A Multi-objective DIRECT algorithm for ship hull optimization

E.F. Campana[*], M. Diez[*], G. Liuzzi[†], S. Lucidi[‡], R. Pellegrini[*], V. Piccialli[§], F. Rinaldi[¶], A. Serani[*]

December 20, 2016

Abstract. The paper is concerned with black-box nonlinear constrained multi-objective optimization problems. Our interest is the definition of a multi-objective deterministic partition-based algorithm. The main target of the proposed algorithm is the solution of a real ship hull optimization problem. To this purpose and in pursuit of an efficient method, we develop an hybrid algorithm by coupling a multi-objective DIRECT-type algorithm with an efficient derivative-free local algorithm. The results obtained on a set of “hard” nonlinear constrained multi-objective test problems show viability of the proposed approach. Results on a hull-form optimization of a high-speed catamaran (sailing in head waves in the North Pacific Ocean) are also presented. In order to consider a real ocean environment, stochastic sea state and speed are taken into account. The problem is formulated as a multi-objective optimization aimed at (i) the reduction of the expected value of the mean total resistance in irregular head waves, at variable speed and (ii) the increase of the ship operability, with respect to a set of motion-related constraints. We show that the hybrid method performs well also on this industrial problem.

Keywords: Multi-objective nonlinear programming, derivative-free optimization, DIRECT-type algorithm

AMS subject classification: 90C30, 90C56, 65K05

[*]Istituto Nazionale per Studi ed Esperienze di Architettura Navale, Consiglio Nazionale delle Ricerche, Via di Vallerano 139, 00128 Roma (Italy)

[†]Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, Via dei Taurini 19, 00185 Roma (Italy)

[‡]Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma (Italy)

[§]Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università degli Studi di Roma “Tor Vergata”, Via del Politecnico 1, 00133 Roma (Italy)

[¶]Dipartimento di Matematica, Università di Padova, Via Trieste 63, 35121 Padova (Italy)

1 Introduction

In this paper we are interested in the multi-objective nonlinear programming problem:

$$\begin{array}{lll} \min & (f_1(x), \dots, f_q(x))^\top & \\ s.t. & g_j(x) \le 0, & j = 1, \dots, m \\ & \ell_i \le x_i \le u_i, & i = 1, \dots, n \end{array} \quad (1)$$

where $q > 1$, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, q$, $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \dots, m$, $\ell_i, u_i \in \mathbb{R}$ are both finite and such that $\ell_i < u_i$, $i = 1, \dots, n$. We assume that f_i, $i = 1, \dots, q$, g_j, $j = 1, \dots, m$ are (Lipschitz) continuous functions and that no global information (convexity, Lipschitz constants, ...) on the problem is available. Furthermore, we assume all the problem functions to be of the *black-box* type, meaning that only function values are available and can be used to solve the problem. This is a common feature in many situations in which computation of the problem functions is the result of time-consuming and complex simulation programs. On top of this, function values are typically affected by noise. Hence, making use of finite differences to obtain function gradients is unpractical, if not untrustworthy.

Multi-objective constrained problems and, more particularly, black-box ones like Problem (1), are almost ubiquitous in real-world applications and very well-studied in the literature [5, 13, 23, 28]. Indeed, the situation in which two (or even more) conflicting performances are to be optimized is becoming more and more frequent in practice. In such situations, the classical optimality definition for single-objective problems must be replaced by the well-known Pareto optimality definition. The most important consequence of such a different optimality definition is that, in the multi-objective context, we have to expect a set of equivalent (or "non-dominated") solutions rather than a single point or a set of points with the same objective function value. Non-dominated solutions are (sort of) indistinguishable one another. This is because they represent trade-off solutions of the problem so that, if we seek for improvement of one objective function, we have to be prepared to accept a deterioration of one of the other objectives.

As concerns solution algorithms, the Pareto optimality definition brings along a peculiar distinction in the multi-objective context, namely the actual separation of roles between the "problem solver" and the "decision maker". The problem solver is the one in charge of finding one or more non-dominated solutions. The decision maker is the one that, given a set of "equivalent" non-dominated solution, picks one solution on the basis of her/his indisputable preferences. Such a distinction is at the basis of a classification of solution algorithms. Indeed, they can be roughly classified with respect to when preferences of the decision maker are considered, as follows.

- Methods *without preferences*: those methods in which preferences of the decision maker are completely disregarded and the problem is considered solved by finding a single non-dominated solution.

- Methods with *a priori* knowledge of preferences: those methods that are based on a complete knowledge of the decision maker preferences. Hence, the solution process is guided by these preferences so that the obtained solution is guaranteed to be acceptable for the decision maker.

- Methods with *a posteriori* knowledge of preferences: those methods in which preferences of the decision maker are taken into account at the end of the solution process. This means that a reasonably rich set of non-dominated solutions must be computed in order to give the decision maker sufficient freedom to select her/his preferred solution among the computed ones.

Although there is a relative interest in methods that are able to compute a single non-dominated solution, i.e. *a priori* and *without preferences* algorithms, methods that are able to reconstruct or to approximate the whole set of Pareto solutions, i.e. *a posteriori* methods, are of great importance.
In this respect, evolutionary algorithms [28] have long been used in that, by evolving a set of individuals, they naturally allow for the computation of a solution set rather than a single solution.

However, in recent years, some deterministic derivative-free algorithms for multi-objective problems [6, 22] have been proposed that belong to the class of *a posteriori* methods. Unfortunately, such algorithms more than frequently get trapped in local Pareto solutions, which can be quite disappointing especially for applications.
On the other hand, in the context of single-objective global optimization, partition-based algorithms [8, 12, 14, 18, 20] are among the most robust ones in the literature, especially for problems with only a limited number of decision variables, such as the industrial problem analyzed in the paper (which has four variables).

Therefore, our aim in the paper is to define a multi-objective `DIRECT` algorithm (`MODIR`). Further, since our main interest is in the solution of problems where the functions are time-consuming to compute, we test the proposed algorithm on a set of "hard" constrained nonlinear multi-objective problems by allowing no more than 20,000 function evaluations. With such a limit, `MODIR` is not able to perform satisfactorily when compared with a multi-objective genetic algorithm. For this reason, we also propose an hybrid algorithm which use `MODIR` to perform an initial exploration of the feasible domain and the local optimization algorithm `DFMO` [22] to refine the set of non-dominated solutions computed by `MODIR`.
The results obtained on the set of "hard" test problems and on the ship hull optimization problem, confirm the effectiveness of the proposed hybrid method.

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

$$\mathcal{D} = \{x \in \mathbb{R}^n : \ \ell_i \leq x_i \leq u_i, \ i = 1, \dots, n\} \tag{2}$$

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

$$\mathcal{F} = \{x \in \mathcal{D} : \ g_j(x) \leq 0, \ j = 1, \dots, m\}.$$

and we assume that $\mathcal{F} \neq \emptyset$, that is Problem (1) is feasible.

As already argued above, when dealing with several objective functions at a time, the concept of Pareto dominance is usually considered in the comparison of two points.

