same problems used for the results in Tables 1 and 2 of the report [9] (with the exception of problem ex736 due to a misprint in the definition).

The results shown in Table 1—relative to the first set of problems—aim to show the effectiveness of Algorithm 2.2 (the sets I_k^{I-ND} and I_k^{I-D} are analyzed separately) when compared with the variant where the sets are combined into a single set. This experiment analyzes the quality of the solutions produced by both algorithms when they stop with $NI_{\max} = 20$ and $NI_{\max} = 50$ iterations. In the table, $|f_{\text{glob}} - f^*|$ is used to check the proximity of the objective function value of the obtained solution, f_{glob}, to the best known optimal solution (f^* taken from [3]) and V_{glob} is the constraint violation, where $V_{\text{glob}} = \max\{\|h(x)\|_\infty, \|g(x)_+\|_\infty\}$. When $NI_{\max} = 20$, we conclude that Algorithm 2.2 produces
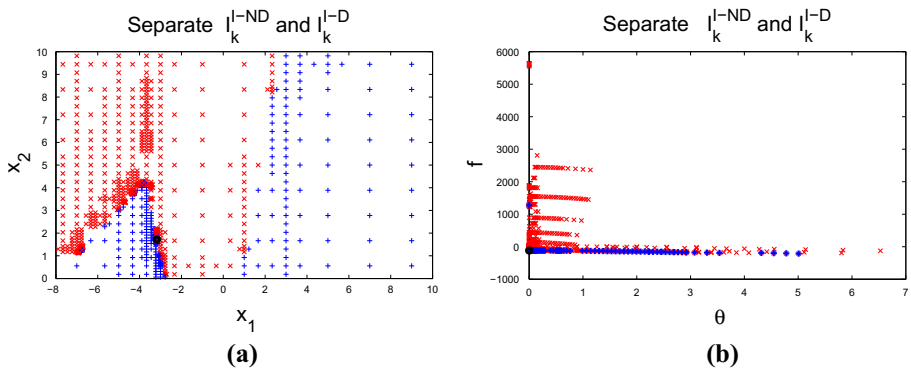
Fig. 4 The sets I_k^{I-ND} and I_k^{I-D} are analyzed separately. **a** '+' (blue): feasible point; '×' (red): infeasible point. **b** '+' (blue): non-dominated point; '×' (red): dominated point. (Color figure online)
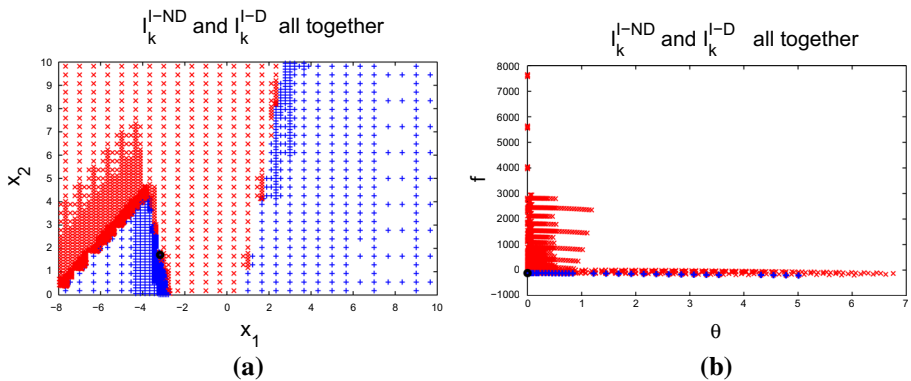
Fig. 5 I_k^{I-ND} and I_k^{I-D} are combined into a single set. **a** '+' (blue): feasible point; '×' (red): infeasible point. **b** '+' (blue): non-dominated point; '×' (red): dominated point. (Color figure online)

better errors on 11 problems, larger errors on three and tie values on six problems. On the other hand, when $NI_{\max} = 50$, Algorithm 2.2 gives better errors on 11 problems, larger errors on four and tie values on five. However, as far as constraint violation is concerned, there are 10 ties and six wins for the Algorithm 2.2 when $NI_{\max} = 20$, and nine ties, one win when $NI_{\max} = 50$. Thus, the variant nearness to the optimal solution mostly is the approach that relies on the separate analysis of the sets I_k^{I-ND} and I_k^{I-D}.

