DMulti-MADS: Mesh adaptive direct multisearch for blackbox multiobjective optimization *

Jean Bigeon† Sébastien Le Digabel‡ Ludovic Salomon§

Abstract

The context of this research is multiobjective optimization where conflicting objectives are present. In this work, these objectives are only available as the outputs of a blackbox for which no derivative information is available. This work proposes a new extension of the mesh adaptive direct search (MADS) algorithm to constrained multiobjective derivative-free optimization. This method does not aggregate objectives and keeps a list of non dominated points which converges to a (local) Pareto set as long as the algorithm unfolds. As in the single-objective optimization MADS algorithm, this method is built around a search step and a poll step. Under classical direct search assumptions, it is proved that the so-called DMulti-MADS algorithm generates multiple subsequences of iterates which converge to a set of local Pareto stationary points.

Finally, computational experiments suggest that this approach is competitive compared to the state-of-the-art algorithms for multiobjective blackbox optimization.

Key words. Multiobjective optimization, derivative-free optimization, blackbox optimization, mesh adaptive direct search, Clarke analysis.

1 Introduction

This work considers the following multiobjective optimization problem:

$$MOP : \min_{x \in \Omega} f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^\top$$

where $\Omega \subseteq \mathbb{R}^n$ is the *feasible decision set*. The functions $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ for $i = 1, 2, \ldots, m \geq 2$, are the outputs of a blackbox, which means that no analytical form is known. Derivatives are not

*GERAD and Département de mathématiques et génie industriel, Polytechnique Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec, Canada H3C 3A7. This work is partly supported by the NSERC CRD RDCPJ 490744-15 grant and by an InnovÉÉ grant, both in collaboration with Hydro-Québec and Rio Tinto.

†www.g-scop.grenoble-inp.fr/Jean.Bigeon.

‡www.gerad.ca/Sebastien.Le.Digabel.

§www.gerad.ca/Ludovic.Salomon.

available so that gradient-based techniques cannot be considered. Allowing f to take infinity values refers to the possibility that evaluations of f can fail. In these cases, blackbox or derivative-free optimization techniques [5, 19] are particularly relevant. The mapping of Ω by the objective function f is designed as the *feasible objective set*. The sets $\mathbb{R}^n$ and $\mathbb{R}^m$ are denoted the *decision space* and the *objective space*, respectively.

The goal is then to find the best set of trade-off solutions in the objective space, named as the Pareto front, given a finite budget of functions evaluations. These solutions can then be presented to the decision maker, who can decide of the most adequate design dependently for her/his problem [17, 23, 24].

Multiobjective heuristics such as evolutionary/genetic algorithms [22] or particule-swarm optimization [26] are commonly used. However, they do not possess mathematical convergence background and require a significantly large amount of functions evaluations, which is not affordable in this research context where problems involve costly blackbox functions. This last limitation has partly been removed: by using cheaper surrogate models such as radial basis functions [31] or krigging metamodeling [27], one can identify the most promising points to be evaluated with the true objective function.

Among supported convergence-based methods, a first approach is to aggregate all objective functions in one parameterized single-objective formulation. By solving the resulting problem with convergence-proved derivative-free techniques, one is able to get a locally optimal Pareto solution. The procedure can be used again to obtain different Pareto solutions by changing the parameters of the current formulation. The BiMADS [9] and MultiMADS [10] methods follow these approaches. They are both based on the Mesh Adaptive Direct Search (MADS) algorithm [3] for single-objective constrained optimization. BiMADS is only designed for biobjective problems contrary to MultiMADS that takes into account more objectives. Several issues are raised with such scalarization-based methods. A first one is the number of evaluations to allocate to each single-objective problem: too few and no promising points can be found; too many and the algorithm can lack budget to explore potential promising zones in the objective space. A second drawback is the large amount of evaluations that the user can fix to obtain points close to the Pareto front. Function evaluations could rather be used to enrich the non dominated set of points constituting the approximated Pareto front returned by the algorithm.

