A DIRECT-type approach for derivative-free constrained global optimization*

G. Di Pillo[‡] G. Liuzzi[†] S. Lucidi[‡] V. Piccialli[§] F. Rinaldi[¶]

September 2, 2016

Abstract

In the field of global optimization, many efforts have been devoted to globally solving bound constrained optimization problems without using derivatives. In this paper we consider global optimization problems where both bound and general nonlinear constraints are present. To solve this problem we propose the combined use of a DIRECT-type algorithm with a derivative-free local minimization of a nonsmooth exact penalty function. In particular, we define a new DIRECT-type strategy to explore the search space by explicitly taking into account the two-fold nature of the optimization problems, i.e. the global optimization of both the objective function and of a feasibility measure. We report an extensive experimentation on hard test problems to show viability of the approach.

Keywords. Global Optimization, Derivative-Free Optimization, Nonlinear Optimization, DIRECT-type Algorithm

*This work has been partially funded by the UE (ENIAC Joint Undertaking) in the MODERN project (ENIAC-120003).

†Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche, Via dei Taurini 19 - 00185 Rome, Italy. e-mail: `giampaolo.liuzzi@iasi.cnr.it`

‡Department of Computer, Control and Management Engineering, "Sapienza" University of Rome, via Ariosto 25 - 00185 Rome, Italy. e-mails: `dipillo@dis.uniroma1.it, lucidi@dis.uniroma1.it`

§Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università degli Studi di Roma "Tor Vergata", Viale del Politecnico 1 - 00133 Rome, Italy. e-mail: `veronica.piccialli@uniroma2.it`

¶Dipartimento di Matematica, Università di Padova, Via Trieste 63 - 35121 Padua, Italy. e-mail: `rinaldi@math.unipd.it`

1 Introduction

In this paper we are interested in the *global solution* of the general nonlinear programming problem:

$$\begin{aligned} \min\ & f(x) \\ & g(x) \le 0 \\ & h(x) = 0 \\ & l \le x \le u \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^p$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $l, u \in \mathbb{R}^n$ both finite, and we assume that f, g and h are continuous functions. At first, we assume that no global information (convexity, Lipschitz constants, ...) on the problem is available. Later, we will assume that only over-estimates of the Lipshitz constants are available.
Our aim is that of finding a global minimum point x^* of problem (1). The latter task is very challenging since it involves a twofold difficulty. Indeed, we want to globally minimize the objective function while guaranteeing feasibility of the final solution. The literature on the subject is rather vast but this is not the case if we confine ourselves to methods which can be proved to converge to a global minimum point. This requirement is obviously not met by all heuristic methods.

When derivatives of the problem functions are available, the use of a merit function to manage general constraints can be envisaged. In particular, in [20] a theoretical analysis has been carried out in the framework of augmented Lagrangian methods. The use of an augmented Lagrangian merit function has been also exploited in [1] to define an efficient solution algorithm.

When derivatives of the problem functions are not available, or impractical to obtain (e.g., when problem functions are expensive to evaluate or somewhat noisy), the problem is even more difficult and challenging. Recently some attempts to solve the problem without using any derivative information have been made. In particular, we refer the interested reader to [19] for the definition of the well-known Branch-and-Reduce Optimization Navigator (BARON) which combines constraint propagation, convexification, interval analysis, and duality with advanced branch-and-bound optimization concepts. More recently, in [4] a DIvide RECTangles (DIRECT) algorithm based on exact penalty functions has been proposed.

In this paper, our aim is to combine

(i) an efficient derivative-free global optimization algorithm for problems with simple bounds,

(ii) an efficient derivative-free local optimization algorithm for problems with general constraints,

in order to develop a derivative-free algorithm for the global solution of optimization problems with general constraints.

In particular, we will make use of the well-know DIRECT algorithm [7, 10, 11] for solving global optimization problems with simple bounds, i.e. point (i) above. Further, as concerns point (ii), we shall use the recently published algorithm DFN$_{con}$ [6], which is a derivative-free algorithm for nonsmooth constrained local optimization.

The paper is organized as follows. In Section 2, we recall the basic DIRECT algorithm and report some basic theoretical properties. In Section 2.1, we present a DIRECT-type algorithm for bound constrained problems and show convergence properties both when an overestimate of the local Lipschitz constant of the objective function is not available and when it is available. In Section 3, we define a DIRECT-type bilevel approach for problems with general nonlinear constraints, along with an analysis of its convergence properties. By this approach, at the lower level we deal with the feasibility problem by minimizing a penalty function, and at the upper level we deal with objective function optimality by evaluating $f(x)$ at points with promising feasibility measure. In Section 4, we discuss how an efficient local optimization algorithm can be embedded into a DIRECT-type algorithm. In Section 5, we report the results of a numerical experimentation of the overall algorithm on a set of constrained global optimization problems from the GLOBALLib collection of COCONUT [18]. Finally, in Section 6 we draw some conclusions and discuss possible lines of future investigation.

To conclude this section, we introduce some notation used throughout the paper. We denote by $\mathcal{D}$ the hyperinterval

$$\mathcal{D} = \{x \in \mathbb{R}^n : \ l \leq x \leq u\} \tag{2}$$

and by $\mathcal{F}$ the feasible set of Problem (1), namely

$$\mathcal{F} = \{x \in \mathcal{D} : \ g(x) \leq 0, \ h(x) = 0\}.$$

We assume that $\mathcal{F} \neq \emptyset$, that is problem (1) is feasible and admits a global solution. Then, we denote by $X^* \subseteq \mathcal{F}$ the set of global solutions of Problem (1), that is

$$X^* = \{x^* \in \mathcal{F} : f(x^*) \leq f(x), \ \forall x \in \mathcal{F}\}.$$

We note that, by assumption, $X^* \neq \emptyset$. Finally, we let f^* be the global minimum value of problem (1), i.e.

$$f^* = f(x), \ \text{with } x \in X^*.$$

2 The DIRECT-type approach

In [10], the DIRECT algorithm was originally proposed to solve the following problem

$$\begin{aligned} \min\ & f(x) \\ & x \in \mathcal{D}, \end{aligned} \tag{3}$$

where $\mathcal{D}$ is given by (2). To be consistent with the notation, in this section $\mathcal{F} \equiv \mathcal{D}$ and $X^* = \{x^* \in \mathcal{D} : f(x^*) \leq f(x) \ \forall x \in \mathcal{D}\}$.

DIRECT produces finer and finer partitions of the set $\mathcal{D}$. At each iteration k, the k-th partition is described by:

$$\{\mathcal{D}^i : \ i \in I_k\}$$

where I_k is the set of indices of the hyperintervals and all the hyperintervals are given by

$$\mathcal{D}^i = \{x \in R^n : \ l^i \leq x \leq u^i\}$$

and satisfy the conditions

$$\mathcal{D} = \cup_{i \in I_k} \mathcal{D}^i, \quad \text{Int}(\mathcal{D}^i) \cap \text{Int}(\mathcal{D}^j) = \emptyset, \quad \forall\, i, j \in I_k, \quad i \neq j.$$

By x_i we denote the centroid of hyperinterval $\mathcal{D}^i$, for all $i \in I_k$.

The approach of the DIRECT algorithm can be described by the framework of a general partition-based algorithm, where, given a partition $\{\mathcal{D}^i\}$, first an *Identification Procedure* is applied in order to choose a subset to be further partitioned, then a particular *Partition Procedure* is applied to this subset.

Partition-Based Algorithm

Step 0: set $\mathcal{D}^0 = \mathcal{D}$, $I_0 = \emptyset$, $k = 0$;

Step 1: given the partition $\{\mathcal{D}^i : i \in I_k\}$ of $\mathcal{D}$

apply the *Identification Procedure* to choose a particular subset $I_k^* \subseteq I_k$;

Step 2: set $\bar{I}^0 = I_k$, $\hat{I}^0 = I_k^*$ and $p = 0$.

Step 3: choose an index $h \in \hat{I}^p$ and let $\mathcal{D}^h$ be the corresponding interval;

Step 4: apply the *Partition Procedure* to determine the partition $\{\mathcal{D}^j : j \in I_h\}$ of $\mathcal{D}^h$;

Step 5: set:

$$\bar{I}^{p+1} = \bar{I}^p \cup I_h \setminus \{h\},$$
$$\hat{I}^{p+1} = \hat{I}^p \setminus \{h\},$$

if $\hat{I}^{p+1} \neq \emptyset$ set $p = p + 1$ and go to Step 3;

Step 7: Set $I_{k+1} = \bar{I}^{p+1}$, $k = k + 1$ and go to Step 1.

An important distinguishing feature of the DIRECT Algorithm is its particular *Partition Procedure* which is described by the following scheme, where a given *Selection Procedure* selects the coordinate axis along which the partition is performed.

DIRECT-type Partition Procedure

Step 0: given the index h and the the corresponding hyperinterval $\mathcal{D}^h$, determine:

$$\begin{aligned}
&\delta = \max_{1\le j\le n}(u^h - l^h)_j, \\
&J = \left\{ j \in \{1,\dots,n\} : (u^h - l^h)_j = \delta \right\}, \\
&m = |J|;
\end{aligned}$$

Step 1: set $\tilde{\mathcal{D}}^1 = \mathcal{D}^h$, $\tilde{J}^1 = J$, $\tilde{I}^1 = \emptyset$, $p = 0$;

Step 2: apply the *Selection Procedure* to determine $\ell \in \tilde{J}^p$.