Using the same set of problems, but now using the stopping conditions in (9), the results produced by Algorithm 2.2 are compared with those of [11], where an exact penalty based algorithm (EPGO) uses the DIRECT algorithm to solve the BCGO problems. In Table 2, f_{glob} and V_{glob} have the same meaning of the previous table, NF is the number of function evaluations and k is the number of DIRECT iterations required to reach the listed solution. $NI_{\max}$ is set to 120 or 200 (depending on the problem). Problems 1, 2(a)–2(d), 5, 9, 13, 14 and 16 have been reformulated by some algebraic manipulation aiming to reduce the number of variables and equality constraints. Furthermore, Problems 3(b) and 8 had their constraints normalized. These procedures are common practice when the problem is hard to solve due to having many equality constraints or different equality and inequality constraints scaling.

Table 1 Comparison based on the quality of the solutions

| Prob. | Algorithm 2.2 - separate I_k^{l-ND} and I_k^{l-D} | | | | I_k^{l-ND} and I_k^{l-D} combined into one set | | | | f^* |
| | $NI_{max} = 20$ | | $NI_{max} = 50$ | | $NI_{max} = 20$ | | $NI_{max} = 50$ | | |
	$|f_{glob} - f^*|$	V_{glob}	$|f_{glob} - f^*|$	V_{glob}	$|f_{glob} - f^*|$	V_{glob}	$|f_{glob} - f^*|$	V_{glob}	
1	7.568e+01	5.2e−05	1.214e+01	4.4e−05	7.568e+01	5.2e−05	1.425e+01	4.3e−05	0.029313
2(a)	5.409e+02	1.3e+02	4.666e+02	0.0e+00	4.282e+02	6.4e+00	4.392e+02	0.0e+00	− 400.00
2(b)	1.434e+03	0.0e+00	9.676e+02	0.0e+00	1.017e+03	6.4e+00	1.022e+03	0.0e+00	− 600.00
2(c)	1.219e+00	9.3e+01	7.704e+02	0.0e+00	7.790e+02	6.4e+00	7.711e+02	0.0e+00	− 750.00
2(d)	2.334e+01	0.0e+00	1.372e−01	0.0e+00	6.518e+01	0.0e+00	1.447e+01	0.0e+00	− 400.00
3(a)	2.221e−01	6.9e−04	1.969e−01	5.2e−05	2.221e−01	9.7e−04	2.222e−01	4.4e−05	− 0.38880
3(b)	2.838e−05	8.9e−05	3.102e−05	9.9e−05	2.879e−05	9.5e−05	2.879e−05	9.5e−05	− 0.38881
4	8.575e−05	0.0e+00	4.973e−05	0.0e+00	8.575e−05	0.0e+00	4.973e−05	0.0e+00	− 6.6666
5	2.512e−04	5.9e−03	6.819e−04	9.6e−05	6.596e−04	1.4e−02	6.819e−04	9.6e−05	201.16
6	1.114e+00	0.0e+00	3.686e−03	9.0e−05	3.047e+00	0.0e+00	1.100e−03	4.2e−05	376.29
7	3.063e−05	3.1e−05	6.248e−05	1.0e−04	1.726e−04	0.0e+00	1.049e−04	0.0e+00	− 2.8284
8	5.372e−02	0.0e+00	3.623e−03	9.6e−05	5.796e−01	0.0e+00	7.660e−02	0.0e+00	− 118.70
9	9.648e−03	0.0e+00	2.229e−04	7.5e−05	1.211e−01	0.0e+00	4.051e−02	0.0e+00	− 13.402
10	1.469e−05	9.5e−05	2.181e−05	1.0e−04	2.249e−06	0.0e+00	1.469e−05	5.5e−05	0.74178
11	0.000e+00	0.0e+00	0.000e+00	0.0e+00	0.000e+00	0.0e+00	0.000e+00	0.0e+00	− 0.50000
12	6.239e−03	4.6e−05	1.991e−03	1.0e−04	2.007e+00	6.3e−05	2.007e+00	8.6e−05	− 16.739
13	1.031e+02	1.8e−04	1.693e+01	4.3e−05	1.166e+02	4.1e−04	1.166e+02	9.1e−05	189.35
14	5.708e−03	0.0e+00	1.800e−05	5.0e−05	1.718e−02	0.0e+00	6.147e−05	8.2e−06	− 4.5142
15	0.000e+00	5.0e−05	0.000e+00	5.0e−05	0.000e+00	5.0e−05	0.000e+00	5.0e−05	0.0000
16	5.521e−06	0.0e+00	5.006e−06	0.0e+00	5.521e−06	0.0e+00	5.006e−06	0.0e+00	0.70492