Recently, new convergence-proved methods for derivative-free multiobjective optimization have emerged, which keep a population of non dominated points that gets closer to the Pareto front as long as the algorithm unfolds [16, 21, 29]. To the best of our knowledge, [21] is the first to propose the DMS framework that extends single-objective direct search algorithms to multiobjective optimization. They prove the existence of at least one subsequence of iterates to a local Pareto point. This approach is used again to build constrained line-search methods for multiobjective optimization [29], implicit-filtering methods for multiobjective optimization [16] and derivative-free trust-region methods for biobjective optimization [33].

Inspired by the works of [21] and [29], this work proposes a new way to extend the MADS algorithm to nonsmooth constrained multiobjective optimization with stronger convergence results than the DMS algorithm. More precisely, under mild assumptions, the DMS algorithm generates at least a sequence of points which converge to a locally Pareto optimal solution. This research goes a step further. It is proved that under the same assumptions, the proposed method generates sequences of points which converge to a set of Pareto stationary points. This result is equally stronger than the one proposed in [29], as their proof requires that their objective functions be

Lipschitz continuous.

This work is organized as follows. Section 2 introduces the notations and definitions relative to multiobjective optimization. Section 3 summarizes the core elements of the MADS algorithm. Section 4 presents the new extension of the MADS algorithm to multiobjective optimization. Section 5 is dedicated to the convergence analysis of the proposed method. Finally, Section 6 reports computational experiments and discussions, followed by the conclusion.

2 Multiobjective optimization and Pareto dominance

In order to compare objective vectors, the following relation order is used [24]:

$$\forall (y^1, y^2) \in (\mathbb{R}^m)^2 \,,\; y^1 \leq y^2 \Leftrightarrow y^2 - y^1 \in \mathbb{R}^m_+ \Leftrightarrow \forall i = 1, 2, \ldots, m,\; y^1_i \leq y^2_i.$$

The relation notations $<$, $>$ and $\geq$ are similarly defined according to the cone $\mathbb{R}^m_+$.
The concept of Pareto dominance can now be introduced:

Definition 2.1. Given two decision vectors x^1 and x^2 in the feasible decision set, it is said that:

- $x^1 \preceq x^2$ (x^1 weakly dominates x^2) if and only $\forall i = 1, 2, \ldots, m,\; f_i(x^1) \leq f_i(x^2)$.
- $x^1 \prec x^2$ (x^1 dominates x^2) if and only $\forall i = 1, 2, \ldots, m,\; f_i(x^1) \leq f_i(x^2)$ and it exists at least an index i_0 such that $f_{i_0}(x^1) < f_{i_0}(x^2)$.
- $x^1 \sim x^2$ (x^1 and x^2 are indifferent) if x^1 does not dominate x^1 nor x^2 does not dominate x^1.

This definition is illustrated in Figure 1 for a biobjective minimization problem in the feasible objective set which is a subset of $\mathbb{R}^2$ delimited by the closed curve. Depending on the x^1 point, three zones in the objective space are considered. The dominance zone is the set of feasible points which dominate x^1. The dominated zone is the set of feasible points which are dominated by x^1. The indifference zone is the set of points which are indifferent to x^1. In this case, $x^4 \prec x^1$, $x^1 \prec x^2$ and $x^3 \sim x^1$.

The above definition enables to define optimality for multiobjective optimization problems.

Definition 2.2 (Global Pareto optimal solution)**.** A point $x^* \in \Omega$ is a global Pareto optimal solution of MOP if there does not exist any other point $x \in \Omega$ such that $x \prec x^*$.

Definition 2.3 (Local Pareto optimal solution)**.** A point $x^* \in \Omega$ is a local Pareto optimal solution of MOP if there does not exist any other point $x \in \Omega \cap \mathcal{N}(x^*)$ such that $x \prec x^*$, where $\mathcal{N}(x^*)$ is a neighbourhood of x^*.