Definition 1 (Pareto dominance) *Given two points, $x, y \in \mathcal{F}$, we say that x dominates y ($x \prec y$) if $f_i(x) \leq f_i(y)$, for all $i = 1, \dots, q$, and an index $\bar{\imath} \in \{1, \dots, q\}$ exists such that $f_{\bar{\imath}}(x) < f_{\bar{\imath}}(y)$.*

Anyway, when coming to optimality, it may not be possible to find a point which is optimal for all the objectives simultaneously. This is the reason why the concept of Pareto dominance is also used to characterize global and local optimality into a multi-objective framework. More specifically, by means of the following two definitions, we are able to identify a set of non-dominated points (the so called Pareto front or frontier) which represents the set of (global or local) optimal solutions of a given multi-objective problem.

Definition 2 (Global Pareto optimality) *A point $x^\star \in \mathcal{F}$ is a global Pareto optimizer of Problem (1) if there does not exist a point $y \in \mathcal{F}$ such that $y \prec x^\star$.*

Definition 3 (Local Pareto optimality) *A point $x^\star \in \mathcal{F}$ is a local Pareto optimizer of Problem (1) if there does not exist a point $y \in \mathcal{F} \cap \mathcal{B}(x^\star, \rho)$ such that $y \prec x^\star$, for some $\rho > 0$.*

The paper is organized as follows. In section 2, the classical `DIRECT` algorithm for single-objective optimization is recalled. In section 3, the multi-objective `DIRECT` algorithm (`MODIR`) is introduced along with its main convergence property. Section 4 is devoted to the numerical experimentation of the newly proposed `MODIR` algorithm and to the comparison with the local derivative-free algorithm `DFMO` from [22]. In a subsection we also introduce an hybrid algorithm that combines `MODIR` and `DFMO`. Finally, results on a difficult ship hull optimization problem are reported. In section 5, we draw some conclusions and discuss possible lines of future investigations.

2 The original `DIRECT` algorithm

In this section we report a brief description of the original `DIRECT` algorithm for single-objective optimization. To this aim, let us consider for the moment the following (single-objective) optimization problem.

$$\begin{array}{l} \min \ f(x) \\ s.t.\ \mathbf{0} \leq x \leq \mathbf{1}, \end{array} \tag{3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{0}, \mathbf{1} \in \mathbb{R}^n$ denote, respectively, the vectors of all zero and one in n dimensions. Let $\mathcal{D}$ denote the feasible set of Problem (3), that is, $\mathcal{D} = \{x \in \mathbb{R}^n : \ \mathbf{0} \leq x \leq \mathbf{1}\}$. It is worth noting that every problem

$$\begin{array}{l} \min \ f(x) \\ s.t.\ \ell \leq x \leq u, \end{array}$$

Set $\mathcal{D}^0 = \mathcal{D}$, $\mathcal{H}_0 = \{D^0\}$, $I_0 = \{0\}$, c = center of D, $k = 0$

repeat

Use the *Identification* procedure to compute $I_k^* \subseteq I_k$

Use the *Partition* procedure to subdivide $\mathcal{D}^i$, for each $i \in I_k^*$

Define the new partition $\mathcal{H}_{k+1}$

Set $k = k + 1$

until (stopping criterion satisfied)

return $f_{min} = \min\{f(c) : c \in C_k\}$, $X_{min} = \{c \in C_k : f(c) = f_{min}\}$, with C_k = {centers of the hyperintervals in $\mathcal{H}_k$}

Figure 1: Sketch of the original DIRECT algorithm.

with $\ell, u \in \mathbb{R}^n$ and $-\infty < \ell_i < u_i < +\infty$, $i = 1, \ldots, n$, through suitable scaling can be restated as Problem (3).

The DIRECT algorithm proceeds by building finer and finer partitions of the initial domain $\mathcal{D}$. Let, at the beginning of the k-th iteration of the algorithm,

$$\mathcal{H}_k = \{\mathcal{D}^i : i \in I_k\},$$

be a partition of $\mathcal{D}$, with

$$\mathcal{D}^i = \{x \in \mathbb{R}^n : \ell^i \leq x \leq u^i\}, \text{ for all } i \in I_k,$$

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

Within the current partition $\mathcal{H}_k$ of $\mathcal{D}$, a set of *potentially optimal* hyperintervals $\{\mathcal{D}^i : i \in I_k^*\}$ is selected by means of a so-called *identification procedure*. Then, each hyperinterval $\mathcal{D}^i$, $i \in I_k^*$, is further partitioned into smaller hyperintervals by means of a so-called *partition procedure*, thus giving birth to partition $\mathcal{H}_{k+1}$ for the next iteration. The latter two-phase process continues until a prescribed number of iterations or function evaluations has been performed or another more sophisticated stopping criterion is met (see, e.g., [20, 24, 26]). In Figure 1, a very basic scheme of the DIRECT algorithm is reported

The identification procedure, basically selects the more "promising" hyperintervals, i.e. those ones which are more likely to contain a global minimum point of Problem (3). More specifically, let us recall the following definition of potentially optimal hyperinterval [16].

Definition 4 (Potential optimality) *Given a partition $\mathcal{H}_k = \{\mathcal{D}^i : i \in I_k\}$ of $\mathcal{D}$ and a scalar $\varepsilon > 0$, an hyperinterval $\mathcal{D}^h \in \mathcal{H}_k$ is* potentially optimal *if a constant $\bar{L}^h$ exists such* *that:*

$$f(x^h) - \frac{\bar{L}^h}{2}\|u^h - \ell^h\| \le f(x^i) - \frac{\bar{L}^h}{2}\|u^i - \ell^i\|, \qquad \forall\, i \in I_k \tag{4}$$

$$f(x^h) - \frac{\bar{L}^h}{2}\|u^h - \ell^h\| \le f_{\min} - \epsilon|f_{\min}| \tag{5}$$

where

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

As it can be noted, the notion of potential optimality is based on some measure of the hyperinterval itself and on the value of $f(x)$ at its center. The identification procedure, at iteration k, computes the set I_k^* of all potentially optimal hyperintervals indices.

As for the partition procedure, let us suppose that hyperinterval $\mathcal{D}^h$, $h \in I_k^*$, has been selected for further subdivision. Let δ and J be the measure of the longest edge and, respectively, the set of the longest edges of $\mathcal{D}^h$, i.e.,

$$\begin{aligned} \delta &= \max_{1 \le j \le n} (u^h - \ell^h)_j, \text{ and,} \\ J &= \{j \in \{1, \dots, n\} : (u^h - \ell^h)_j = \delta\}, \end{aligned}$$

Then, $2m$, with $m = |J|$, new points are defined, i.e., for every $j \in J$,

$$c^{h_j} = c^h + \frac{\delta}{3} e_j, \qquad c^{h_{j+m}} = c^h - \frac{\delta}{3} e_j,$$

and the objective function is evaluated at these new points. Finally, hyperinterval $\mathcal{D}^h$ is partitioned into $2m + 1$ smaller hyperintervals having the latter points plus c^h as their centers. If there are multiple longest edges, the partition is carried out in such a way that the biggest new hyperintervals contain the points with the best function values.