Step 3: determine the sets:

$$\begin{aligned}
&\mathcal{D}^{h_\ell} = \tilde{\mathcal{D}}^p \cap \{x \in R^n : (u^h)_\ell - \frac{\delta}{3} \le x_\ell \le (u^h)_\ell\}, \\
&\mathcal{D}^{h_\ell + m} = \tilde{\mathcal{D}}^p \cap \{x \in R^n : (l^h)_\ell \le x_\ell \le (l^h)_\ell + \frac{\delta}{3}\},
\end{aligned}$$

and set:

$$\begin{aligned}
&\tilde{\mathcal{D}}^{p+1} = \tilde{\mathcal{D}}^p \cap \{x \in R^n : (l^h)_\ell + \frac{\delta}{3} \le x_\ell \le (u^h)_\ell - \frac{\delta}{3}\}, \\
&\tilde{I}^{p+1} = \tilde{I}^p \cup \{h_\ell, h_{\ell+m}\}, \\
&\tilde{J}^{p+1} = \tilde{J}^p \setminus \{\ell\};
\end{aligned}$$

Step 4: if $\tilde{J}^{p+1} \neq \emptyset$, set $p = p + 1$ and go to Step 2;

Step 5: set:

$$\begin{aligned}
&\mathcal{D}^{h_0} = \tilde{\mathcal{D}}^{p+1}, \\
&I_h = \tilde{I}^{p+1} \cup \{h_0\};
\end{aligned}$$

and return.

The complete definition of a DIRECT-type algorithm requires the description of

- the *Identification Procedure* which determines the set I_k^* of the indices of the hyperintervals $\mathcal{D}^i$ to be further partitioned;

- the *Selection Procedure* which appears in the Partition Procedure and selects the axis along which to carry out the partition.

However, independently from the particular Identification and Selection Procedures, the previous DIRECT-type Partition Procedure guarantees some good theoretical proprieties to a partition-based algorithm.

We recall that the theoretical properties of a partition-based algorithm can be referred to the asymptotic behavior of the infinite sequence of partitions produced by the algorithm. This infinite sequence of partitions consists of an infinite number of sequences of subsets $\{\mathcal{D}^{i_k}\}$. Each of these sequences $\{\mathcal{D}^{i_k}\}$ can be defined by identifying a predecessor $\mathcal{D}^{i_{k-1}}$, with $i_{k-1} \in I_{k-1}$ to every subset $\mathcal{D}^{i_k}$, with $i_k \in I_k$, in the following way:

- if the set $\mathcal{D}^{i_k}$ has been produced at the k-th iteration, then $\mathcal{D}^{i_{k-1}}$ is the set whose partion has generated the subset $\mathcal{D}^{i_k}$ at the k-th iteration;

- if the set $\mathcal{D}^{i_k}$ has not been produced at the k-th iteration then $\mathcal{D}^{i_{k-1}} = \mathcal{D}^{i_k}$.

By definition, the sequences $\{\mathcal{D}^{i_k}\}$ are *nested* sequences of subsets, namely sequences such that, for all k,

$$\mathcal{D}^{i_k} \subseteq \mathcal{D}^{i_{k-1}}.$$

Particular nested sequences are the so called *strictly nested* sequences of subsets which have the property that for infinitely many times it results

$$\mathcal{D}^{i_k} \subset \mathcal{D}^{i_{k-1}}.$$

Returning to the properties that can be ensured by the described DIRECT-type Partition Procedure, the results reported in [13, 14, 17] allow us to state the following result.

Proposition 1 *A Partition-Based Algorithm using the DIRECT-type Partition Procedure produces at least a sequence of hyperintervals $\{\mathcal{D}^{i_k}\}$ that is strictly nested.*
Furthermore, this sequence of hyperintervals is characterized by the following properties

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\}, \qquad \textit{where} \qquad \bar{x} \in \mathcal{D};$$

$$\lim_{k\to\infty} \|u^{i_k} - l^{i_k}\| = 0.$$

The Identification Procedure of the DIRECT algorithm is based on the following definition.

Definition 1 *Given a partition $\{\mathcal{D}^i : i \in I_k\}$ of $\mathcal{D}$ and a scalar $\varepsilon > 0$, an hyperinterval $\mathcal{D}^h$ is* potentially optimal with respect to the function f *if a constant $\bar{L}^h$ exists such that:*

$$f(x^h) - \frac{\bar{L}^h}{2}\|u^h - l^h\| \le f(x^i) - \frac{\bar{L}^h}{2}\|u^i - l^i\|, \qquad \forall\ i \in I_k$$

$$f(x^h) - \frac{\bar{L}^h}{2}\|u^h - l^h\| \le f_{\min} - \epsilon|f_{\min}|$$

where

$$f_{\min} = \min_{i\in I_k} f\left(x^i\right) \tag{4}$$

At each iteration, the algorithm selects the potentially optimal hyperintervals, and considers these intervals promising and worthy of being further partitioned. This choice guarantees that the algorithm tends to generate a dense set of point over the feasible set, namely it guarantees that the algorithm has the so-called *every-where dense convergence*, which is a condition required when no global information is available, see e.g. [10].

In the following we report a proposition (see [13, 14] for the proof) which describes the everywhere dense convergence of the DIRECT Algorithm by means of the sequences of hyperintervals $\{\mathcal{D}^{i_k}\}$ generated by the algorithm.

Proposition 2 *Consider a DIRECT-type algorithm with an Identification Procedure which selects*

$$I_k^* = \big\{h \in I_k : \mathcal{D}^h \text{ is potentially optimal w.r.t. } f\big\}, \tag{5}$$

then

i) all the sequences of sets $\{\mathcal{D}^{i_k}\}$ produced by the algorithm model are strictly nested, namely for every $\{\mathcal{D}^{i_k}\}$ there exists a point $\tilde{x} \in \mathcal{D}$ such that

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

ii) for every $\tilde{x} \in \mathcal{D}$, the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ such that

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

In the sequel of the paper, we denote by *DIRECT-type algorithm* a partition-based algorithm which uses the previous DIRECT-type Partition Procedure. Therefore, any DIRECT-type Algorithm has the theoretical properties described by the previous propositions. We will show that stronger theoretical properties relevant to constrained global optimization can be stated by suitable choices of the identification procedure.

2.1 Exploiting an overestimate of the Lipschitz constant

The DIRECT Algorithm can solve a given global optimization problem without requiring any information on the value of the Lipschitz constant of the objective function. However, if an overestimate of the Lipschitz constant were available, a DIRECT-type Algorithm could exploit such an information to improve its theoretical properties.

First of all, it is possible to introduce the following definition.

Definition 2 *Given a partition $\{\mathcal{D}^i : i \in I_k\}$ of $\mathcal{D}$, a scalar $\varepsilon > 0$, a scalar $\eta > 0$ and a scalar $\bar{L} > 0$, an hyperinterval $\mathcal{D}^h$, with $i \in I_k$ is* $\bar{L}$-potentially optimal with respect to the function *f if one of the following conditions is satisfied:*

i) a constant $\tilde{L}^h \in (0, \bar{L})$ exists such that:

$$f(x^h) - \frac{\tilde{L}^h}{2}\|u^h - l^h\| \le f(x^i) - \frac{\tilde{L}^h}{2}\|u^i - l^i\|, \qquad \forall\, i \in I_k, \tag{6}$$

$$f(x^h) - \frac{\tilde{L}^h}{2}\|u^h - l^h\| \le f_{\min} - \epsilon \max\{|f_{\min}|, \eta\}, \tag{7}$$

where $f_{\min}$ is given by (4);

ii) the constant $\bar{L}$ satisfies:

$$f(x^h) - \frac{\bar{L}}{2}\|u^h - l^h\| \leq f(x^i) - \frac{\bar{L}}{2}\|u^i - l^i\|, \qquad \forall\, i \in I_k. \tag{8}$$

By using the previous definition and by assuming that the scalar $\bar{L}$ is an overstimate of the local Lipschitz constant of the objective function, it is possible to define a DIRECT-type algorithm with strong convergence properties as shown in the following result.

Proposition 3 *Consider a DIRECT-type algorithm with an Identification Procedure which selects*

$$I_k^* = \big\{h \in I_k : \mathcal{D}^h \text{ is } \bar{L}\text{-potentially optimal w.r.t. } f\big\},$$

then

i) the algorithm produces at least a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$;

ii) assume that a global minimum point x^* and an index $\bar{k}$ exist such that for all $k \geq \bar{k}$,

$$f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq f^*,$$

where $j_k \in I_k$ is the index related to the hyperinterval $\mathcal{D}^{j_k}$ such that $x^* \in \mathcal{D}^{j_k}$. Then, every strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ produced by the algorithm satisfies:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} \subseteq X^*;$$

iii) assume that for every global minimum point x^* there exist a constant $\delta > 0$ and an index $\bar{k}$ (both possibly depending on x^*) such that for all $k \geq \bar{k}$,

$$f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| < f^* - \delta\|u^{j_k} - l^{j_k}\|,$$

where $j_k \in I_k$ is the index related to the hyperinterval $\mathcal{D}^{j_k}$ such that $x^* \in \mathcal{D}^{j_k}$. Then, for every global minimum point x^*, the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ which satisfies

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{x^*\}.$$

Proof. Point i) follows from Proposition 1.

Point ii). Assume, by contradiction, that the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ such that

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

with

$$f(\bar{x}) > f^*. \tag{9}$$

Let $K \subseteq \{1, 2, \dots\}$ be the subset of iteration indices where the sets $\mathcal{D}^{i_k}$ are partitioned.

By the instructions of the algorithm, we have that, for all $k \in K$, the sets $\mathcal{D}^{i_k}$ satisfy Definition 2 and, hence, one of the two conditions of Definition 2 must hold.
If condition i) holds, a constant $\tilde{L}^{i_k} \in (0, \bar{L})$ exists such that:

$$\tilde{L}^{i_k} \geq 2\left(\frac{f(x^{i_k}) - f_{min} + \varepsilon \max\{|f_{min}|, \eta\}}{\|u^{i_k} - l^{i_k}\|}\right). \tag{10}$$

Since the sequence $\{\mathcal{D}^{i_k}\}$ is strictly nested, Proposition 1 guarantees

$$\lim_{k\to\infty} \|u^{i_k} - l^{i_k}\| = 0. \tag{11}$$

Relation (10) and limit (11) imply that,for sufficiently large values for k, we have:

$$\tilde{L}^{i_k} \geq \bar{L}.$$

Therefore an index $\tilde{k}$ exists such that for all $k \in K$ and $k \geq \tilde{k}$ condition ii) must hold. Hence, for every $k \in K$ and $k \geq \tilde{k}$, we have:

$$f(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\|,$$

which, by recalling the assumption made in point ii) of the Proposition, implies that, for all $k \in K$ and $k \geq \max\{\bar{k}, \tilde{k}\}$ we have:

$$f(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq f^*. \tag{12}$$

Since the sequence $\{\mathcal{D}^{i_k}\}$ is strictly nested, the limit (11) holds.
Now by taking the limits for $k \to \infty$ in the two terms of (12) we obtain:

$$f(\bar{x}) \leq f^*,$$

which produces a contradiction with (9).