The set of global Pareto optimal solutions in the feasible decision set Ω is called the *Pareto set* denoted by $\mathcal{X}_P$ and its mapping by the objective function f is the *Pareto front* denoted by $\mathcal{Y}_P$. The image of a set of locally Pareto optimal points is called a *local Pareto front*.

The Pareto set is usually composed of many elements [24], which cannot be all enumerated. Solving a multiobjective optimization problem aims at finding a good representative subset of the Pareto front [34]. It is then convenient to introduce the concept of *Pareto approximation set* [39].

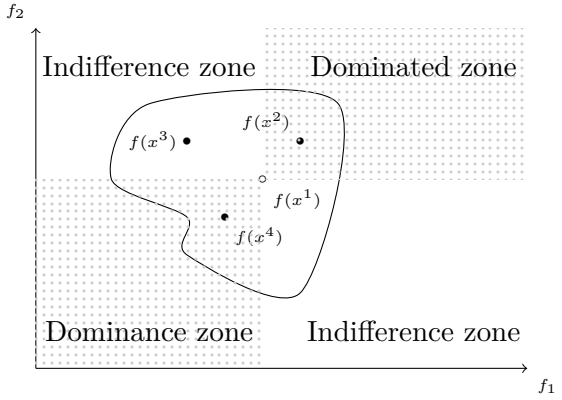


Figure 1: An illustration of the Pareto dominance for a minimization biobjective problem. $x^4 \prec x^1$, $x^1 \prec x^2$, $x^4 \prec x^2$ and $x^3 \sim x^4$.

Definition 2.4 (Pareto set and front approximation)**.** A set of vectors X_N in the feasible decision set Ω is called a Pareto set approximation if no element of this set is dominated by another. Its image in the objective space is called a Pareto front approximation.

All elements of a Pareto set approximation have to be non dominated relatively to each other. A Pareto set approximation should ideally contain elements of the Pareto set. The algorithm described in this work guarantees a convergence towards a Pareto set approximation whose elements are locally Pareto optimal.

Ideally, a Pareto set approximation should contain *extreme points of the Pareto set*. Extreme points of the Pareto set are decision vectors that are the solutions of each single-objective subproblem $\min_{x\in\Omega} f_i(x)$ for $i = 1, 2, \ldots, m$, which are non dominated. With the knowledge of extreme Pareto points, one can get the *ideal objective vector* y^I of MOP, defined by

$$y^I = \left(\min_{x\in\Omega} f_1(x), \min_{x\in\Omega} f_2(x), \ldots, \min_{x\in\Omega} f_m(x) \right).$$

Figure 2 illustrates these concepts.

3 The MADS algorithm

The proposed algorithm is an extension of the MADS algorithm [3] to multiobjective optimization under general constraints. It differs from the existing BiMADS and MultiMADS methods [9, 10], as it does not rely on a scalarization-based approach. This section summarizes the main steps of the MADS algorithm. All the definitions come from [5]. The reader is invited to consult [3] for more details.

The MADS algorithm is a direct search method initially designed to solve optimization single-objective optimization blackbox problems, i.e. $\min_{x\in\Omega} f(x)$, with $f : \mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$ a scalar-valued function and Ω the same subset of $\mathbb{R}^n$ as defined for MOP.

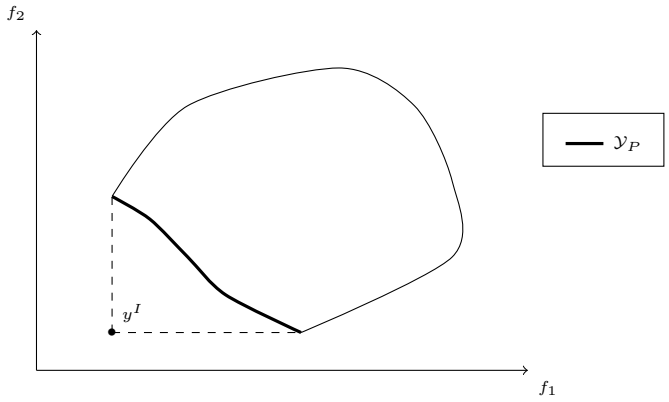


Figure 2: Objective space, Pareto front $\mathcal{Y}_P$ (represented in black) and ideal objective vector y^I for a minimization biobjective problem.