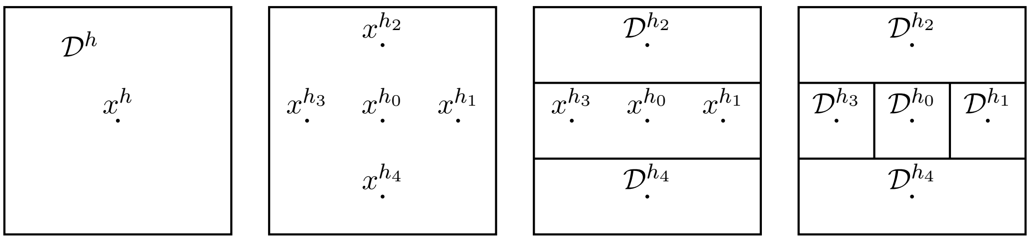

Figure 2: Hyperinterval partitioning example

Figure 2 shows an illustrative example of the behavior of the partition procedure. In the example we assume that the lowest objective function value is attained at the point x^{h_4} and this determines the order in which edges are subdivided, thus making $\mathcal{D}^{h_4}$ one of the largest hyperintervals after subdivision.

The minimum of f over all the centers of the final partition, and the corresponding centers, provide an approximate solution to the problem.

Further details on the original DIRECT algorithm can be found in [16, 19]. The convergence of DIRECT is proved (see, e.g., [16, 19]) by showing that the set of sampled points, i.e. hyperinterval's centers, becomes everywhere dense in $\mathcal{D}$ as the number of iterations k goes to infinity. For a thorough convergence analysis of DIRECT-type algorithms, we refer the interested reader to [17, 20, 25, 26].

3 The multi-objective DIRECT (MODIR) algorithm

In this section, we extend the basic DIRECT algorithm for single-objective optimization problems, to the solution of the multi-objective optimization Problem (1). To this aim, two aspects are to be specified, namely,

i) the handling of the nonlinear constraints and

ii) the definition of potential optimality for multi-objective problems.

These two aspects will be discussed in the following.

Handling of the nonlinear constraints. Drawing inspiration from [22], given Problem (1) and a vector of (penalty) parameters $\varepsilon_i > 0$, $i = 1, \dots, m$, we introduce the following penalty functions

$$Z_j(x;\varepsilon) = f_j(x) + \sum_{i=1}^{m} \frac{1}{\varepsilon_i} \max\{0, g_i(x)\}, \quad \text{for all } j = 1, \dots, q.$$

Then, in place of the nonlinearly constrained Problem (1), we define the following penalized and bound constrained multi-objective problem

$$\begin{array}{l} \min\ Z(x;\varepsilon) = (Z_1(x;\varepsilon), \dots, Z_q(x;\varepsilon))^\top \\ s.t.\ \ell \le x \le u, \end{array} \tag{6}$$

where $\varepsilon = (\varepsilon_1, \dots, \varepsilon_m)^\top$. Note that, as showed in [22], for sufficiently small values of the penalty parameters ε_i, $i = 1, \dots, m$, solving Problem (6) is equivalent to solving Problem (1).

Multi-objective potential optimality. In the case of single-objective optimization, the first inequality of Definition 4 basically states that an hyperinterval is deemed potentially optimal when it is able to provide a lower bound on the objective function which is equal to (if not better than) the best lower bound provided by all other hyperintervals. When more than one objective function is present, one might extend Definition 4 by saying that $\mathcal{D}^h$ is potentially optimal if a constant $\bar{L}^h$ exists such that, for all other $\mathcal{D}^i$,

$$f_j(x^h) - \frac{\bar{L}^h}{2}\|u^h - \ell^h\| \le f_j(x^i) - \frac{\bar{L}^h}{2}\|u^i - \ell^i\|, \quad \forall\ j = 1, \dots, q,$$

i.e., $\mathcal{D}^h$ is potentially optimal when it is potentially optimal with respect to all the objective functions at the same time. Unfortunately, for non-trivial multi-objective problems, such a definition would be useless since no such hyperintervals might exist. A more thorough analysis of the situation led us to the introduction of a condition weaker than the above one. More in particular, we introduce the following definition of potential Pareto-optimality.

Definition 5 (Potential Pareto-optimality) *Given a partition* $\{\mathcal{D}^i : i \in I_k\}$ *of* $\mathcal{D}$*, an hyperinterval* $\mathcal{D}^h$ *is* potentially Pareto-optimal *if a constant* $\bar{L}^h$ *exists such that, for all* $i \in I_k$ *and index* $j_i \in \{1, \dots, q\}$ *exists which satisfies*

$$f_{j_i}(x^h) - \frac{\bar{L}^h}{2}\|u^h - \ell^h\| \le f_{j_i}(x^i) - \frac{\bar{L}^h}{2}\|u^i - \ell^i\|. \tag{7}$$

Concerning the above definition, it can be noted that an hyperinterval $\mathcal{D}^h$ is deemed potentially Pareto-optimal if, for all other $\mathcal{D}^i$, inequality (7) holds for at least an objective function index. Furthermore, we also note that the set of potentially Pareto-optimal hyperintervals is a superset of potentially optimal hyperintervals with respect to every single-objective function. More precisely, let us define $I_k^{j,*}$ the set of potentially optimal hyperintervals indices with respect to objective function $f_j(x)$, $j = 1, \dots, q$, and I_k^* the set of potentially Pareto-optimal hyperintervals indices, then

$$I_k^{j,*} \subseteq I_k^*, \quad \text{for all } j = 1, \dots, q.$$

Furthermore, we remark that, from Definition 5, it follows that set I_k^* contains at least an index of one of the largest hyperintervals belonging to the current partition $\mathcal{H}_k$. Namely,

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

where

$$I_k^{max} = \left\{i \in I_k : \left\|u^i - \ell^i\right\| = d_k^{max}\right\}, \; d_k^{max} = \max_{i \in I_k} \left\|u^i - \ell^i\right\|.$$

The latter consideration is of paramount importance since it allows us to conclude that the set of hyperintervals centers becomes dense in the initial domain $\mathcal{D}$ as the iterations go to infinity (see Proposition 2 of [20]). Then, at least in the limit, the algorithm would sample a point arbitrarily close to any given Pareto optimal solution of Problem (6) and, provided that ε is sufficiently small, of the original constrained Problem (1).

Algorithm `MODIR`. When it comes to the actual implementation of the identification procedure, namely the selection of the potential Pareto-optimal hyperintervals (according to Definition 5), it turns out that such a procedure would be very expensive so as to make the algorithm too slow to be useful even for small problems. Hence, a more efficient procedure must be defined which approximates the "ideal" selection procedure.