Point iii). Again we assume, by contradiction, that there exists a global minimum point $x^* \in X^*$ for which the sequence $\{\mathcal{D}^{j_k}\}$, verifying $x^* \in \mathcal{D}^{j_k}$, for all k, is not strictly nested.
In this case, Proposition 1 implies that a scalar $\varepsilon > 0$ and an index $\bar{k}$ exist such that:

$$\|u^{j_k} - l^{j_k}\| \geq \varepsilon, \tag{13}$$

for all $k \geq \bar{k}$.
Let $\{\mathcal{D}^{i_k}\}$ be a strictly nested sequence produced by the algorithm satisfying

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

with $\bar{x} \neq x^*$.
Let $K \subseteq \{1, 2, \ldots\}$ be the subset of iteration indices where the sets $\mathcal{D}^{i_k}$ are partitioned.
Therefore, $i_k \in I_k^*$, for all $k \in K$ and, hence, for all $k \in K$, the sets $\mathcal{D}^{i_k}$ satisfy one of the two conditions of Definition 2.

By repeating the same reasoning of the proof of point ii) of the Proposition we obtain that an index $\tilde{k}$ exists such that for all $k \in K$ and $k \geq \tilde{k}$ the sets $\mathcal{D}^{i_k}$ satisfy condition ii) of Definition 2. Then, for all $k \in K$ and $k \geq \tilde{k}$, we have:

$$f(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\|.$$

By using the assumption of the point iii) of the Proposition, (13), (11) and by taking the limits for $k \to \infty$ we get the following contradiction

$$f(\bar{x}) \leq f^* - \delta\varepsilon$$

which concludes the proof. $\square$

The next proposition shows that the knowledge of the overestimate of the local Lipschitz constant of the objective function allows to define a stopping criterion for a DIRECT-type algorithm.

Proposition 4 *Consider a DIRECT-type algorithm with an Identification Procedure which selects*

$$I_k^* = \big\{h \in I_k : \mathcal{D}^h \text{ is } \bar{L}\text{-potentially optimal w.r.t. } f\big\}.$$

Assume that a global minimum point $x^* \in X^*$ and an index $\bar{k}$ exist such that for all $k \geq \bar{k}$,

$$f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq f^*,$$

where $j_k \in I_k$ is the index related to the hyperinterval $\mathcal{D}^{j_k}$ such that $x^* \in \mathcal{D}^{j_k}$.
Then, for all $k \geq \bar{k}$, the following inequality holds

$$f(x^{h_k}) - f^* \leq \frac{\bar{L}}{2}\|u^{h_k} - l^{h_k}\|,$$

where the index h_k is given by:

$$f(x^{h_k}) - \frac{\bar{L}}{2}\|u^{h_k} - l^{h_k}\| = \min_{i \in I_k}\left\{f(x^i) - \frac{\bar{L}}{2}\|u^i - l^i\|\right\}.$$

Proof. The assumption made in the proposition and the definition of the index h_k imply that, for all $k \geq \bar{k}$,

$$f^* \geq f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \geq f(x^{h_k}) - \frac{\bar{L}}{2}\|u^{h_k} - l^{h_k}\|,$$

and hence the result follows. $\square$

3 General constrained global optimization problems

In this section, we describe some DIRECT-type algorithms for dealing with constrained problems of the form (1). The main idea is that of solving these problems by means of a bilevel approach. In order to do that, we first define the following penalty function

$$w(x) = \left\|\begin{pmatrix} \max\{0, g(x)\} \\ h(x) \end{pmatrix}\right\|,$$

and the related *feasibility problem*

$$\begin{array}{ll} \min & w(x) \\ & l \le x \le u. \end{array} \tag{14}$$

More specifically, we want to solve the following *optimality problem*

$$\begin{array}{lll} \min & f(y) & \\ & y \; = \; \arg\min & w(x) \\ & & l \le x \le u. \end{array} \tag{15}$$

In theory, we should solve the feasibility problem at the lower level. Then, at the upper level, we should try to identify the optimal solutions of the constrained problem among all the solutions of the lower level problem.
Hence, a sketch of the bilevel approach is the following.

1) Solve Problem (14) to global optimality, i.e. identify set $\mathcal{F} \subseteq \mathcal{D}$

2) Find a global minimum point x^* of $f(x)$ on $\mathcal{F}$, i.e. $x^* \in X^*$

We will now describe some DIRECT-type algorithms inspired by this bilevel approach. What we want is to perform a "two-level" Identification Procedure for selecting the indices of the hyperintervals to be further partitioned. At the "lower level", we identify the indices of the hyperintevals which are more promising for the feasibility problem (14). Then, at the "upper level", we choose, among those indices selected at the lower level, those ones that are also promising with respect to the objective function and, hence, promising for problem (15).

Taking into account the feasibility problem, we define an hyperinterval potentially optimal with respect to the function w if it satisfies Definition 1 by replacing the function f with the function w.
The next proposition describes the theoretical properties of a DIRECT-type algorithm for constrained problems that considers promising all the hyperintervals which, first, are potentially optimal with respect to the penalty function w and, then, are potentially optimal also with respect to the objective function f.

Proposition 5 *Consider a DIRECT-type algorithm with an Identification Procedure which selects*

$$I_k^* = \big\{h \in I_k^w : \mathcal{D}^h \text{ is potentially optimal w.r.t. } f\big\}, \tag{16}$$

where

$$I_k^w = \big\{h \in I_k : \mathcal{D}^h \text{ is potentially optimal w.r.t. } w\big\},$$

then

i) all the sequences of sets $\{\mathcal{D}^{i_k}\}$ produced by the algorithm are strictly nested, namely for every $\{\mathcal{D}^{i_k}\}$ there exists a point $\tilde{x} \in \mathcal{D}$ such that

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

ii) for every $\tilde{x} \in \mathcal{D}$, the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ such that

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

Proof. For all $\mathcal{D}^i$, $i \in I_k$ we denote the length of its diagonal by

$$d^i = ||u^i - l^i||$$

and we define with

$$d_k^{\max} = \max_{i \in I_k} d^i$$

the largest diagonal length in the partition, and denote by $I_k^{\max}$ the index subset of hyperintervals with largest diagonal:

$$I_k^{\max} = \{i \in I_k : d^i = d_k^{\max}\}$$

The convergence properties of the algorithm follow from the results contained in [13, 14] by showing (see Proposition 2 of [13]) that, for all k,

$$I_k^{max} \cap I_k^* \neq \emptyset.$$

Let us define the following index set:

$$\hat{I}_k^{max} = \big\{h \in I_k^{max} : w(x^h) = \min_{i \in I_k^{max}} w(x^i)\big\}.$$

For any choice of $h \in \hat{I}_k^{max}$ it is sufficient to choose $\tilde{L}^h > 0$ such that:

$$\tilde{L}^h = 2 \max \left\{ \frac{w(x^h) - w_{min} + \varepsilon |w_{min}|}{d^h}, \max_{j \in I_k \setminus I_k^{max}} \frac{w(x^h) - w(x^j)}{d^h - d^j} \right\},$$

to obtain $h \in I_k^w$, where I_k^w is given by (16).

From the preceding relation and the definition of potentially optimal hyperinterval it follows that:

$$\hat{I}_k^{max} = I_k^w \cap I_k^{max}.$$

Let $\ell \in \hat{I}_k^{max}$ be such that $f(x^\ell) \leq f(x^i)$ for all $i \in \hat{I}_k^{max}$. By choosing $\bar{L}^\ell > 0$ such that:

$$\bar{L}^\ell > 2 \max \left\{ \frac{f(x^\ell) - f_{min} + \varepsilon |f_{min}|}{d^\ell}, \max_{j \in I_k^w \setminus \hat{I}_k^{max}} \frac{f(x^\ell) - f(x^j)}{d^\ell - d^j} \right\},$$

we get $\ell \in I_k^*$, so that

$$I_k^* \cap I_k^{max} \neq \emptyset,$$

which concludes the proof. □

The previous result shows that the described DIRECT-type algorithm for constrained optimization problems has the every-where dense convergence property.

Similarly to the box constrained problems, in case estimates on Lipschitz constants of the problem functions are available, they can be exploited to define a DIRECT-type algorithm with stronger theoretical properties.
To this aim, we introduce the definition of a $\bar{L}$-potentially optimal hyperinterval with respect to the function w. In this case, we exploit the knowledge of the optimal value $w^* = 0$ of the function w, being $\mathcal{F} \neq \emptyset$ by assumption.

Definition 3 *Given a partition* $\{\mathcal{D}^i : i \in I_k\}$ *of* $\mathcal{D}$ *, a scalar* $\varepsilon > 0$*, a scalar* $\eta > 0$ *and a scalar* $\bar{L} > 0$ *, an hyperinterval* $\mathcal{D}^h$*, with* $i \in I_k$ *is* $\bar{L}$-potentially optimal with respect to the function w *if one of the following conditions is satisfied:*

i) a constant $\tilde{L}^h \in (0, \bar{L})$ *exists such that:*

$$
\begin{aligned}
&w(x^h) - \frac{\tilde{L}^h}{2}\|u^h - l^h\| \leq w(x^i) - \frac{\tilde{L}^h}{2}\|u^i - l^i\|, && \forall\ i \in I_k, \\
&w(x^h) - \frac{\tilde{L}^h}{2}\|u^h - l^h\| \leq w_{\min} - \epsilon \max\{|w_{\min}|, \eta\},
\end{aligned}
\tag{17}
$$

where $w_{\min}$ *is given by (4) (by replacing* f *with* w*);*

ii) the constant $\bar{L}$ *satisfies:*

$$
w(x^h) - \frac{\bar{L}}{2}\|u^h - l^h\| \leq \max\left\{0, w(x^i) - \frac{\bar{L}}{2}\|u^i - l^i\|\right\}, \qquad \forall\ i \in I_k.
$$

The next proposition describes the properties of an algorithm using overestimates of Lipschitz constants of both the penalty function w and the objective function f.

