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

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

July 4, 2014

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:

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

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 ultimate 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 adoption of a merit function to manage general constraints can be envisaged. In particular, in [16] 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 unavailable, 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 [15] 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 [3] 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 [6, 8, 7] 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} [5], which is a derivative-free algorithm for non smooth constrained local optimization.

The paper is organized as follows. In Section 2, we recall the basic DIRECT algorithm and report a first and quite preliminary theoretical property. In Section 3, 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 known and when it is known. In Section 4, 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 promising to be feasible. In Section 5, we discuss how an efficient local optimization algorithm can be embedded into a DIRECT-type algorithm. In Section 6, 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 [14]. Finally, in Section 7 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 DIRECT-type approach

The DIRECT algorithm has been originally proposed to solve bound constrained global optimization problems, i.e. $\min f(x)$, $x \in \mathcal{D}$ by producing 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:

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

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:

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

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

Step 5: set:

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

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, 10, 11] 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

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

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

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 proposition. We will show that stronger theoretical properties relevant to constrained global optimization can be stated by suitable choices of the identification procedure.

3 A DIRECT-type approach for box constrained global optimization problems

In [7], 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 $X^* = \{x^* \in \mathcal{D} : f(x^*) \leq f(x)\ \forall x \in \mathcal{D}\}$.

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:*

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

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. [7].

In the following we report a proposition (see [10, 11] 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

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

The DIRECT Algorithm was proposed to solve 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:*

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

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

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. □

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. □

4 A DIRECT-type approach for 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*

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

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

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

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*

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

where

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

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 [10, 11] by showing (see Proposition 2 of [10]) that, for all k,

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

Let us define the following index set:

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

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:

$$w(x^h) - \frac{\tilde{L}^h}{2}\|u^h - l^h\| \leq w(x^i) - \frac{\tilde{L}^h}{2}\|u^i - l^i\|, \qquad \forall\, i \in I_k, \tag{17}$$

$$w(x^h) - \frac{\tilde{L}^h}{2}\|u^h - l^h\| \leq w_{\min} - \epsilon \max\{|w_{\min}|, \eta\},$$

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:

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

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

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

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 $k \ge \bar{k}$ 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. □

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*

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

where

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

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 $k \ge \bar{k}$,

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

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 [11], 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. □

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 $\{\mathcal{D}^i,\ i \in I_k^f\}$

5 A DIRECT-type approach with 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-conditionings 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 algorithm are described in [9, 12] for box constrained problems, and in [4] 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, 12].

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 the available efficient local minimization algorithms. These results can be used also in the field of constrained optimization whenever the original constrained problem is transformed in 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 [5], 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 [4] 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 [5] (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.

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

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 $\mathcal{A}(\bar{x}^i)\ \forall\ i \in I_k^f$ and possibly improve f_{min}

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

6 Numerical results

In this section we report the results obtained using algorithm DIRECT+DFN$_{con}$, which implements the DIRECT-type approach for general constrained problem described in Section 4 enriched, as in Section 5, with the Local Search algorithm DFN$_{con}$ [5].
The numerical experimentation has been performed on a set of 84 problems from the GLOBALLib collection of COCONUT [14] with number of variables $n \leq 10$. All test problems are of the kind

$$\begin{array}{l} \min f(x) \\ g(x) \leq 0 \\ h(x) = 0 \\ l \leq x \leq u. \end{array}$$

For the test problems in this collection, no global information was available, therefore $\bar{L}$-potentially optimal intervals could not be used.

In DIRECT, we set $\epsilon = 10^{-4}$ and the maximum number of hyperintervals equal to $500 * n * (mi + me)$, where n is the number of variables, mi is the number of inequality constraints and

me is the number of equality costraints. This maximum number of hyperintervals gives also the stopping condition for the algorithm.
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 $w^j = \min\{w(x^h + \frac{\delta}{3} e_j), w(x^h - \frac{\delta}{3} e_j)\}$.
The numerical experimentation was carried out on an Intel Core 2 Duo 3.16 GHz processor with 3.25 GB RAM. DIRECT and DFN$_{con}$ Algorithms were implemented in Fortran 90 (double precision).