To this aim, it is worth recalling Definition 4 for single-objective problems and considering what inequalities (4) and (5) imply. To better understand the meaning of inequalities (4) and (5), in Figure 3 we report the partition $\mathcal{H}$ obtained by `DIRECT` at a generic iteration. Specifically, in the graph, every hyperinterval $\mathcal{D}^i \in \mathcal{H}$ (with $i \in I$) is represented by a point on the Cartesian plane with hyperinterval diameters on the x-axis and centroid function values on the y-axis. Now, inequality (4) in Definition 4 forces (potentially optimal) hyperintervals to be on the lower right part of the convex hull of the set of points, except for the points not satisfying inequality (5). Further, note that the set of potentially optimal hyperintervals in Figure 3 is a subset of the set of points that are not dominated when we consider the bi-objective problem

$$\min_{i \in I} \; (f(c^i), -||u^i - \ell^i||)^\top.$$

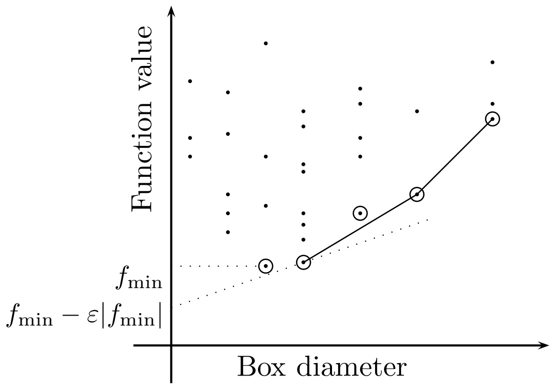

Figure 3: Potentially optimal hyperintervals

Such points are evidenced by circles in Figure 3.
In a multi-objective context, we might extend the above reasoning by selecting those hyperintervals whose representative points are not dominated for the multi-objective problem (with $q+1$ objective functions)

$$\min_{i\in I} \ (Z_1(c^i), \dots, Z_q(c^i), -||u^i - \ell^i||)^\top. \tag{8}$$

To be more precise, let $\mathcal{H} = \{\mathcal{D}^i : \ i \in I\}$ be the current partition of the initial domain. Given a penalty parameter $\varepsilon > 0$ and set I defining $\mathcal{H}$, let us define

$$\mathcal{S}_I = \left\{ (Z_1(c^i;\varepsilon), \dots, Z_q(c^i;\varepsilon), -\|u^i - \ell^i\|)^\top : \ i \in I \right\}.$$

The (multi-objective) identification procedure that we propose consists in defining set I^* as the set of indices of non-dominated $(q+1)$-tuples belonging to the finite set $\mathcal{S}_I$.

Summarizing, algorithm `MODIR` can be obtained from the `DIRECT` algorithm by replacing the (single-objective) *identification* procedure of `DIRECT` with the multi-objective identification procedure just described.
As concerns the theoretical properties of algorithm `MODIR`, they are a straightforward consequence of the definition of both the *partition* and *identification* procedures. Let us recall from [20], that algorithm `MODIR` produces certain sequences of hyperintervals $\{\mathcal{D}^{i_k}\}$. These sequences can be defined by associating to every hyperinterval $\mathcal{D}^{i_k}$, with $i_k \in I_k$, a predecessor $\mathcal{D}^{i_{k-1}}$, with $i_{k-1} \in I_{k-1}$, as follows:

- if hyperinterval $\mathcal{D}^{i_k}$ has been generated at the k-th iteration, then $\mathcal{D}^{i_{k-1}}$ is the hyperinterval which has been partitioned at the k-th iteration and which has generated hyperinterval $\mathcal{D}^{i_k}$;
- if hyperinterval $\mathcal{D}^{i_k}$ has not been generated at the k-th iteration than $\mathcal{D}^{i_{k-1}} = \mathcal{D}^{i_k}$.

Then, by definition, all the sequences $\{\mathcal{D}^{i_k}\}$ are *nested* sequences, i.e. sequences such that $\mathcal{D}^{i_k} \subseteq \mathcal{D}^{i_{k-1}}$. Among such sequences, of particular importance are the *strictly nested* sequences, namely those sequences for which $\mathcal{D}^{i_k} \subset \mathcal{D}^{i_{k-1}}$ infinitely many times.
Now, for algorithm `MODIR` the following proposition can be stated.

Set $\mathcal{D}^0 = \mathcal{D}$, $\mathcal{H}_0 = \{D^0\}$, $I_0 = \{0\}$, $c =$ center of $\mathcal{D}$, $k = 0$

repeat

Use the (multi-objective) *Identification* procedure to compute $I_k^* \subseteq I_k$

Use the *Partition* procedure to subdivide $\mathcal{D}^i$, for each $i \in I_k^*$

Define the new partition $\mathcal{H}_{k+1}$

Set $k = k + 1$

until (stopping criterion satisfied)

return $P \equiv \{c \in C_k : \nexists\, \tilde{c} \in C_k$ that dominates $c\}$

Figure 4: Sketch of Algorithm `MODIR`.

Proposition 1 *The following statements hold for algorithm* `MODIR`*.*

(i) All the sequences of hyperintervals $\{\mathcal{D}^{i_k}\}$ *produced by algorithm* `MODIR` *are strictly nested.*

(ii) For every $\tilde{x} \in \mathcal{D}$*, algorithm* `MODIR` *produces a strictly nested sequence of hyperintervals* $\{\mathcal{D}^{i_k}\}$ *such that*

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

Proof. Since the set I^* is set of indices of non-dominated $(q+1)$-tuples belonging to the finite set $\mathcal{S}_I$ and since the last elements of the $(q+1)$-tuples of the set $\mathcal{S}_I$ are given by $-\|u^i - \ell^i\|)$ for $i \in I$, it results that

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

Then points (i) and (ii) of the proposition follow again from Proposition 2 of [20] □

We note that in the single-objective global optimization field the `DIRECT` algorithm is well known for having a good ability to locate interesting regions of the feasible domain, but also for having a very slow convergence to a satisfactory approximation of the global minimum, due to the generation of a dense set of points covering the whole feasible set. In order to speed up its convergence it is a common technique to integrate the `DIRECT` algorithm with a local search engine (see for example [2, 8, 19, 21]). This strategy will have the same beneficial effects on our multi-objective version, leading to our hybrid algorithm introduced in Section 4.2.