Proposition 6 *Consider a DIRECT-type algorithm with an Identification Procedure which chooses*

$$
I_k^* = \left\{h \in \bar{I}_k^w : \mathcal{D}^h \text{ is } \bar{L}\text{-potentially optimal w.r.t. } f\right\}, \tag{18}
$$

where

$$
\bar{I}_k^w = \left\{h \in I_k : \mathcal{D}^h \text{ is } \bar{L}\text{-potentially optimal w.r.t. } w\right\}, \tag{19}
$$

then

i) the algorithm produces at least a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$*;*

ii) assume that a global minimum point $x^* \in X^*$ *and an index* $\bar{k}$ *exist such that for all* $k \geq \bar{k}$*,*

$$
w(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq 0, \tag{20}
$$

$$
f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq f^*, \tag{21}
$$

where $\mathcal{D}^{j_k}$*, with* $j_k \in I^k$ *is the hyperinterval such that* $x^* \in \mathcal{D}^{j_k}$*. Then, every strictly nested sequence of sets* $\{\mathcal{D}^{i_k}\}$ *produced by the algorithm satisfies:*

$$
\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} \subseteq X^*;
$$

iii) assume that for every global minimum point $x^* \in X^*$ *there exist a constant* $\delta_f \geq 0$ *and an index* $\bar{k}$ *such that for all* $k \geq \bar{k}$*,*

$$
w(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq 0, \tag{22}
$$

$$
f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| < f^* - \delta_f\|u^{j_k} - l^{j_k}\|, \tag{23}
$$

where $\mathcal{D}^{j_k}$, with $j_k \in I^k$ is the hyperinterval such that $x^* \in \mathcal{D}^{j_k}$. Then, for every global minimum point $x^* \in X^*$, the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ which satisfies

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{x^*\}.$$

Proof. The proof of the Proposition uses the same arguments given in the proof of Proposition 3.

Again, Proposition 1 guarantees Point (i).

Point ii). First of all we note that (20) guarantees that for all $k \geq \bar{k}$ we have

$$j_k \in \bar{I}_k^w \tag{24}$$

where $x^* \in \mathcal{D}^{j_k}$.

Then, by contradiction, we assume that that the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ such that

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

with

$$\bar{x} \notin \mathcal{F} \qquad \text{or} \qquad f(\bar{x}) > f^*. \tag{25}$$

Let $K \subseteq \{1, 2, \dots\}$ be the subset of iteration indices where the sets $\mathcal{D}^{i_k}$ are partitioned.

By repeating the same steps of Proposition 3 we obtain that an index $\tilde{k}$ exists such that condition ii) of Definition 3 must hold for all $k \in K$ and $k \geq \tilde{k}$. Hence, for every $k \in K$ and $k \geq \max\{\bar{k}, \tilde{k}\}$, we have that (24) holds and

$$w(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq w(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\|,$$
$$f(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq f(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\|,$$

which, by recalling the assumption made in point ii) of the proposition, implies that:

$$w(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq 0,$$
$$f(x^{i_k}) - \frac{\bar{L}}{2}\|u^{i_k} - l^{i_k}\| \leq f^*,$$

for all $k \in K$ and $k \geq \max\{\bar{k}, \tilde{k}\}$.

Since the sequence $\{\mathcal{D}^{i_k}\}$ is strictly nested, the limit (11) holds.

Now by taking the limits for $k \to \infty$ in the previous inequalities we obtain:

$$\bar{x} \in \mathcal{F} \qquad \text{and} \qquad f(\bar{x}) \leq f^*,$$

which produces a contradiction with (25).

Point iii). The assumption of point iii) and (22) ensure that for all global minimum point $x^* \in X^*$ there exists an index such that for all $k \geq \bar{k}$ we have

$$j_k \in \bar{I}_k^w,$$

where $x^* \in \mathcal{D}^{j_k}$.
Then, the proof of this point follows by using exactly the same arguments the proof of point iii) of Proposition 3. $\square$

The next result analyzes the case where only information related to the Lipschitz constants of the penalty function w is available.

Proposition 7 *Consider a DIRECT-type algorithm with an Identification Procedure which chooses*

$$I_k^* = \big\{h \in \bar{I}_k^w : \mathcal{D}^h \text{ is potentially optimal w.r.t. } f\big\},$$

where

$$\bar{I}_k^w = \big\{h \in I_k : \mathcal{D}^h \text{ is } \bar{L}\text{-potentially optimal w.r.t. } w\big\},$$

then

i) the algorithm produces at least a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$;

ii) assume that a feasible point $\tilde{x} \in \mathcal{F}$ and an index $\bar{k}$ exist such that for all $k \geq \bar{k}$,

$$w(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq 0,$$

where $\mathcal{D}^{j_k}$, with $j_k \in I^k$ is the hyperinterval such that $\tilde{x} \in \mathcal{D}^{j_k}$. Then, every strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ produced by the algorithm satisfies:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} \subseteq \mathcal{F};$$

iii) assume that for every feasible point $\tilde{x} \in \mathcal{F}$ there exists an index $\bar{k}$ such that for all $k \geq \bar{k}$,

$$w(x^{j_k}) - \frac{\bar{L}}{2}\|u^{j_k} - l^{j_k}\| \leq 0,$$

where $\mathcal{D}^{j_k}$, with $j_k \in I^k$ is the hyperinterval such that $\tilde{x} \in \mathcal{D}^{j_k}$. Then, for every feasible point $\tilde{x} \in \mathcal{F}$, the algorithm produces a strictly nested sequence of sets $\{\mathcal{D}^{i_k}\}$ which satisfies

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

Proof. Point i) derives from Proposition 1. Point ii) follows from the same arguments used for the prof of point ii) of Proposition 3 by taking into account that the set of global minimum points of the penalty function w is the set $\mathcal{F}$ and that the assumption implies that, for $k \geq \bar{k}$ we have $j_k \in \bar{I}_k^w$ where $\tilde{x} \in \mathcal{D}^{j_k}$.

Point iii). In order to prove this point, first we show that:

$$\lim_{k \to \infty} \tilde{d}_k^{\max} = 0 \tag{26}$$

where

$$\tilde{d}_k^{\max} = \max_{i \in \bar{I}_k^w} d^i, \qquad \text{and} \qquad d^i = ||u^i - l^i||.$$

Assume, by contradiction that (26) does not hold. Since, by definition, the sequence $\{\tilde{d}_k^{\max}\}$ is not increasing and bounded from below, we have that

$$\lim_{k\to\infty} \tilde{d}_k^{\max} = \bar{\delta} > 0.$$

Therefore, we have that

$$\tilde{d}_k^{\max} \geq \bar{\delta} > 0 \qquad \text{for all } k, \tag{27}$$

which implies that the following set of indices

$$\tilde{I}_k(\bar{\delta}) = \{i \in \bar{I}_k^w : \|u^i - l^i\| \geq \bar{\delta}\},$$

is not empty for every k. Then we have

$$\{i \in \bar{I}_k^w : d^i = \tilde{d}_k^{\max}\} \subseteq \tilde{I}_k(\bar{\delta}).$$

Let $\ell_k \in \{i \in \bar{I}_k^w : d^i = \tilde{d}_k^{\max}\}$ be such that

$$f(x^{\ell_k}) \leq f(x^j) \qquad \text{for all} \qquad j \in \{i \in \bar{I}_k^w : d^i = \tilde{d}_k^{\max}\}.$$

By choosing $\bar{L}^{\ell_k} > 0$ such that:

$$\bar{L}^{\ell_k} > 2\max\left\{\frac{f(x^{\ell_k}) - f_{min} + \varepsilon|f_{min}|}{d^{\ell_k}}, \max_{j\in\{i\in\bar{I}_k^w : d^i \neq \tilde{d}_k^{\max}\}} \frac{f(x^\ell) - f(x^j)}{d^\ell - d^j}\right\},$$

we get $\ell_k \in I_k^*$.

Hence, for all k there exists an index $\ell_k \in \tilde{I}_k(\bar{\delta}) \cap I_k^*$ such that the corresponding hyperinterval $\mathcal{D}^{\ell_k}$ is partitioned by the algorithm. The DIRECT-type Partition Procedure produces subsets $\mathcal{D}^{h_j}$, $j = 1, \dots, m$, that, recalling Proposition 2.2 of [14], satisfy:

$$\|u^{h_j} - l^{h_j}\| \leq \varepsilon\|u^{\ell_k} - l^{\ell_k}\| = \varepsilon\, \tilde{d}_k^{\max}, \qquad j = 1, \dots, m,$$

with $\varepsilon \in (0, 1)$.

The compactness of the set $\mathcal{D}$ ensures that there exists a scalar N such that

$$|\tilde{I}_k(\bar{\delta})| \leq N, \qquad \text{for every } k.$$

Hence after N iterations we have that:

$$\tilde{d}_{k+N}^{\max} \leq \varepsilon\, \tilde{d}_k^{\max}.$$

By repeating the same arguments, after pN iterations , we obtain:

$$\tilde{d}_{k+pN}^{\max} \leq \varepsilon^p\, \tilde{d}_k^{\max}, \quad \text{for} \quad p = 1, 2, \dots.$$

which, for sufficiently large values of p, contradicts (27). This proves that limit (26) holds.