The results for the problem set are collected in Tables 1 and 2, where we report the name of the problem, the number **n** of variables, the number **me** and **mi** of equality and inequality constraints (excluding simple bounds), the **CPU time** required to attain the stop condition, the optimal function value $\mathbf{f(x^\star)}$, the constraints violation **viol**, namely $cv(x^\star) = \max\{\ \|g^+(x^\star)\|_\infty,\ \|h(x^\star)\|_\infty\ \}$.
In order to synthesize the results we make reference to Figures 1 and 2, where we make also a comparison with the results obtained by using the derivative-free algorithm for constrained global optimization DF-EPGO+DFN$_{con}$ (see [4], for further details). The plot in Figure 1 gives on the y-axis the number of obtained solutions whose constraints violation is smaller or equal than the value given in the x-axis. The plot in Figure 2 gives on the y-axis the number of obtained feasible solutions (namely those solutions $x^\star$ where $cv(x^\star) \le 10^{-4}$) whose relative gap, given by

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

with $f^\star$ the best known value reported in [14], is smaller or equal than the value given in the x-axis. From these Figures, we note that our algorithm guarantees quite good performances in terms of both feasibility and optimality, and, in general, outperforms DF-EPGO+DFN$_{con}$. More specifically, by taking a look at Figure 1 we can see that almost 95% of the solutions have a constraint violation w lower than 10^{-4}. Furthermore, as we can see from Figure 2, around 70% of the problems have a gap lower than 0.1. We would also like to notice that the algorithm gives these good solutions in a reasonable time. Indeed the average CPU time for the considered problems is 10.37 seconds.

7 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 either when no information on the local Lipschitz constants of the objective function f or of the penalty function w is available or 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 of the approach.
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. This would allow to considerably increase the

PROBLEM	n	me	mi	time	val funct	viol	$f^\star$
ex542	8	6	0	9.4086	8870.5123	0.00000E+000	7512.2259
ex6212	4	0	2	0.4040	0.2927	1.00000E-007	0.2892
chance	4	2	1	0.6080	29.8950	0.00000E+000	29.8944
circle	3	10	0	4.8643	4.5868	0.00000E+000	4.5742
dispatch	4	1	1	0.5080	3155.3752	1.48367E-010	3155.2879
ex1411	3	4	0	11.5527	0.0022	0.00000E+000	0
ex1412	6	8	1	95.3060	0.4988	2.49999E-009	0
ex1413	3	4	0	8.9726	0.0010	0.00000E+000	0
ex1414	3	4	0	9.5046	0.0004	0.00000E+000	0
ex1415	6	2	4	24.2895	0.0034	8.88178E-016	0
ex1416	9	14	1	145.7211	0.0344	7.99179E-011	0
ex1417	10	16	1	73.7966	1.7815	0.00000E+000	0
ex1418	3	4	0	10.2606	0.0048	0.00000E+000	0
ex1419	2	2	0	2.9002	0.0004	0.00000E+000	0
ex1421	5	6	1	12.6608	0.0237	1.38950E-009	0
ex1422	4	4	1	2.9882	0.0636	1.38950E-009	0
ex1423	6	8	1	24.3215	0.0590	4.12566E-010	0
ex1424	5	6	1	22.7894	0.0760	1.38950E-009	0
ex1425	4	4	1	3.5442	0.0895	1.38950E-009	0
ex1426	5	6	1	26.0456	0.0243	1.38950E-009	0
ex1427	6	8	1	42.0666	0.1538	4.12566E-010	0
ex1428	4	4	1	6.0204	0.0651	1.38950E-009	0
ex1429	4	4	1	7.1084	0.0635	1.38950E-009	0
ex211	5	1	0	0.1080	-17.0000	0.00000E+000	-17
ex212	6	2	0	0.3400	-24.3116	0.00000E+000	-213
ex214	6	4	0	1.0041	-10.9937	0.00000E+000	-11
ex215	10	11	0	10.0526	-267.5463	0.00000E+000	-268.0146
ex216	10	5	0	2.6082	-39.0000	0.00000E+000	-39
ex219	10	0	1	0.3320	-0.2727	2.21033E-005	-0.375
ex311	8	6	0	5.6324	8173.7786	0.00000E+000	7049.2083
ex312	5	6	0	2.7442	-30665.5761	4.43936E-005	-30665.54
ex313	6	6	0	2.1001	-9488.6046	0.00000E+000	-310
ex314	3	3	0	0.3280	-4.0000	0.00000E+000	-4
ex418	2	0	1	0.0600	-16.7386	1.06558E-004	-16.7389
ex419	2	2	0	0.1840	-5.4961	0.00000E+000	-5.508
ex522case1	9	2	4	6.4484	-136.6245	0.00000E+000	-400
ex522case2	9	2	4	7.6885	-301.4679	0.00000E+000	-600
ex522case3	9	2	4	6.4564	-554.9412	0.00000E+000	-750
ex524	7	5	1	3.2562	-360.5533	0.00000E+000	-450
ex611	8	0	6	3.8442	-0.0202	6.91414E-008	-0.0202
ex612	4	0	3	0.5720	-0.0293	1.00000E-006	-0.0325