4 Numerical results

This section is devoted to the experimentation and comparison of the proposed multi-objective approach. To this aim and considering that our approach is for constrained

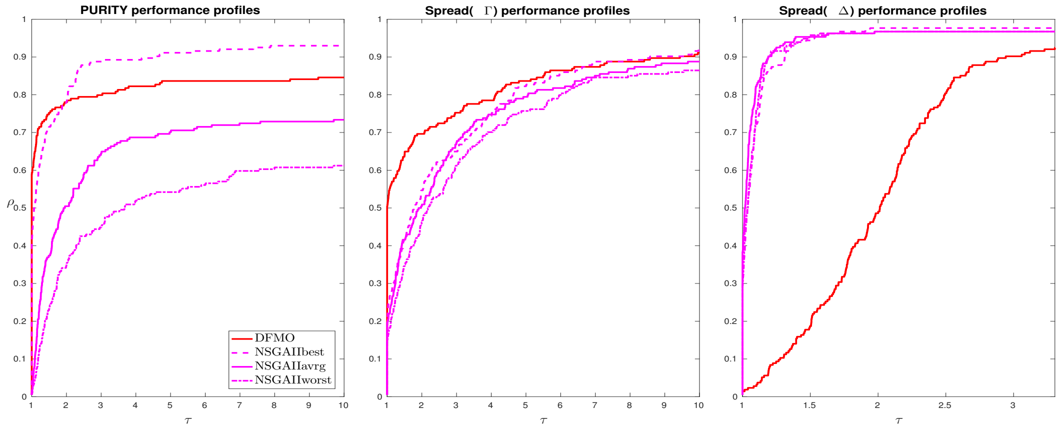

Figure 5: Comparison between DFMO and NSGA-II by using performance profiles with the metrics Purity (left), Spread(Γ) (center), and Spread(Δ) (right).

multi-objective global optimization problems, we adopt the collection of problems recently proposed in [22]. This collection is composed by 214 nonlinearly constrained multi-objective problems with $n \in [3, 30]$, $q \in [2, 4]$ and $m \in [1, 29]$. To have a feeling of how difficult these problems are in terms of *global* Pareto optimality, in Figure 5 we report comparison between the local method proposed in [22] DFMO and the well-known genetic algorithm NSGA-II [7]. The idea is to select those problems where there is a need for a global optimization method as NSGA-II, neglecting those problems where DFMO has already satisfactory performances. Both the codes have been run using their default settings for a maximum of 20,000 function evaluations. NSGA-II has been run with a population of 100 individuals which, given the maximum number of function evaluations, amounts to 200 generations. Furthermore, since NSGA-II is a probabilistic code, for every problem we run it ten times with different seeds of the random number sequence. Then, let $\mathcal{P}_{i,j}$ be the set of non-dominated points found by the j-th run of NSGA-II on the i-th problem, and let

$$\mathcal{P}_i = \text{nondom}\left(\bigcup_{j=1}^{10} \mathcal{P}_{i,j}\right).$$

Then, for problem i and run j, let $v_{i,j} = |\mathcal{P}_i \cap \mathcal{P}_{i,j}|$, i.e. the number of points found by run j that are on the Pareto front defined by all of the runs. Using numbers $v_{i,j}$, we can define, for every problem i, the best worst and median run of NSGA-II. More precisely, the best run is run j_{max} such that $v_{i,j_{max}}$ is maximum; the worst run is run j_{min} such that $v_{i,j_{min}}$ is minimum; the median run is run $\bar{\jmath}$ such that $v_{i,\bar{\jmath}}$ is the median value of the $v_{i,j}$'s.

The results of the comparison between DFMO and NSGA-II are reported in Figure 5 in terms of the Purity [1] and Spread metrics Γ and Δ (both metrics have been defined in [6]) by using performance profiles [11].

As it can be noted, DFMO performs quite well with respect to the average version of NSGA-II. Furthermore, we can say that DFMO has a better efficiency, in terms of purity, than the best version of NSGA-II, even though it is less robust. This fact is indeed quite relevant since it tells that the considered problems might not be too difficult in terms of global Pareto optimality. However, this could also be explained by saying that the great majority of problems are not difficult ones.

4.1 Results on hard test problems

Hence, among the 214 problems, we extract a subset of 38 "hard" problems, namely those problems onto which DFMO performs poorly with respect to NSGA-II in terms of purity, that is the most suitable metric to evaluate the global ability of a multi-objective optimization method (see, e.g., [1, 6]). In Table 1 we report the features of these 38 "hard" problems. In particular, we report, for each problem, the number n of variables, the number m of nonlinear constraints, and the number q of objective functions. The column labeled "Problem" reports the name of the constrained problem. This is composed by the name of the unconstrained problem from the collection of [6], and by a parenthesized letter which defines the constraint family according to the following table:

(a)	$g_j(x) = (3 - 2x_{j+1})x_{j+1} - x_j - 2x_{j+2} + 1 \leq 0,$	$j = 1, \ldots, m,\ m = n - 2;$
(b)	$g_j(x) = (3 - 2x_{j+1})x_{j+1} - x_j - 2x_{j+2} + 2.5 \leq 0,$	$j = 1, \ldots, m,\ m = n - 2;$
(c)	$g_j(x) = x_j^2 + x_{j+1}^2 + x_j x_{j+1} - 2x_j - 2x_{j+1} + 1 \leq 0,$	$j = 1, \ldots, m,\ m = n - 1;$
(d)	$g_j(x) = x_j^2 + x_{j+1}^2 + x_j x_{j+1} - 1 \leq 0,$	$j = 1, \ldots, m,\ m = n - 1;$
(e)	$g_j(x) = (3 - 0.5x_{j+1})x_{j+1} - x_j - 2x_{j+2} + 1 \leq 0,$	$j = 1, \ldots, m,\ m = n - 2;$
(f)	$g_j(x) = \sum_{i=1}^{n+1} ((3 - 0.5x_{j+1})x_{j+1} - x_j - 2x_{j+2} + 1) \leq 0,$	$j = 1, \ldots, m,\ m = 1.$

The comparison between DFMO and NSGA-II is reported in Figure 6. As we can see, now the situation, at least in terms of purity, is almost reversed with DFMO performance near to the worst version of NSGA-II rather than above the average version as in Figure 5. This, in our opinion, reasonably qualifies these 38 problems as hard multi-objective problems from the point of view of global Pareto optimality.

Now, in Figure 7, we report the comparison between algorithms MODIR and NSGA-II on the set of hard problems. As it can be noted, the performance of MODIR is quite disappointing since its profiles are worse than the worst version of NSGA-II. However, these results are not that much unexpected. Indeed, the behavior of MODIR can be explained (to a large extent) by recalling the limit of 20,000 function evaluations we imposed on the runs, and the deterministic nature of algorithm MODIR. Such a limit is definitely too low for MODIR. In fact, as a partitioning algorithm, MODIR spends many function evaluations exploring "unattractive" regions of the feasible domain simply because at each iteration the largest hyperinterval must be selected and further partitioned. This is a common feature (and the main curse alas) of DIRECT-type algorithms. They exhibit quite a slow convergence, meaning that they are quite robust but at the expense of a "huge" amount of function evaluations. Particularly, it turns out that quite a large number of function evaluations is consumed to guarantee the "everywhere dense" convergence property, which is why DIRECT-type algorithms improve the estimate of the solution set slower and slower as the iteration count grows.