Table 2 Comparison between the filter-based DIRECT (Algorithm 2.2), the EPGO in [11] and the DF-EPGO in [12]

Prob.	Algorithm 2.2				EPGO			DF-EPGO	
	f_{glob}	V_{glob}	NF	k	f_{glob}	V_{glob}	NF	f_{glob}	V_{glob}
1	0.398895	1.00e−04	25, 425	200	0.0625	2.35e−07	39, 575	0.06256	2.35e−07
2(a)	−22.44495	0.00e00	697, 169	200	−134.1127	8.43e−04	115, 107	−134.09839	2.17e−04
2(b)	53.68674	0.00e00	421, 197	200	−768.4569	5.30e−04	120, 057	−705.13184	1.30e−03
2(c)	−38.79484	0.00e00	724, 337	200	−82.9774	8.43e−04	102, 015	−82.95282	2.17e−04
2(d)	−399.96613	0.00e00	16, 715	59	−385.1704	0.00e00	229, 773	−399.76355	0.00e00
3(a)	−0.383170	6.22e−05	1109, 995	120	−0.3861	1.02e−06	48, 647	−0.38612	3.15e−06
3(b)	−0.388802	5.98e−05	347	11	−0.3888	0.00e00	3449	−0.38881	0.00e00
4	−6.666209	0.00e+00	543	15	−6.6666	0.00e00	3, 547	−6.66665	0.00e00
5	201.159343	7.83e−05	1, 009	30	201.1593	1.66e−04	14, 087	201.15915	2.49e−04
6	376.300244	0.00e+00	1323	31	0.4701	2.05e−05	1523	376.29266	0.00e00
7	−2.828227	0.00e+00	1417	16	−2.8058	0.00e00	13, 187	−2.80585	0.00e00
8	−118.700976	0.00e+00	881	23	−118.7044	0.00e00	7621	−118.70476	0.00e00
9	−13.401764	5.58e−05	2203	23	−13.4026	1.35e−04	68, 177	−13.40125	2.16e−05
10	0.741833	1.59e−05	587	15	0.7420	0.00e00	6739	0.74178	0.00e00
11	−0.500000	0.00e+00	5	1	−0.5000	0.00e00	3579	−0.50000	0.00e00
12	−16.738797	2.15e−05	6655	100	−16.7389	5.36e−06	3499	−16.73887	9.66e−06
13	195.339906	9.90e−05	10, 583	200	195.9553	9.21e−04	8085	195.94547	3.28e−04
14	−4.513963	0.00e+00	1967	30	−4.3460	9.22e−05	19, 685	−4.35233	3.56e−06
15	0.000000	4.95e−05	105	7	0.0000	4.94e−05	1645	0.00000	4.95e−05
16	0.705008	0.00e+00	151	8	0.7181	2.00e−04	22, 593	0.71809	1.11e−04
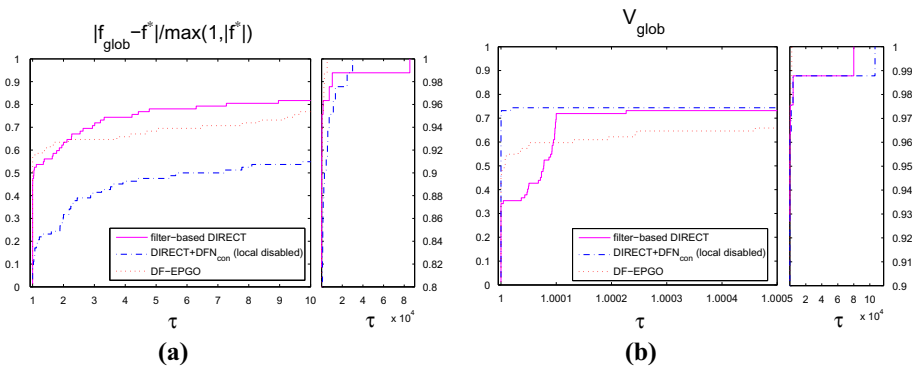
Fig. 6 Performance profiles: filter-based DIRECT algorithm, DIRECT + DFN$_{con}$ (local disabled) and DF-EPGO. **a** Metric $\dfrac{|f_{glob} - f^*|}{\max\{1, |f^*|\}}$. **b** Metric V_{glob}