Definition 3.1. Let $G \in \mathbb{R}^{n\times n}$ be a non-singular matrix and $Z \in \mathbb{Z}^{n\times l}$ be such that the columns of Z form a positive spanning set for $\mathbb{R}^n$. Define $D = GZ$. At iteration k, the current *mesh of coarseness* $\delta^k > 0$, generated by D is defined by

$$M^k = \bigcup_{x\in V^k} \left\{x + \delta^k Dz : z \in \mathbb{N}^l\right\}$$

where V^k is the set of points already evaluated by the start of iteration k.

At each iteration k, MADS attempts to find a better point in the decision space, belonging to the current mesh defined above.

V^0 represents the set of starting points indicated by the user. The mesh is generated with a finite set of l directions $D \subset \mathbb{R}^n$ scaled with the mesh size parameter $\delta^k > 0$. Generally, one considers the positive spanning set $D = [I_n, -I_n]$ where I_n is the identity matrix of size n but other choices are possible.

Each iteration is composed of two steps: the search and the poll. The search step enables the user to define its own search strategy as long as new evaluated points remain on the mesh M^k. If a better point is found during the search step, the poll step is not executed. As the convergence analysis depends on the poll step, it is more rigidly defined. It generates points on the mesh around an incumbent solution $x^k \in V^k$. The poll points must belong to the frame of extent Δ^k centered at x^k defined below.

Definition 3.2. Let $G \in \mathbb{R}^{n\times n}$ be a non-singular matrix and $Z \in \mathbb{Z}^{n\times l}$ be such that the columns of Z form a positive spanning set for $\mathbb{R}^n$. Let $\delta^k > 0$ be the mesh size parameter and let Δ^k be such that $\delta^k \leq \Delta^k$. At iteration k, the frame of extent Δ^k generated by D, centered at x^k is defined by

$$F^k = \left\{x \in M^k :\| x - x^k \|_\infty \leq \Delta^k b\right\}$$

with $b = \max\{\| d' \|_\infty : d' \in D\}$ and Δ^k is the frame size parameter such that $\delta^k \leq \Delta^k$.

The initial frame size parameter $\Delta^0 \in \mathbb{R}_+$ can be provided by the user or automatically fixed according to the bound constraints of the optimization problem or the coordinates of an initial starting point [7, 8].

The new candidates in the poll step must belong to the poll set P^k defined by

$$P^k = \left\{x^k + \delta^k d : d \in \mathbb{D}_\Delta^k\right\} \subset F^k$$

where $\mathbb{D}_\Delta^k$ is a positive spanning set of directions. To satisfy these properties, the authors of [3] propose the following relation between δ^k and Δ^k:

$$\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}.$$

Poll points can be evaluated opportunistically (as soon as a better solution is found, the poll step is interrupted) or completely (all candidates are evaluated) as it does not affect the convergence analysis. If a better point is found after the search and poll steps, the iteration is marked as successful. If not, it is considered as unsuccessful. In the first case, the frame size parameter is increased or kept constant. In the second case, the frame size parameter is reduced, increasing the mesh resolution and reducing the exploration field around the current incumbent point x^k. Directions $d \in \mathbb{D}_\Delta^k$ may be generated according to the OrthoMADS instantiation of MADS [1]. Algorithm 1 summarizes the main steps of MADS. More details can be found in [4, 7].

Algorithm 1 The mesh adaptive direct search algorithm (MADS)

Input : Choose a set of initial starting points $V^0 \subset \mathbb{R}^n$, $\Delta^0 > 0$ and $D = GZ$ be a positive spanning set matrix.

for $k = 0, 1, 2, \ldots$ **do**

Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$.

1. **Search step** (optional): Evaluate f at a finite set of points S^k on the mesh M^k. If successful, go to 3.