Problem	n	m	q	Problem	n	m	q
DTLZ3 (c)	12	11	3	L3ZDT4 (a)	30	28	2
DTLZ3 (d)	12	11	3	L3ZDT4 (c)	30	29	2
DTLZ4 (d)	12	11	3	L3ZDT6 (a)	10	8	2
FES1 (a)	10	8	2	L3ZDT6 (c)	10	9	2
FES3 (a)	10	8	4	MOP2 (e)	4	2	2
I2 (a)	8	6	3	MOP2 (f)	4	1	2
I3 (c)	8	7	3	OKA2 (c)	3	2	2
I5 (c)	8	7	3	QV1 (a)	10	8	2
L1ZDT4 (a)	10	8	2	QV1 (f)	10	1	2
L1ZDT4 (c)	10	9	2	TKLY1 (c)	4	3	2
L1ZDT4 (f)	10	1	2	TKLY1 (d)	4	3	2
L2ZDT2 (a)	30	28	2	WFG1 (a)	8	6	3
L2ZDT2 (c)	30	29	2	WFG1 (b)	8	6	3
L2ZDT3 (c)	30	29	2	ZDT1 (a)	30	28	2
L2ZDT6 (a)	10	8	2	ZDT2 (a)	30	28	2
L2ZDT6 (c)	10	9	2	ZDT4 (a)	10	8	2
L3ZDT1 (c)	30	29	2	ZDT4 (b)	10	8	2
L3ZDT2 (a)	30	28	2	ZDT4 (f)	10	1	2
L3ZDT3 (a)	30	28	2	ZDT6 (a)	10	8	2

Table 1: Characteristics of the hard multi-objective problems

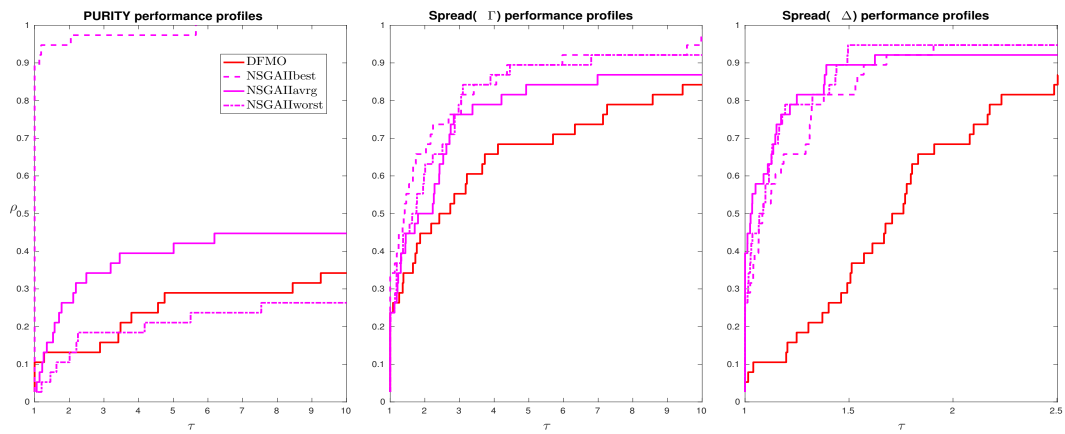

Figure 6: Comparison between `DFMO` and `NSGA-II` on the 38 hard problems by using performance profiles with the metrics Purity (left), Spread(Γ) (center), and Spread(Δ) (right).

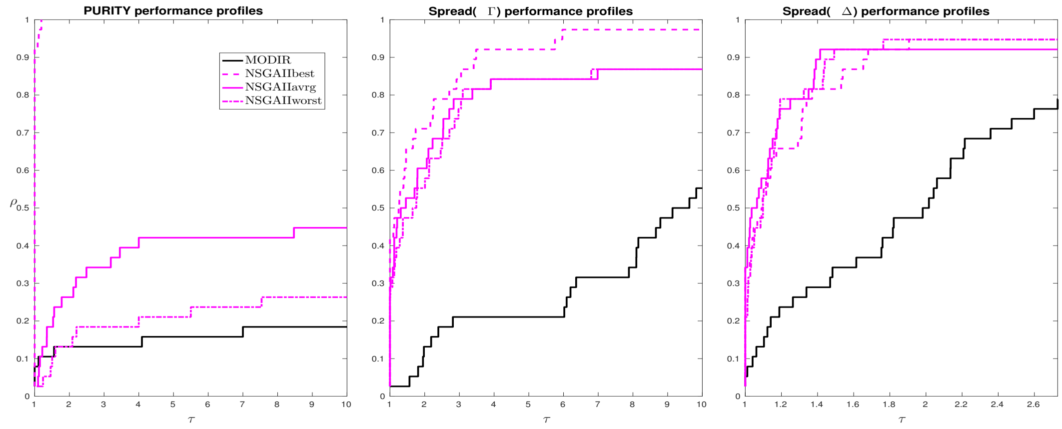

Figure 7: Comparison between MODIR and NSGA-II on the 38 hard problems by using performance profiles with the metrics Purity (left), Spread(Γ) (center), and Spread(Δ) (right).

Let `maxnf` be the budget of function evaluations and $\gamma \in [0, 1]$

(Phase 1) Compute P_0 running `MODIR` for $\gamma\cdot$`maxnf` function evaluations

(Phase 2) Compute P^* running `DFMO` for $(1-\gamma)\cdot$`maxnf` function evaluations starting from points in P_0

return P^*

Figure 8: Sketch of Algorithm `MODIR+DFMO`.

4.2 The hybrid scheme `MODIR+DFMO`

However, since we do believe in the good global ability of `MODIR`, drawing inspiration from the approaches proposed in [8, 19, 2, 21] we can devise an hybrid algorithm by gluing together `MODIR` and the local algorithm `DFMO` proposed in [22].
Indeed, a partial remedy to the slow convergence problem can be that of using `MODIR` to carry out a first (rough) exploration of the search space thus producing some (rough) estimate P of the solution set, i.e. the set of Pareto optimal points of Problem (1). Then, set P is further improved by means of the "faster" multi-objective local optimization algorithm `DFMO`.
More specifically, let P_k be the set of non-dominated points at the beginning of iteration k of algorithm `DFMO`. Through the use of multi-objective linesearches along suitable directions, `DFMO` tries to improve set P_k into set P_{k+1}. For a detailed description of algorithm `DFMO` and for a thorough analysis of its convergence we refer the interested reader to [22].
In Figure 8, we report a sketch of the hybrid scheme that we propose.
Note that, when `maxnf` $= +\infty$,

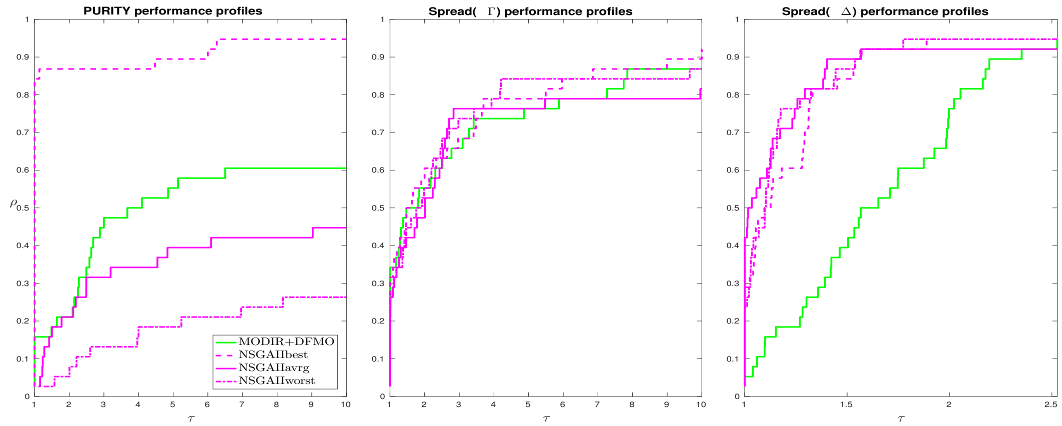

Figure 9: Comparison between MODIR+DFMO and NSGA-II on the 38 hard problems by using performance profiles with the metrics Purity (left), Spread(Γ) (center), and Spread(Δ) (right).