Now, we assume, by contradiction, that there exists a feasible point $\tilde{x} \in \mathcal{F}$ for which the sequence $\{\mathcal{D}^{j_k}\}$, verifying $\tilde{x} \in \mathcal{D}^{j_k}$, for all k, is not strictly nested. Then, by Proposition 1, we have that there exist a scalar $\varepsilon > 0$ and an index $\hat{k}$ such that:

$$\|u^{j_k} - l^{j_k}\| \geq \varepsilon, \tag{28}$$

for all $k \geq \hat{k}$.
On the other hand, the assumption of point iii) implies that for all $k \geq \bar{k}$

$$j_k \in \bar{I}_k^w,$$

which produces a contradiction between (26) and (28) and this proves the result. $\square$

We finally report below the scheme of the k-th iteration of the DIRECT algorithm for general constrained problems.

k-th iteration of the DIRECT algorithm for general constrained problems

Input: Current partition $\{\mathcal{D}^i,\ i \in I_k\}$

Step 1: Determine $I_k^w \subseteq I_k$ related to Potentially Optimal Hyperintervals for w

Step 2: Determine $I_k^f \subseteq I_k^w$ related to Potentially Optimal Hyperintervals for f

Step 3: Execute the Partition Procedure on $\{\mathcal{D}^i,\ i \in I_k^f\}$

4 Exploiting derivative-free local searches

The numerical experience obtained by using DIRECT-type algorithms for solving real or test problems in the field of box constrained optimization has pointed out that this class of algorithms is relatively efficient in locating good approximations of the global minimum points. Unfortunately, this efficiency decreases heavily as the dimensions of the problems and the ill-conditioning of the objective functions increase. This behavior is common to most global optimization methods and a possible tool to overcome it is to combine the given global method with an efficient local algorithm. Some recent examples of such mixed strategies for DIRECT-type algorithms are described in [12, 15] for box constrained problems, and in [5] for general constrained problems. In the first paper, it is shown that the DIRECT approach can be significantly improved, in term of efficiency and robustness, by combining it with a local truncated Newton method. In the other two papers, it is shown that similar improvements can be guaranteed also by using derivative-free local minimization algorithms within a DIRECT-type approach.

In this section, we thus try to improve the efficiency of the DIRECT-type approaches described in the previous section for general constraints by making use of a suitable derivative-free local algorithm.

As in the case of box constrained optimization, the idea draws inspiration from the classical multi-start approach used in global optimization where multiple local minimizations are performed starting from points generated according to some suitable distribution over the feasible set. The main drawback of multi-start approaches is that they usually generate new starting points without taking into account the information generated in the previous iterations, thus wasting a large number of local minimizations. Then the idea that we pursue is that of replacing the random generation in a multistart approach with deterministic partitioning of a DIRECT-type strategy as proposed in [2, 15].

In practice, it is possible to use the most promising points generated by DIRECT as starting points for the local algorithm by performing a derivative-free local minimization from each centroid of the potentially optimal intervals or $\bar{L}$-potentially optimal intervals for the function f. This strategy allows to exploit the ability of the DIRECT algorithm of producing partitions which eventually locate points in promising regions (i.e. points in an *attraction region* of a global solution). Indeed, it is possible to state the following result.

Proposition 8 *Let $\{\mathcal{D}^i, i \in I_k\}$ be sequence of partitions produced by one of the DIRECT-type algorithms described in the previous section. For every global minimum point x^* of $f(x)$ on $\mathcal{F}$ and for every neighborhood $\mathcal{B}(x^*, \epsilon)$, with $\epsilon > 0$, there exists an iteration k and an index $h \in I_k$ such that $x^h \in \mathcal{B}(x^*, \epsilon)$.*

The property described by the previous proposition can be fully exploited by combining a DIRECT strategy with a local minimization technique satisfying the following assumption.

Assumption 1 *For any starting point $x_0 \in \mathcal{D}$ the local minimization algorithm produces a bounded sequence of points $\{x_k\}$ which satisfies the following conditions:*

- *every accumulation point $\bar{x}$ of the sequence $\{x_k\}$ is a stationary point of the original problem;*

- *for every global solution $x^* \in X^*$, an open set $\mathcal{L}$ exists such that if $x_0 \in \mathcal{L}$ then*

$$\lim_{k \to \infty} x_k = x^*.$$

In this way, we can take advantage of some available efficient local minimization algorithm.
These results can be used also in the field of constrained optimization whenever the original constrained problem is transformed into an unconstrained problem by means of an exact penalty function.

In this paper, we propose to use, as local search engine, the derivative-free method for constrained optimization problems described in [6], based on a box-constrained minimization of a nonsmooth exact penalty function, i.e. the algorithm named DFN$_{con}$. Under suitable assumptions, Theorem 4.1 of [5] guarantees that this particular derivative-free local algorithm satisfies Assumption 1.

In conclusion, the combination of one of the DIRECT-type algorithms described in the previous section (for which Proposition 8 holds) with the derivative-free algorithm of [6] (for which Assumption 1 holds) produces a global constrained minimization method which, after a finite number of iterations, locates a global minimum point of the original problem. We refer to such an algorithm as DIRECT+DFN$_{con}$.

Below we report the scheme of the k-th iteration of the DIRECT algorithm with local searches, where $\mathcal{A}(x)$ denotes the execution of a local search starting from point x.

k-th iteration of the DIRECT algorithm with local searches for general constrained problems (DIRECT+DFN$_{con}$)

Input: Current partition $\{\mathcal{D}^i,\ i \in I_k\}$

Step 1: Determine $I_k^w \subseteq I_k$ related to Potentially Optimal Hyperintervals for w

Step 2: Determine $I_k^f \subseteq I_k^w$ related to Potentially Optimal Hyperintervals for f

Step 3: Execute $\mathcal{A}(x^i)$, $\forall\ i \in I_k^f$ and possibly improve f_{min}

Step 4: Execute the Partition Procedure on $\{\mathcal{D}^i,\ i \in I_k^f\}$

Concerning the above scheme, we remark that the local searches are performed starting from all the centroids of the potentially optimal hyperintervals for f, namely $\mathcal{D}^i$, $i \in I_k^f$. Furthermore, we recall that points x^i, $i \in I_k^f$, i.e. the starting points of the local searches $\mathcal{A}(x^i)$, are the centroids of hyperintervals $\mathcal{D}^i$, $i \in I_k^f$.

5 Numerical experiments

This section is devoted to the numerical experimentation of our algorithm DIRECT+DFN$_{con}$ and to its comparison with some other derivative-free methods suitable for constrained global optimization, namely a multistart derivative-free simulated annealing algorithm (DFSA) [16], the well-known nondominated sorting genetic algorithm NSGA-II [3] (C version of the code), and the covariance matrix adaptation evolution strategy algorithm CMA-ES [8, 9]. To simplify notation, in Figures 1–10 our algorithm DIRECT+DFN$_{con}$ is marked as "glob".

Test problems. The numerical experimentation has been carried out by considering the AMPL versions of the constrained global optimization problems belonging to the GLOBALLib collection, which is part of the COCONUT project [18]. The entire problem set can be downloaded at the URL

`http://www.mat.univie.ac.at/~neum/glopt/coconut/Benchmark/Benchmark.html`

From the above collection, we considered all the problems with $m \geq 1$ general constraints and $n \leq 100$ variables. This amounts to a total of 135 problems excluding problem `nemhaus`, which is solved by the AMPL presolver, and problem `ex9.1.6`, which has 6 binary variables.
The selected test problems have bound constraints on the variables, that is

$$\tilde{l}_i \leq x_i \leq \tilde{u}_i, \qquad i = 1, \dots, n,$$

where $\tilde{l}_i \in \mathbb{R} \cup \{-\infty\}$, $\tilde{u}_i \in \mathbb{R} \cup \{+\infty\}$, meaning that possibly not all the variables are bounded both from above and below. Then, to be able to apply the considered algorithms, we modified the problems by defining

$$l_i = \begin{cases} -10^3 & \text{if } \tilde{l}_i = -\infty \\ \tilde{l}_i & \text{otherwise} \end{cases} \qquad u_i = \begin{cases} 10^3 & \text{if } \tilde{u}_i = +\infty \\ \tilde{u}_i & \text{otherwise} \end{cases}$$

and considering the following possibly modified problems (in place of the original ones)

$$\begin{array}{ll} \min f(x) & \\ g_j(x) \le 0, & j = 1, \dots, mi, \\ h_j(x) = 0, & j = 1, \dots, me, \\ l_i \le x_i \le u_i, & i = 1, \dots, n, \end{array} \tag{29}$$

where the numbers of inequality and equality constraints are such that $mi, me \ge 0$ with $mi + me > 0$, and $-\infty < l_i < u_i < +\infty$, $i = 1, \dots, n$. We finally remark that for the test problems in this collection, no global information was available, therefore $\bar{L}$-potentially optimal intervals could not be used.

Solvers. Here is a detailed description of the solvers used in our experiments.

- DIRECT+DFN$_{con}$ implements the DIRECT-type approach for general constrained problems described in Section 3 enriched, as described in Section 4, with the local search algorithm DFN$_{con}$ [6]. We set $\epsilon = 10^{-4}$, in DIRECT, and run each local search specifying 10^{-4} as tolerance in the stopping condition. Furthermore, we choose, in the Selection Procedure, the index $\ell \in \tilde{J}^p$ according to the following rule:

  $$w^\ell = \min_{j \in \tilde{J}^p} w^j,$$

  with $w^j = \min\{w(x^h + \frac{\delta}{3} e_j), w(x^h - \frac{\delta}{3} e_j)\}$. In figure legends, DIRECT+DFN$_{con}$ is denoted by "glob" for brevity.

- DFSA is a multistart-type algorithm controlled by means of a simulated annealing criterion. It is freely available for download at the URL

  `http://www.dis.uniroma1.it/~lucidi/DFL/`

  It is described in [16] except that the local search engine DFA is substituted by DFN$_{con}$ which is the same used within DIRECT+DFN$_{con}$ (obviously, we used the same tolerance in the stopping condition as specified above);

- NSGA-II is a well-known and efficient genetic algorithm both for single and multi-objective optimization. We employed the C version of the algorithm, which is available at the URL

  `http://www.egr.msu.edu/~kdeb/codes.shtml`

  As concerns the parameters defining NSGA-II, we adopted the same values as proposed in [3], i.e.