The results available in [12], an exact penalty approach that relies on DIRECT to solve the BCGO problems, are also included in the comparison. For a fair comparison, the results selected for the table correspond to the version denoted by DF-EPGO, which does not include the local search phase (Table 1 in [12]). We note here that the values of NF are not provided in the cited paper. Apart from the obtained solutions to the Problems 1, 2(a)–2(c) and 6, the others are similar in the three algorithms. A simplistic analysis allows us to conclude that our algorithm produces the best f_{glob} on five problems and the second best on four problems; EPGO produces the best solution on five problems and the second best on another six; and DF-EPGO produces the best solution on nine problems and the second best on another nine. Problem 15 is a tie.

The results relative to the second set of problems are compared with those obtained by a two-level derivative-free DIRECT-type algorithm (DIRECT + DFN$_{con}$) published in [10]. The results therein published, as well as those available in the report [9], correspond to a version that is enriched with a derivative-free local search procedure (the DFN$_{con}$ proposed in [14]). We have downloaded the code in Fortran 90 (double precision), available in http://www.iasi.cnr.it/~liuzzi/[1] and disabled the local search for a more fair comparison. The 82 problems were also coded in Fortran 90. The results reported in [12] (Table 3 in the cited paper) that correspond to the algorithm DF-EPGO without the local search phase are also included in this comparison. The comparisons are made by using a graphical procedure to visualize the performance differences among the results produced by the three algorithms, in relative terms on the 82 problems, known as performance profiles [13]. Each plot reports (on the vertical axis) the percentage of problems solved with each algorithm that is within a certain threshold, τ, (on the horizontal axis) of the best result. The performance profiles of the values of the relative gap (as defined in (9)) are shown in the Fig. 6a. Figure 6b corresponds to the metric V_{glob}. The higher the percentage the better. A higher value for $\tau = 1$ means that the corresponding algorithm achieves the lowest relative gap value (or the smallest V_{glob}) mostly. Although we have imposed on our algorithm the limit on the number of hyperrectangles, $500n(m + p)$ as shown in [9], the conditions referred to in (9) are also used. It is possible to conclude that, regarding the relative gap, filter-based DIRECT

---

[1] DIRDFN—DIRECT Algorithm with derivative-free local minimizations for constrained global optimization problems (A Derivative-Free algorithm for general constrained global optimization problems - Copyright (C) 2016 G. Di Pillo, G. Liuzzi, S. Lucidi, V. Piccialli, F. Rinaldi).

produces the best results in about 40% of the problems, i.e., the probability that filter-based DIRECT gives the best result on a given problem is 0.4. Similarly, the probability that the DIRECT $+$ DFN$_{con}$ (local disabled) gives the best result on a problem is about 0.1 and the probability that the DF-EPGO gives the best result on a problem is 0.56. When the profiles for V_{glob} are analyzed, we may conclude that the probability that the filter-based DIRECT algorithm gives the best result is about 0.33, while the probability that the DIRECT $+$ DFN$_{con}$ (local disabled) gives the best result on a given problem is about 0.65. The corresponding probability for DF-EPGO is 0.42.