- MODIR+DFMO turns to be MODIR provided that $\gamma = 1$;
- MODIR+DFMO turns to be DFMO provided that $\gamma = 0$.

In Figure 9, we report the comparison of the proposed hybrid algorithm where we set maxnf$= 20,000$ and

$$\gamma = \frac{500 \cdot n}{\texttt{maxnf}}. \tag{9}$$

As it can be noted, the performances of the hybrid algorithm, with respect to NSGA-II on the hard problems, are quite better than those of MODIR alone. In fact, in terms of purity, MODIR+DFMO is better than the average version of NSGA-II, and this is the most significant performance measure in order to evaluate the global efficiency of our method.

4.3 Results on a real problem

In this subsection we consider an application related to a ship hull optimization problem. More in particular, the industrial application presented pertains to the reliability-based robust optimization of the hull form of a 100 m high-speed catamaran [3, 10], sailing in head waves in the North Pacific Ocean. Figure 10 shows the model used at CNR-INSEAN for the experiments and an example of the wave pattern obtained by URANS simulation [15].

The problem is formulated as a multi-objective optimization aimed at (I) the reduction of the expected value of the mean total resistance in irregular head waves at variable speed and (II) the increase of the ship operability, with respect to a set of motion-related constraints.

Specifically, the following problem is considered.

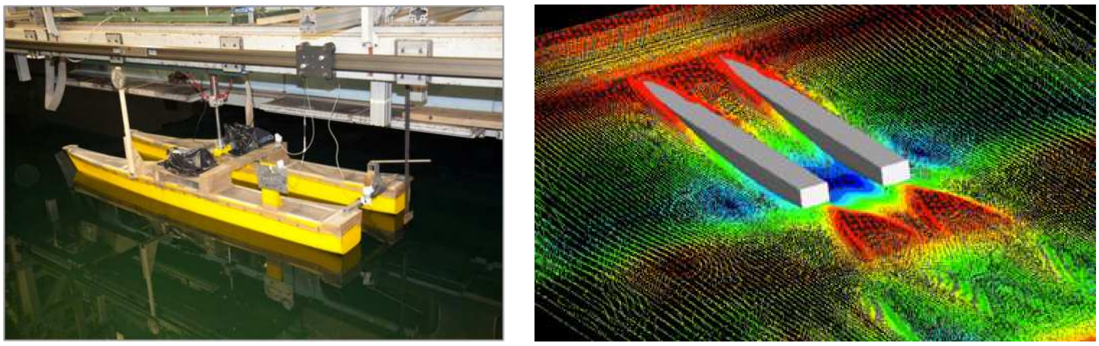

Figure 10: High-speed catamaran: (a) CNR-INSEAN model and (b) URANS wave pattern [15]

$$
\begin{array}{ll}
\min & (\phi_1(x),\ -\phi_2(x))^\top \\
s.t. & \phi_1(x) \le 0,\ \ \phi_2(x) \ge 0, \\
 & g(x) \le 0, \\
 & \ell \le x \le u
\end{array}
\tag{10}
$$

where ϕ_1 and ϕ_2 are the expected value of the mean total resistance and the ship operability evaluated in irregular head wave for variable sea state and speed, respectively defined as

$$
\phi_1(x) = \iint_{S,U} \bar{R}_T\left(x,S,U\right) p\left(S,U\right) dU dS \tag{11}
$$

$$
\phi_2(x) = \iint_{S,U} \bigcap_{j=1}^{J} [h_j\left(x,S\right) \le 0] p\left(S,U\right) dU dS \tag{12}
$$

where $\bar{R}_T$ is the mean value of the total resistance in irregular waves, x is the design variable vector, S is the sea state, U is the speed, h_j are the motion constraints, and p is the joint probability density function of S and U. g is a geometrical constraint related to the maximum overall beam. Details may be found in [4].
The design optimization problem is taken from [10]. For the sake of the present study, objective function values are obtained by means of stochastic radial-basis functions interpolation (details may be found in [27]) of high-fidelity URANS simulations. Four design variables (x) control the global shape modifications of the catamaran hull, based on the Karhunen-Loève expansion of the shape modification vector [9]. The inequality constraints in Problem (10) are needed to guarantee improvement with respect to the initial design.

Figure 11(a) shows $\tilde{P}$, preliminary (and putative) Pareto solutions of Problem (10) which can thus be considered a trade-off between operability and total resistance of the catamaran.

In Figure 11(b) comparison between $\tilde{P}$ and the front found by `DFMO` (with `maxnf`$= 20,000$)

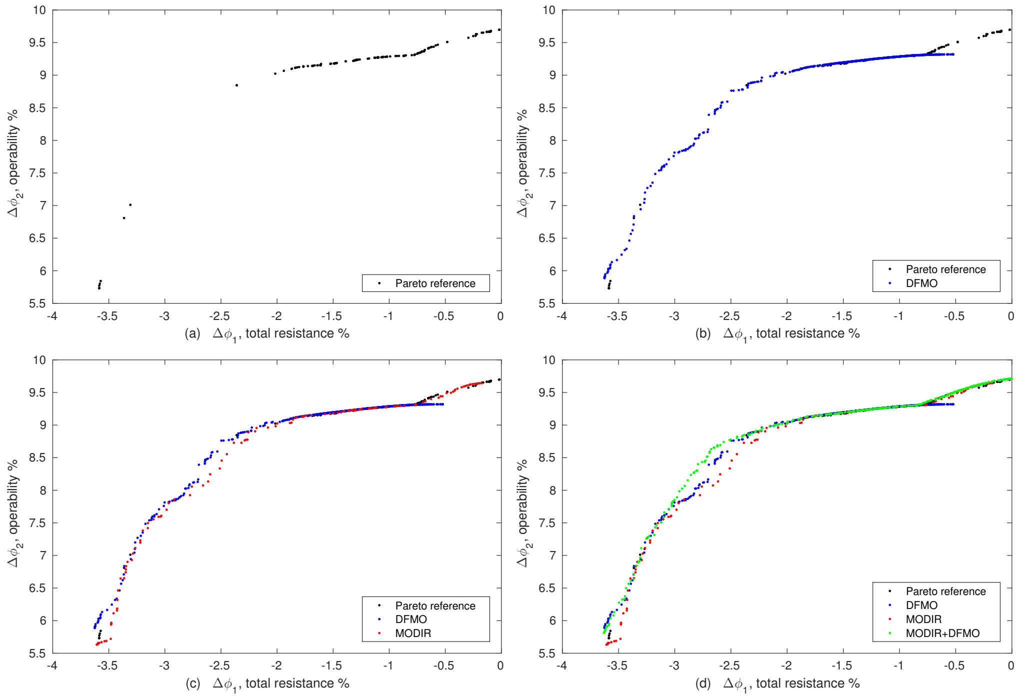

Figure 11: Results for the ship hull optimization problem. Putative Pareto points are reported in plot (a). Plots (b), (c), and (d) report comparison between points found by DFMO, MODIR, and MODIR+DFMO, respectively, with the putative solutions.

is reported. As we can see, the behavior of DFMO is satisfactory except for the upper right part of the front where DFMO evidently got trapped in local solutions of the problem.