  - population size $p = 100$
  - probability of crossover $\pi_c = 0.9$;
  - probability of mutation $\pi_m = 1/n$;
  - distribution index for crossover $\eta_c = 20$;
  - distribution index for mutation $\eta_m = 20$.

  We acknowledge that the performances of NSGA-II could significantly change by using different values for these parameters but we did not carry out any parameter optimization.

- CMA-ES (Covariance Matrix Adaptation Evolution Strategy) is an evolutionary algorithm for difficult nonlinear non-convex black-box optimization problems, see [8, 9] for more details. We employed the C version of the algorithm, which is available at the URL

  `https://www.lri.fr/~hansen/cmaes_inmatlab.html`

  For CMA-ES, we adopted default values for all of the algorithmic parameters. Since the problems in the test set are defined in such a way that the objective function can be computed also for points that do not satisfy the nonlinear constraints, nonlinear constraints have been managed by means of a penalty function, i.e., in place of the constrained problem (29) we consider the penalized problem

  $$\begin{array}{ll} \min\, P_\varepsilon(x) & \\ l_i \le x_i \le u_i, & \quad i = 1, \dots, n, \end{array}$$

  where

  $$P_\varepsilon(x) = f(x) + \frac{1}{\varepsilon}\sum_{j=1}^{mi} \max\{0, g_j(x)\} + \frac{1}{\varepsilon}\sum_{j=1}^{me} |h_j(x)|,$$

  with $\varepsilon = 10^{-3}$. On the basis of the results, this management of the nonlinear constraints is more efficient than the method proposed in the user guide of CMA-ES.

It is worth noting that both DIRECT+DFN$_{con}$ and DFSA share the same local minimization algorithm, namely DFN$_{con}$. Hence, the main difference between them is in the generation of the starting points for the local minimizations.

DIRECT+DFN$_{con}$, DFSA and CMA-ES were run by allowing at most $1.3 \times 500 \times n(mi + me)$ function evaluations. As concerns NSGA-II, we recall that it is a population based solver and that at every generation exactly p function evaluations are performed. Hence, in order to limit the maximum number of function evaluations performed by NSGA-II to the same limit used for the other codes, we impose a limit on the maximum number of generations equal to $\lceil(1.3 \times 500 \times n(me + mi))/p\rceil$.

Results. All the runs were carried out on an Intel Core 2 Duo 3.16 GHz processor with 4GB RAM. DIRECT+DFN$_{con}$ and DFSA Algorithms were implemented in Fortran 90 (with double precision arithmetic) whereas for NSGA-II and CMA-ES we used their C implementations. We ran algorithms DFSA, NSGA-II and CMA-ES, i.e. the probabilistic ones, ten times by using pre-specified different seeds for the random number sequence generator.

Concerning the running times, we point out that the ten runs of NSGA-II took 1669.74 seconds on average to solve within the given maximum number of generations all of the 153 considered problems. DFSA did 14,792.5 local searches on average and took 770.231 seconds (on average) to solve all the problems. DIRECT+DFN$_{con}$ did 1,412,329 local searches and took 988.038 seconds to complete. CMA-ES took 2825.54 seconds (on average) to complete. It is worth noting that DIRECT+DFN$_{con}$ executed almost 100 times the local searches performed by DFSA. This can be explained by considering that in DFSA the starting points of the local searches must satisfy the simulated annealing criterion. Hence, the simulated annealing criterion indeed filters out many possible starting points. This, along with the limit on the maximum number of function evaluations, explains in our opinion the different usage of local searches between DIRECT+DFN$_{con}$ and DFSA.

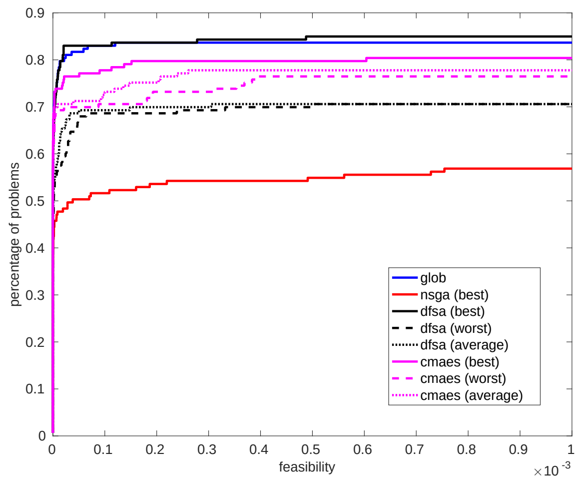

Figure 1: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right). Note that the x-axis range is from 0 to 10^{-3} and that "glob" denotes DIRECT+DFN$_{con}$

In Figures 1 and 2 we report the cumulative distribution functions for the obtained feasibility violation and optimality. More specifically, each curve $\rho(\alpha)$ gives the fraction of problems for which a solution with feasibility violation (optimality) smaller than α has been found. The optimality measure is so computed:

$$\frac{|f(x^*) - f^*|}{\max\{1,\ |f^*|\}},$$

where x^* is the best solution among those with a feasibility violation smaller that 10^{-4} and $f^\star$ is the best known value as reported in [18].

As it can be seen from Figures 1 and 2, DIRECT+DFN$_{con}$ and CMA-ES are undoubtly the best solvers for the chosen test bed. In fact, the profiles of DIRECT+DFN$_{con}$ and of all versions of CMA-ES are always the top ones (save that they are beaten by the profile of the best version of DFSA), both in terms of feasibility and optimality. It can also be noted that DIRECT+DFN$_{con}$ is the clear winner when the feasibility is taken into account, whereas CMA-ES is better in terms of robustness when it comes to optimality.

However, the fact that DIRECT+DFN$_{con}$ shows some lack of efficiency in terms of optimality can be explained by considering the well-known weakness of all DIRECT-type algorithms for problems with many variables (say more than 20). For this reason, in Figures 3 and 4, we report performance profiles of the above solvers when problems with $n \leq 20$ are taken into consideration.

As it can be noted in Figures 3 and 4, the performances of DIRECT+DFN$_{con}$ significantly change. Indeed, for problems with $n \leq 20$, i.e. moderated size problems, the proposed method, beside confirming its performances in terms of feasibility, is as efficient as CMA-ES but more robust in terms of optimality (at least when we compare with the average version of CMA-ES).

Results in the presence of noise. We further investigated the behavior and performances of DIRECT+DFN$_{con}$ and CMA-ES on problems with $n \leq 20$. Indeed, one of the main concerns when developing a solution method for derivative-free or black-box optimization is its sensitive-

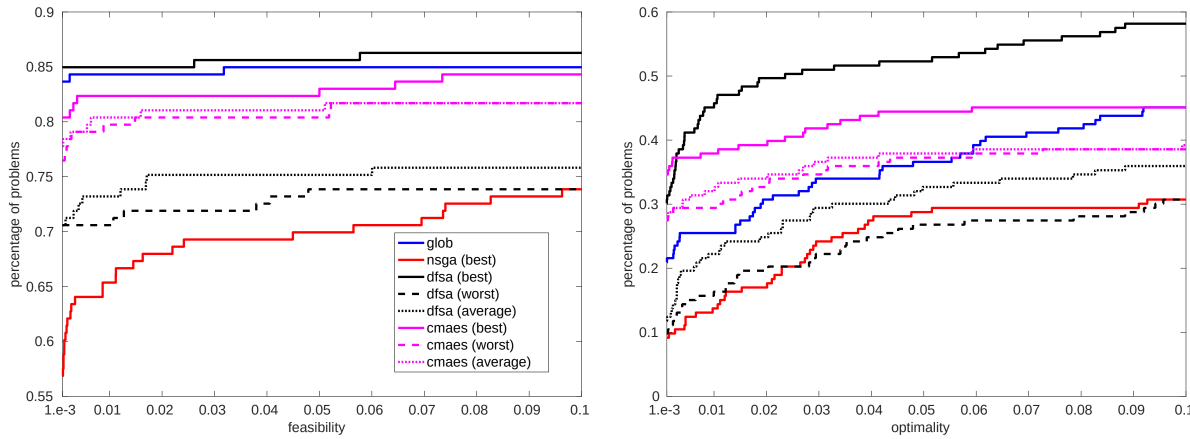

Figure 2: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right). Note that the x-axis range is from 10^{-3} to 0.1

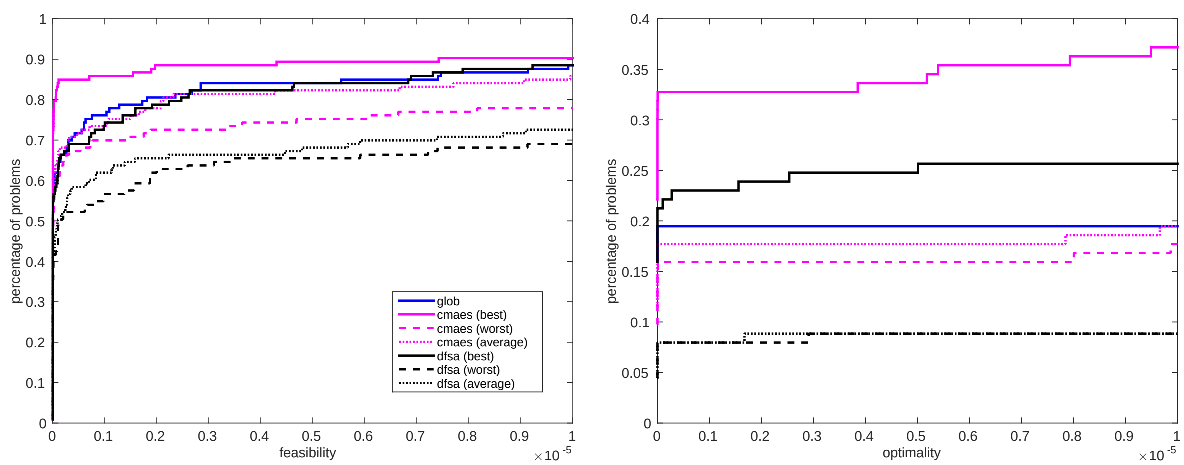

Figure 3: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right) for problems with $n \leq 20$. Note that the x-axis range is from 0 to 10^{-5}