4 Conclusions

In this paper, a filter-based DIRECT method for solving nonsmooth and nonconvex CGO problems is proposed. With the filter methodology, we aim to minimize the constraint violation and the objective function values simultaneously. In the DIRECT context, the filter methodology gives priority to the selection of hyperrectangles with feasible center points, followed by those with infeasible and non-dominated center points and finally by those that have infeasible and dominated center points. To define the convex hull and the POH, the hyperrectangles with infeasible center points are analyzed separately depending on the non-dominance/dominance property, which in turn are addressed separately from those with feasible centers. During hyperrectangles division, special rules to define a "preference point" by dimension and a "preference order" are proposed. The convergence properties of the algorithm are analyzed. The reported results show that the filter-based DIRECT method is able to compete with other DIRECT-type methods.

A future idea to pursue is concerned with a new strategy that aims to progressively discard a portion of the infeasible region of being explored since it is unlikely that the search may return back to the feasible region from that part of the space. To improve the solution accuracy, a local search procedure will be incorporated in the algorithm.

References

1. Audet, C., Dennis Jr., J.E.: A pattern search filter method for nonlinear programming without derivatives. SIAM J. Optim. **14**(4), 980–1010 (2004)
2. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
3. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints. Technical Report MCDO121206, January 22, 2007, http://www.ime.usp.br/~egbirgin/
4. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints. Math. Program. Ser. A **125**(1), 139–162 (2010)
5. Birgin, E.G., Martínez, J.M.: Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization. Comput. Optim. Appl. **51**(3), 941–965 (2012)
6. Chin, C.M., Fletcher, R.: On the global convergence of an SLP-filter algorithm that takes EQP steps. Math. Program. **96**(1), 161–177 (2003)
7. Costa, M.F.P., Fernandes, F.P., Fernandes, E.M.G.P., Rocha, A.M.A.C.: Multiple solutions of mixed variable optimization by multistart Hooke and Jeeves filter method. Appl. Math. Sci. **8**(44), 2163–2179 (2014)