2. **Poll step** : Select a positive spanning set $\mathbb{D}_\Delta^k \subset D$. Evaluate f at the set of poll points $P^k \subset F^k$ where F^k is the frame of extent Δ^k.

3. **Parameter update**: Update the cache V^{k+1}, the incumbent x^{k+1} and the frame size parameter Δ^k.

end for

Figure 3: A simplified version of the MADS algorithm.

Under mild assumptions, the MADS convergence analysis provided in [3] guarantees the existence of an accumulation point $\hat{x}$ such that its Clarke generalized derivative $f^0(\hat{x}; d)$ is non negative [15] for all the directions $d \in \mathbb{R}^n$ belonging to the hypertangent cone $T_\Omega^H(\hat{x})$ [15]. More details are given in the convergence analysis of the new algorithm in Section 5.

4 The mesh adaptive direct multisearch algorithm (DMulti-MADS) for multiobjective optimization

This section presents the new constrained blackbox algorithm for multiobjective optimization, named DMulti-MADS. It is divided into three subsections. The first subsection gives a high-level description of DMulti-MADS. The two other subsections address specific points: the updating of the mesh size and frame size parameters of the list of non dominated points and the choice of the current incumbent point.

4.1 The DMulti-MADS algorithm

Dmulti-MADS deals with constraints via the extreme barrier approach [21]. Specifically, the objective function f is extended to an extreme barrier function by setting

$$f_\Omega(x) = \begin{cases} f(x) & \text{if } x \in \Omega, \\ [+\infty, +\infty, \ldots, +\infty]^\top & \text{otherwise.} \end{cases}$$

Concretely, all the points that do not satisfy constraint are affected an infinite objective value.

Similarly to the DMS [21] and DFMO [29] algorithms, DMulti-MADS generates a Pareto set approximation at each iteration. More specifically, at each iteration k, DMulti-MADS keeps a finite set L^k which stores all feasible non dominated points found until iteration k, called an *iterate list*. For each k, L^k is a finite set defined as

$$L^k = \left\{(x^j, \Delta^j) : x \in \Omega \text{ and } \Delta^j > 0, j = 1, 2, \ldots, l^k\right\}$$

where $l^k = |L^k|$ and Δ^j is the frame size parameter associated to the j-th non dominated element of the list L^k, x^j. $\delta^j = \min\left\{\Delta^j, \left(\Delta^j\right)^2\right\}$ is the mesh size parameter associated with x^j.

The DMulti-MADS algorithm is an extension of the direct search method MADS. As in single-objective optimization, its functioning is organized around a poll step and a search step, this last one being optional as the convergence analysis does not depend on it. All notations P^k, M^k, F^k have the same mathematical meaning as in single-objective optimization. The algorithm is described in Figure 2.

At the beginning of iteration k, an element $\left(x^k, \Delta^k\right)$ of the list L^k is selected as the current incumbent point at iteration k. The choice of the current incumbent point is crucial in the convergence analysis and will be detailed later on. A temporary list of points L^{add} is initialized to keep track of all the new generated points during iteration k.

As for the MADS algorithm for single-objective optimization, the search step is optional: it aims at improving the performance of the algorithm by evaluating points on the mesh of coarseness δ^k. The poll step obeys to the same rules as in single-objective optimization. To guarantee convergence, evaluated points during the poll step must belong to the poll set P^k.

An iteration is said to be successful as soon as a new point dominating the current incumbent x^k is found. Otherwise, it is said to be unsuccessful. As in single-objective optimization, one can choose the opportunistic or the complete polling strategy.

The two next subsections address main details left open during the description of DMulti-MADS.