Table 1: Performance of the Algorithm on the GLOBALLib test set (Part I).

PROBLEM	n	me	mi	time	val funct	viol	$f^\star$
ex614	6	0	4	1.3881	-0.2918	3.69329E-007	-0.2945
ex6210	6	0	3	1.1361	-3.0507	1.00000E-007	-3.052
ex6211	3	0	1	0.1640	0.0000	2.48323E-007	0
ex6213	6	0	3	1.1721	-0.2186	1.00000E-007	-0.2162
ex6214	4	0	2	0.4480	-0.6954	1.00000E-007	-0.6954
ex625	9	0	3	10.0686	-70.3860	2.00000E-007	-70.7521
ex626	3	0	1	0.1880	0.0000	4.94405E-008	0
ex627	9	0	3	9.5206	-0.1449	3.00000E-007	-0.1608
ex628	3	0	1	0.1880	-0.0270	7.97572E-007	-0.027
ex629	4	0	2	0.4640	-0.0278	1.00000E-007	-0.0341
ex721	7	14	0	35.4782	1272.7361	0.00000E+000	1227.1896
ex722	6	1	4	1.9561	-0.3887	3.52918E-010	-0.3888
ex723	8	6	0	24.6815	7242.7603	0.00000E+000	7049.2181
ex724	8	4	0	2.7322	3.9584	0.00000E+000	3.918
ex725	5	6	0	2.9962	10694.8585	0.00000E+000	10122.4828
ex726	3	1	0	0.1480	-75.6879	0.00000E+000	-83.2499
ex727	4	2	0	0.4840	-5.7397	0.00000E+000	-5.7399
ex728	8	4	0	3.5322	-6.0645	0.00000E+000	-6.082
ex729	10	6	0	6.6044	1.9376	0.00000E+000	1.1436
ex731	4	7	0	1.3681	2.0000	0.00000E+000	0.3417
ex732	4	7	0	10.7167	1.0902	0.00000E+000	1.0899
ex733	5	6	2	3.5322	1.0599	4.40382E-008	0.8175
ex736	1	0	2	0.0240	0.0000	7.40842E-006	0
ex817	5	4	1	2.0641	0.0700	5.19382E-007	0.0293
ex818	6	1	4	1.9641	-0.3887	3.52918E-010	-0.3888
ex912	10	0	9	6.5924	-16.0001	4.06901E-005	-16
ex914	10	0	9	7.8285	-37.0011	7.18335E-005	-37
ex921	10	0	9	31.3060	16.9992	9.15527E-005	17
ex922	10	1	8	17.8211	100.0000	0.00000E+000	99.9995
ex924	8	0	7	4.1003	2.5000	4.59772E-009	0.5
ex925	8	0	7	6.7484	5.0018	3.71094E-005	5
ex927	10	0	9	12.0688	16.9980	0.00000E+000	17
ex928	3	0	2	0.2000	1.5000	1.52588E-005	1.5
himmel11	9	0	3	3.5962	-30665.7678	2.83252E-004	-30665.54
house	8	4	4	30.2219	-3620.6989	3.67897E-007	-4500
least	3	0	0	$< 10^{-3}$	752888.0000	0.00000E+000	14085.1398
meanvar	7	0	2	0.8120	5.5642	1.20637E-005	5.2434
mhw4d	5	0	3	0.6800	0.0352	1.75659E-006	0.0293
process	8	0	6	13.0688	0.0000	1.33000E+002	-1161.3366
rbrock	2	0	0	$< 10^{-3}$	3918.5000	0.00000E+000	0
sample	4	2	0	0.5480	67033.3333	0.00000E+000	726.6367
wall	6	0	6	2.1761	-1.0004	2.41675E-004	-1