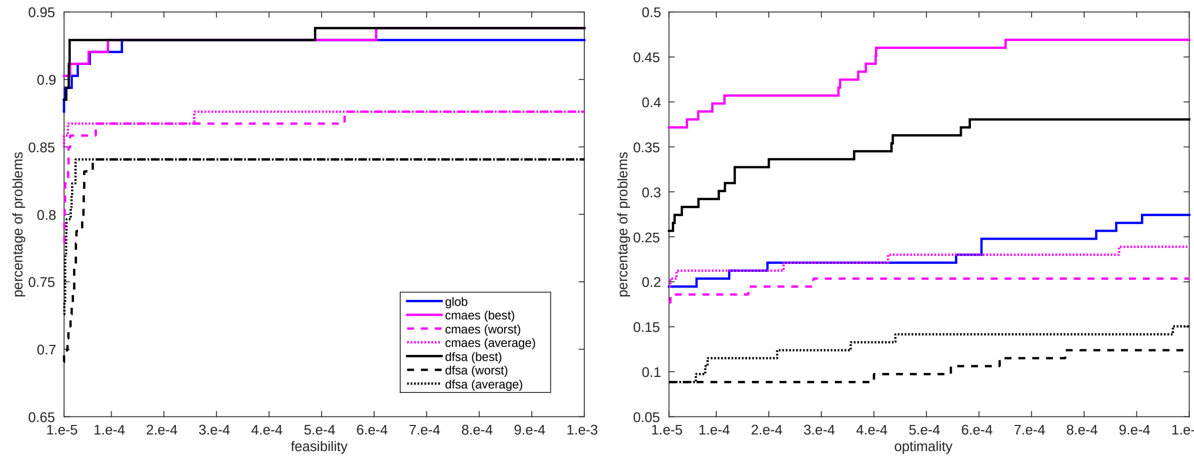

Figure 4: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right) for problems with $n \leq 20$. Note that the x-axis range is from 10^{-5} to 10^{-3}

ness to noise. Hence, we decided to carry out one more experiment where objective function values are perturbed by the presence of a certain amount of noise. We use the Gaussian noise model proposed for the Real-Parameter Black-Box Optimization annual competition which we slightly modified to take into account that the problems in our test set have a global minimum value $f^* \neq 0$. Specifically, let $f_{GN}(x, \beta)$ be defined as

$$f_{GN}(x, \beta) = (f(x) - f^*) \times e^{\beta \mathcal{N}(0,1)} + f^*,$$

where f^* is the known global minimum value, $\beta > 0$ is a parameter controlling the amount of noise and $\mathcal{N}(0, 1)$ denotes a random number drawn from a normal distribution with zero mean and standard deviation 1. Then we consider the following perturbed objective function:

$$\tilde{f}(x) = \begin{cases} f(x) & \text{if } f(x) < f^* + 10^{-8}, \\ f_{GN}(x, \beta) + 1.01 \times 10^{-8} & \text{otherwise.} \end{cases}$$

By varying the parameter β, we can control the amount of noise and, in particular, we consider two cases: 1) moderate noise level, when $\beta = 0.01$ and 2) severe noise level, when $\beta = 1$. Further, note that the constraints of the problems have not been perturbed with noise. The results of this experiment for DIRECT+DFN$_{con}$ and CMA-ES, in terms of performance profiles, are reported in Figures 5 and 6 for $\beta = 0.01$ and in Figures 7 and 8 for $\beta = 1$.
As it can be seen, when function values are noisy, no matter the amount of noise introduced, DIRECT+DFN$_{con}$ is the clear winner in terms of feasibility. As concerns the optimality, for high precision level, the average version of DIRECT+DFN$_{con}$ outperforms CMA-ES average. For moderate precision levels, we can conclude that DIRECT+DFN$_{con}$ is more robust than CMA-ES with the two codes being almost comparable in terms of efficiency.

Global capacity of DIRECT+DFN$_{con}$. We finally run a further experiment. It is more concerned with the ability of DIRECT+DFN$_{con}$ to obtain good levels of optimality and, in particular, it is aimed at understanding if and how the maximum number of function evaluations has an impact on this performance. Hence, what we did was to increase the maximum number of function

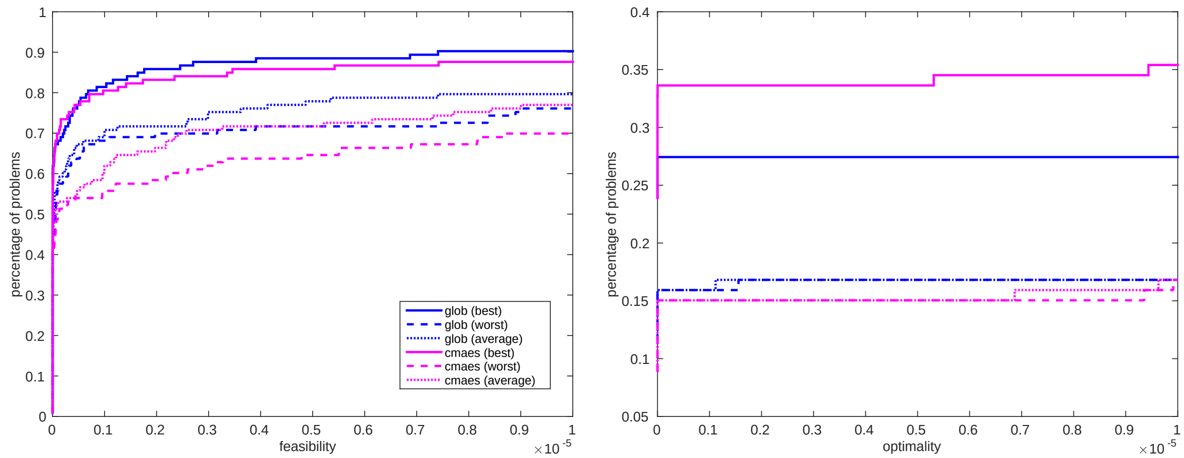

Figure 5: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right) for problems with $n \leq 20$ and when objective function are perturbed with a moderate ($\beta = 0.01$) amount of noise. Note that the x-axis range is from 0 to 10^{-5}

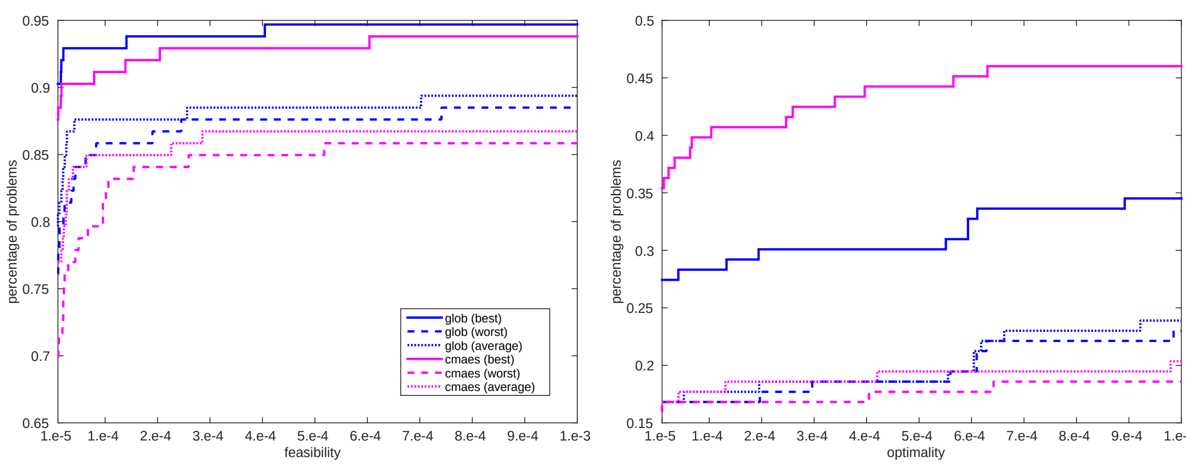

Figure 6: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right) for problems with $n \leq 20$ and when objective function are perturbed with a moderate ($\beta = 0.01$) amount of noise. Note that the x-axis range is from 10^{-5} to 10^{-3}

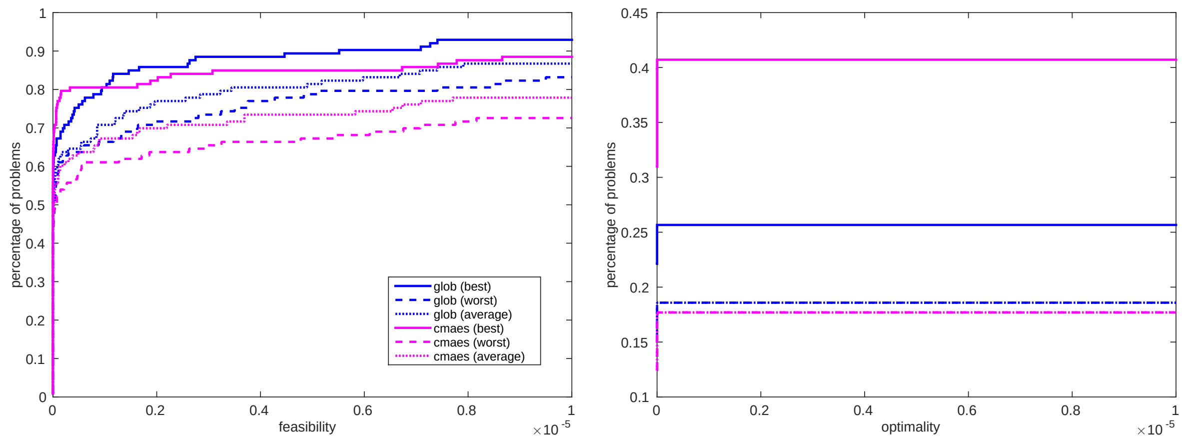

Figure 7: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right) for problems with $n \leq 20$ and when objective function are perturbed with a large ($\beta = 1$) amount of noise. Note that the x-axis range is from 0 to 10^{-5}