8. Dennis Jr., J.E., Price, C.J., Coope, I.D.: Direct search methods for nonlinear constrained optimization using filters and frames. Optim. Eng. **5**(2), 123–144 (2004)
9. Di Pillo, G., Liuzzi, G., Lucidi, S., Piccialli, V., Rinaldi, F.: A DIRECT-type approach for derivative-free constrained global optimization. Technical Report, July 4, 2014, http://www.math.unipd.it/~rinaldi/papers/glob_con.pdf
10. Di Pillo, G., Liuzzi, G., Lucidi, S., Piccialli, V., Rinaldi, F.: A DIRECT-type approach for derivative-free constrained global optimization. Comput. Optim. Appl. **65**(2), 361–397 (2016)
11. Di Pillo, G., Lucidi, S., Rinaldi, F.: An approach to constrained global optimization based on exact penalty functions. J. Glob. Optim. **54**(2), 251–260 (2012)
12. 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)
13. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. Ser. A **91**(2), 201–213 (2002)
14. Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A linesearch-based derivative-free approach for nonsmooth constrained optimization. SIAM J. Optim. **24**(3), 959–992 (2014)
15. Ferreira, P.S., Karas, E.W., Sachine, M., Sobral, F.N.C.: Global convergence of a derivative-free inexact restoration filter algorithm for nonlinear programming. Optimization **66**(2), 271–292 (2017)
16. Finkel, D.E.: DIRECT Optimization Algorithm User Guide. Center for Research in Scientific Computation. North Carolina State University, Raleigh (2003)
17. Finkel D.E., Kelley C.T.: Convergence Analysis of the DIRECT Algorithm. Technical Report CRSC-TR04-28, Center for Research in Scientific Computation, North Carolina State University (2004)
18. Finkel, D.E., Kelley, C.T.: Additive scaling and the DIRECT algorithm. J. Glob. Optim. **36**(4), 597–608 (2006)
19. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. Ser. A **91**(2), 239–269 (2002)
20. Gablonsky J.M.: DIRECT version 2.0 user guide. Technical Report CRSC-TR-01-08, Center for Research in Scientific Computation, North Carolina State University (2001)
21. Gablonsky, J.M., Kelley, C.T.: A locally-biased form of the DIRECT algorithm. J. Glob. Optim. **21**(1), 27–37 (2001)
22. Gould, N.I.M., Leyffer, S., Toint, PhL: A multidimensional filter algorithm for nonlinear equations and nonlinear least squares. SIAM J. Optim. **15**(1), 17–38 (2004)
23. He, J., Watson, L.T., Sosonkina M.: Algorithm 897: VTDIRECT95: serial and parallel Codes for the global optimization algorithm DIRECT. ACM Trans. Math. Softw., **36**(3), Article no. 17 (2009)
24. Hedar, A.-R., Fahim, A.: Filter-based genetic algorithm for mixed variable programming. Numer. Algebra Control Optim. **1**(1), 99–116 (2011)
25. Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization. Kluwer, Dordrecht (2000)
26. Jones, D.R.: The DIRECT global optimization algorithm. In: Floudas, C., Pardalos, P. (eds.) Encyclopedia of Optimization, pp. 431–440. Kluwer Academic Publisher, Boston (2001)
27. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**(1), 157–181 (1993)
28. Karas, E., Ribeiro, A., Sagastizábal, C., Solodov, M.: A bundle-filter method for nonsmooth convex constrained optimization. Math. Program. Ser. B **116**, 297–320 (2009)
29. Liu, M., Li, X., Wu, Q.: A filter algorithm with inexact line search. Math. Probl. Eng., Article ID 349178 20 pages (2012)
30. Liu, Q.: Linear scaling and the DIRECT algorithm. J. Glob. Optim. **56**(3), 1233–1245 (2013)
31. Liu, Q., Cheng, W.: A modified DIRECT algorithm with bilevel partition. J. Glob. Optim. **60**(3), 483–499 (2014)
32. Liu, Q., Zeng, J.: Global optimization by multilevel partition. J. Glob. Optim. **61**(1), 47–69 (2015)
33. Liu, Q., Zeng, J., Yang, G.: MrDIRECT: a multilevel robust DIRECT algorithm for global optimization problems. J. Glob. Optim. **62**(2), 205–227 (2015)
34. Liuzzi, G., Lucidi, S., Piccialli, V.: A partition-based global optimization algorithm. J. Glob. Optim. **48**(1), 113–128 (2010)
35. Liuzzi, G., Lucidi, S., Piccialli, V.: Exploiting derivative-free local searches in DIRECT-type algorithms for global optimization. Comput. Optim. Appl. **65**(2), 449–475 (2016)
36. Peng, Y., Liu, Z.: A derivative-free filter algorithm for nonlineat complementarity problem. Appl. Math. Comput. **182**(1), 846–853 (2006)
37. Price, C.J., Reale, M., Robertson, B.L.: Stochastic filter methods for generally constrained global optimization. J. Glob. Optim. **65**(3), 441–456 (2016)
38. Ribeiro, A.A., Karas, E.W., Gonzaga, C.C.: Global convergence of filter methods for nonlinear programming. SIAM J. Optim. **19**(3), 1231–1249 (2008)

39. Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P.: A filter-based artificial fish swarm algorithm for constrained global optimization: theoretical and practical issues. J. Glob. Optim. **60**(2), 239–263 (2014)
40. Sergeyev, Y.D., Kvasov, D.E.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. SIAM J. Optim. **16**(3), 910–937 (2006)
41. Shcherbina, O., Neumaier, A., Sam-Haroud, D., Vu, X.-H., Nguyen, T.-V.: Benchmarking global optimization and constraint satisfaction codes. In: Bliek, C., Jermann, C., Neumaier, A. (eds.) Global Optimization and Constraint Satisfaction, LNCS 2861, pp. 211–222. Springer, Berlin (2003)
42. Shen, C., Leyffer, S., Fletcher, R.: A nonmonotone filter method for nonlinear optimization. Comput. Optim. Appl. **52**(3), 583–607 (2012)
43. Su, K., Pu, D.: A nonmonotone filter trust region method for nonlinear constrained optimization. J. Comput. Appl. Math. **223**(1), 230–239 (2009)
44. Su, K., Lu, X., Liu, W.: An improved filter method for nonlinear complementarity problem. Math. Probl. Eng., 2013, Article ID 450829 7 pages (2013)