Table 2: Performance of the Algorithm on the GLOBALLib test set (Part II).

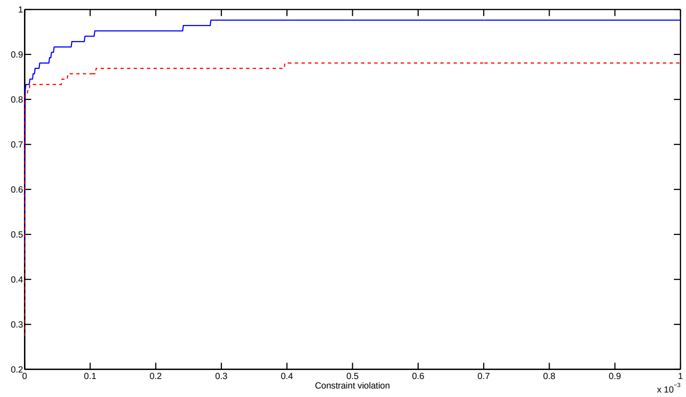

Figure 1: Performance Comparison between the DIRECT+DFN$_{con}$ Algorithm (continuous line) DF-EPGO+DFN$_{con}$ Algorithm (dashed line) on the GLOBALLib test set: number of feasible solutions.

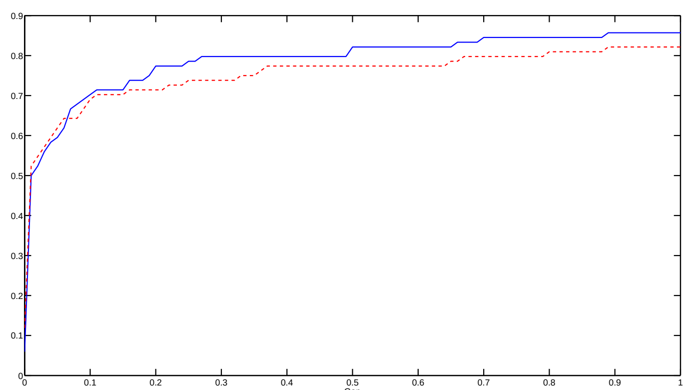

Figure 2: Performance Comparison between the DIRECT+DFN$_{con}$ Algorithm (continuous line) DF-EPGO+DFN$_{con}$ Algorithm (dashed line) on the GLOBALLib test set: number of best feasible solutions.

dimensions of problems that can be solved with our approach. On the other hand, by 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] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] G. Liuzzi, S. Lucidi, and V. Piccialli. Exploiting derivative-free local searches in direct-type algorithms for global optimization. Technical Report 14-01, IASI-CNR, 2014.

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

[14] 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.

[15] 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.

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