The front found by algorithm MODIR (with maxnf$= 20,000$) can be found in Figure 11(c). Here, we can see that MODIR correctly reconstructs the upper right part of the front. However, it again emerges the main drawback of MODIR when a relative small number of function evaluations are allowed.

Finally, in Figure 11(d), we report the front found by the proposed hybrid algorithm MODIR+DFMO with maxnf$= 20,000$ and γ given by (9). It can be seen that the hybrid algorithm reconstructs the upper right part of the front and also improves the central part of the front, thus being the best algorithm for the considered ship hull optimization problem.

5 Conclusions

In the paper, we presented MODIR, a multi-objective DIRECT algorithm for constrained multi-objective problems. Furthermore, we proposed a hybrid method (MODIR+DFMO) which uses the local algorithm DFMO to refine a set of non-dominated solutions computed

by an initial exploration of the search domain by MODIR.
Algorithm MODIR is based on a brand new definition of Pareto potential optimality which extends to the multi-objective context the potential optimality definition used within the original DIRECT algorithm. We proposed a way to efficiently compute a superset of the set of Pareto potentially optima hyperintervals.
In the numerical result section, we report the results obtained with algorithms MODIR and MODIR+DFMO, on a set of "hard" test problems, and the comparison with the well-known genetic algorithm NSGA-II for multi-objective problems. Further, we describe an industrial problem related to the ship hull optimization of a high-speed catamaran. The results obtained with the hybrid algorithm are very promising and confirm the efficiency and effectiveness of the proposed approach.
Among the possible lines of future investigations, we would like to consider the following two:

i) Study of more efficient ways to approximate (or to compute) the set of Pareto potentially optimal hyperintervals (recall Definition 5) within the MODIR algorithm.

ii) Development of more clever ways to connect MODIR and DFMO in the hybrid scheme. In particular, it would be advisable that calls to MODIR and DFMO were not simply sequential but rather interspersed one with another using a bi-directional exchange of information between global and local search engines.

References

[1] S. Bandyopadhyay, S.K. Pal, and B. Aruna. Multiobjective GAs, quantitative indices, and pattern classification. *Systems, Man, and Cybernetics, Part B: IEEE Transactions on Cybernetics*, 34(5):2088–2099, 2004.

[2] Emilio F. Campana, Matteo Diez, Umberto Iemma, Giampaolo Liuzzi, Stefano Lucidi, Francesco Rinaldi, and Andrea Serani. Derivative-free global ship design optimization using global/local hybridization of the DIRECT algorithm. *Optimization and Engineering*, 17(1):127–156, 2015.

[3] X. Chen, M. Diez, M. Kandasamy, E.F. Campana, and F. Stern. Design optimization of the waterjet-propelled delft catamaran in calm water using urans, design of experiments, metamodels and swarm intelligence. In *Proceedings of the 12th international conference on fast sea transportation (FAST2013), Amsterdam, The Netherlands*, 2013.

[4] Xi Chen, Matteo Diez, Manivannan Kandasamy, Zhiguo Zhang, Emilio F. Campana, and Frederick Stern. High-fidelity global optimization of shape design by dimensionality reduction, metamodels and deterministic particle swarm. *Engineering Optimization*, 47(4):473–494, 2015.

[5] A. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

[6] A.L. Custódio, J.F.A. Madeira, A.I.F. Vaz, and L.N. Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011.

[7] 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.

[8] G. Di Pillo, G. Liuzzi, S. Lucidi, V. Piccialli, and F. Rinaldi. A direct-type approach for derivative-free constrained global optimization. *Computational Optimization and Applications*, 65(2):361–397, 2016.

[9] M. Diez, E.F. Campana, and F. Stern. Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 283:1525–1544, 2015.

[10] M. Diez, X. Chen, E.F. Campana, and F.Stern. Reliability-based robust design optimization for ships in real ocean environment. 12th International Conference on Fast Sea Transportation, FAST2013, Amsterdam, The Netherlands, 2013.

[11] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[12] J.M. Gablonsky and C.T. Kelley. A locally-biased form of the DIRECT algorithm. *Journal of Global Optimization*, 21(1):27–37, 2001.

[13] M. Gen, R. Cheng, and L. Lin. *Multiobjective Genetic Algorithms*, pages 1–47. Springer, 2008.

[14] J. He, A. Verstak, L.T. Watson, and M. Sosonkina. Design and implementation of a massively parallel version of DIRECT. *Computational Optimization and Applications*, 40:217–245, 2008.

[15] W. He, M. Diez, Z. Zou, E.F. Campana, and F. Stern. URANS study of Delft catamaran total/added resistance, motions and slamming loads in head sea including irregular wave and uncertainty quantification for variable regular wave and geometry. *Ocean Engineering*, 74:189 – 217, 2013.

[16] 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.

[17] D.E. Kvasov and Ya.D. Sergeyev. Deterministic approaches for solving practical black-box global optimization problems. *Advances in Engineering Software*, 80:58–66, 2015.

[18] Q. Liu and J. Zeng. Global optimization by multilevel partition. *Journal of Global Optimization*, 61(1):47–69, 2015.

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

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

[21] G. Liuzzi, S. Lucidi, and V. Piccialli. Exploiting derivative-free local searches in direct-type algorithms for global optimization. *Computational Optimization and Applications*, 65(2):449–475, 2016.

[22] G. Liuzzi, S. Lucidi, and F. Rinaldi. A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM Journal on Optimization*, 2016.

[23] K. Miettinen. *Nonlinear Multiobjective Optimization.* International Series in Operations Research & Management Science. Springer, 1998.

[24] R. Paulavičius, Ya.D. Sergeyev, D.E. Kvasov, and J. Žilinskas. Globally-biased DISIMPL algorithm for expensive global optimization. *Journal of Global Optimization*, 59(2-3):545–567, 2014.

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

[26] Ya.D. Sergeyev and D.E. Kvasov. Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM Journal on Optimization*, 16(3):910–937, 2006.

[27] S. Volpi, M. Diez, N.J. Gaul, H. Song, U. Iemma, K.K. Choi, E.F. Campana, and F. Stern. Development and validation of a dynamic metamodel based on stochastic radial basis functions and uncertainty quantification. *Structural and Multidisciplinary Optimization*, 51(2):347–368, 2015.

[28] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.