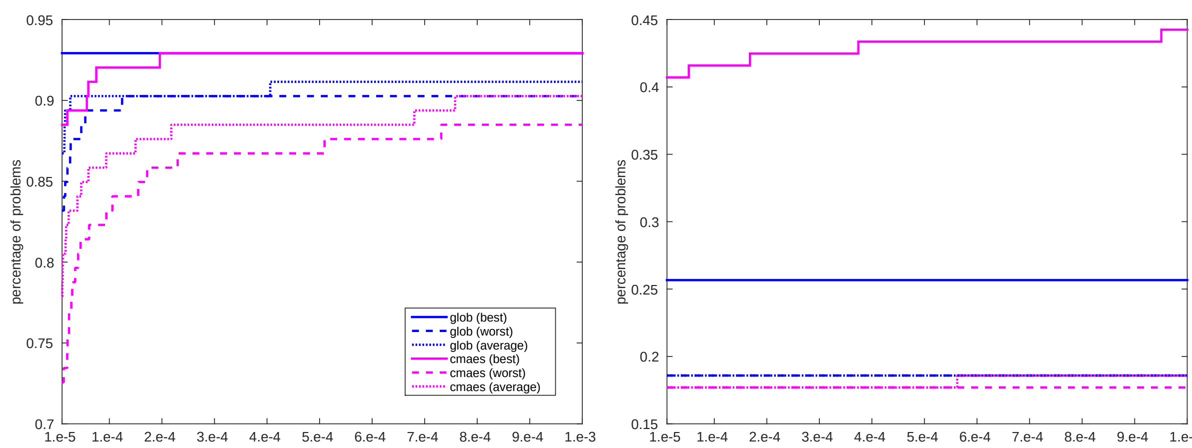

Figure 8: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right) for problems with $n \leq 20$ and when objective function are perturbed with a large ($\beta = 1$) amount of noise. Note that the x-axis range is from 10^{-5} to 10^{-3}

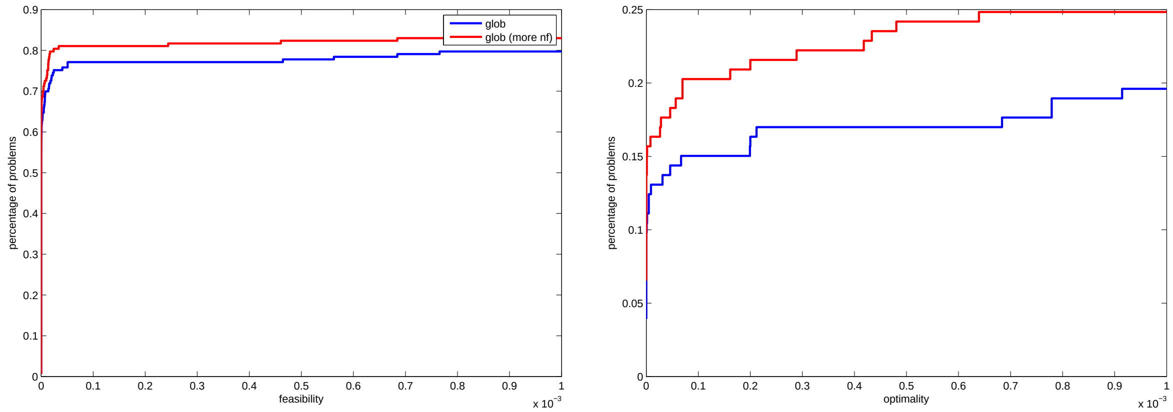

Figure 9: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right). Note that the x-axis range is from 0 to 10^{-3}

evaluations from $1.3 \times 500 \times n(mi + me)$ to $1.3 \times 5000 \times n(mi + me)$, i.e. ten times larger than the previous limit, and see what impact this has on the performance of DIRECT+DFN$_{con}$. Figures 9 and 10 reports these results by using again cumulative distribution functions for obtained feasibility violation and optimality. As we can see, and as it was largely expected, an increase of the maximum number of function evaluations immediately brings to an increased ability of the solver to have better results both in terms of feasibility violation and optimality. The total running time required by DIRECT+DFN$_{con}$ increases to 12879.5 seconds which is slightly larger than ten times the previous figure. This is reasonable given the amount of overhead computation required to manage the larger data structure necessary for DIRECT to hold information on a larger set of points.

6 Conclusions

In the paper we presented a two-level derivative-free DIRECT-type algorithm for the global solution of optimization problems with general nonlinear constraints. We carried out a complete convergence analysis of the algorithm both when no information on the local Lipschitz constants of the objective function f or of the penalty function w is available and when an overestimate of one of them is known. Further, in order to improve the numerical efficiency of the overall algorithm we embed within the two-level DIRECT-type algorithm a derivative-free local search method. The reported numerical results show the efficiency and robustness of the approach both in terms of feasibility and optimality. We also investigated the behavior of the proposed method when objective function values are perturbed by gaussian noise, showing that DIRECT+DFN$_{con}$ performances not only do not deteriorate, but they seem to improve when compared with other state-of-the-art solvers for black-box optimization.

As concerns possible aspects for future investigation and work, some lines can be envisaged. First of all, the development of more efficient versions of the DIRECT algorithm is of sure interest. This, in our opinion, can be done in at least two ways. On the one hand, by devising partition schemes less prone to dimensionality issues. This would allow to considerably increase the dimensions of problems that can be solved with our approach. On the other hand, by

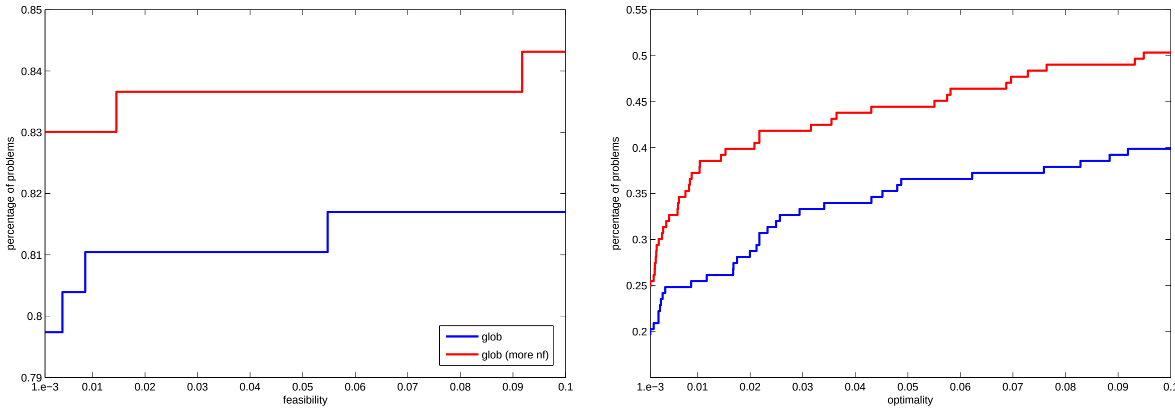

Figure 10: Comparison by using cumulative distribution functions for obtained feasibility violation (left) and optimality (right). Note that the x-axis range is from 0 to 10^{-3}

devising clever ways to take into account estimates of global information on the problem.

References

[1] E.G. Birgin, C.A. Floudas, and J.M. Martinez. Global minimization using an augmented lagrangian method with variable lower-level constraints. *Mathematical Programming (Ser. A)*, 125(1):139–162, 2010.

[2] E.F. Campana, G. Liuzzi, S. Lucidi, D. Peri, A. Pinto, and V. Piccialli. New global optimization methods for ship design problems. *Optimization and Engineering*, 10(4):533–555, 2009.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] G. Di Pillo, S. Lucidi, and F. Rinaldi. An approach to constrained global optimization based on exact penalty functions. *Journal on Global Optimization*, 54(2):251–260, 2012.

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

[6] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi. A linesearch-based derivative-free approach for nonsmooth constrained optimization. *SIAM Journal on Optimization*, 2014.

[7] J. M. Gablonsky. *DIRECT Version 2.0, User Guide.*

[8] N. Hansen, S.D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18, 2003.

[9] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[10] D.R. Jones. The direct global optimization algorithm. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, pages 431–440. Kluwer Academic Publishers, Dordrecht, 2001.

[11] 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.

[12] 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(2):353–375, 2010.

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

[14] G. Liuzzi, S. Lucidi, and V. Piccialli. Partitioning techniques for global optimization. In V. De Simone, D. Di Serafino, and G. Toraldo, editors, *Recent Advances in Nonlinear Optimization and Equilibrium Problems: A Tribute to Marco D'Apuzzo*, volume 27 of *Quaderni di Matematica, Dipartimento di Matematica, Seconda Università di Napoli*, pages 253–272. Aracne Editrice S.r.l., 2012.

[15] G. Liuzzi, S. Lucidi, and V. Piccialli. Exploiting derivative-free local searches in direct-type algorithms for global optimization. *Computational Optimization and Applications*, pages 1–27, 2015.

[16] 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.

[17] Y.D. Sergeyev. On convergence of divide the best global optimization algorithms. *Optimization*, 44:303–325, 1998.

[18] O. Shcherbina, A. Neumaier, D. Sam-Haroud, X.H. Vu, and T.V. Nguyen. Benchmarking global optimization and constraint satisfaction codes. In Ch. Bliek, Ch. Jermann, and A. Neumaier, editors, *Global Optimization and Constraint Satisfaction*, pages 211–222. Springer, Berlin, 2003.

[19] M. Tawarmalani and N.V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming.* Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[20] C.Y. Wang and D. Li. Unified theory of augmented lagrangian methods for constrained global optimization. *Journal of Global Optimization*, 44:433–458, 2009.