Algorithm 2 DMulti-MADS algorithm with extreme barrier

Input : Choose $x^0 \in \Omega$, $\Delta^0 > 0$, $D = GZ$ be a positive spanning set matrix and $\tau \in (0,1) \cap \mathbb{Q}$ the frame size adjustment parameter. Initialize the list of non dominated points $L^0 = \{(x^0, \Delta^0)\}$.

for $k = 0, 1, 2, \ldots$ **do**

Selection of the current incumbent point: Select $\left(x^k, \Delta^k\right)$ element of L^k such that $(x^k, \Delta^k) :=$ `selectCurrentIncumbent`(L^k). Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$.

Initialize $L^{add} := \emptyset$.

Search step (optional): Evaluate f_Ω at a finite set of points S^k on the mesh $M^k = \left\{x^k + \delta^k Dz : z \in \mathbb{N}^p\right\}$. Set $L^{add} := \{(x, \Delta^k) : x \in S^k\}$.

If $t \prec x^k$ for some $t \in S^k$, declare the iteration as successful and skip the poll step.

Poll step : Select a positive spanning set $\mathbb{D}_\Delta^k \subset D$. Evaluate f_Ω at the set of poll points $P^k = \{x^k + \delta^k d : d \in \mathbb{D}_\Delta^k\}$ subset of the frame F^k of extent Δ^k. Set $L^{add} := \{(x, \Delta^k) : x \in P^k\} \cup L^{add}$.

If $t \prec x^k$ for some $t \in P^k$, declare the iteration as successful. Otherwise declare the iteration as unsuccessful.

Parameter update: Remove all dominated points of L^{add}. Call the procedure $L^{k+1} :=$ `updateList`(L^k, L^{add}, τ).

If the iteration is unsuccessful, replace the poll center (x^k, Δ^k) by (x^k, Δ^{k+1}) with $\Delta^{k+1} < \Delta^k$, i.e. $\Delta^{k+1} := \tau\Delta^k$.

end for

Figure 4: Description of the DMulti-MADS algorithm with extreme barrier.

4.2 Updating the list L^k

At the end of iteration k, L^k is updated as described in Algorithm 3. Let emphasize that L^{add} contains only non dominated points relatively to each other.

The `updateList` procedure successively adds new points found during iteration k to the current list L^k and remove dominated points from L^k. At the end of the procedure, the updated L^k list contains only non dominated points.

At iteration k, the DMulti-MADS algorithm attempts to find at least a new point dominating the current incumbent x^k. If found, the iteration is marked as successful. In this case, L^{add} contains at least a point dominating the current incumbent x^k. The first condition of Line 2 of Algorithm 3 ensures the replacing of element (x^k, Δ^k) by the new element (x^j, Δ^j) with $x^j \prec x^k$ and $\Delta^j > \Delta^k$. In case of an unsuccessful iteration, no element of L^{add} dominates the current incumbent x^k. Consequently, the element $(x^k, \Delta^k) \in L^k$ is substituted by (x^k, Δ^{k+1}) with $\Delta^{k+1} < \Delta^k$ as described in Algorithm 2.

Algorithm 3 `updateList`(L^k, L^{add}, τ)

1: **for** $j = 0, 1, 2, \ldots, |L^{add}|$ **do**
2: **if** there exists at least an element $(x, \Delta) \in L^k$ such that $x^j \prec x$ **then**
3: set $L^k := L^k \setminus \left\{(x, \Delta) \in L^k : x^j \prec x\right\} \cup (x^j, \tau^{-1}\Delta^j)$.
4: **else if** there exists $i = 1, 2, \ldots, m$ such that $f_i(x^j) < \min_{x \in L^k} f_i(x)$ **then**
5: set $L^k := L^k \cup (x^j, \tau^{-1}\Delta^j)$.
6: **else if** $x^j \sim x$ for all $(x, \Delta) \in L^k$ **then**
7: set $L^k := L^k \cup (x^j, \Delta^j)$.
8: **end if**
9: **end for**
10: **return** L^k.

Figure 5: Procedure to update the iterate list L^k.

During the search and poll steps, the algorithm can generate points which improve the Pareto set approximation L^k without dominating the current poll center x^k. Typically, a good Pareto set approximation should verify three criteria [2, 38]:

- Its representation in the objective space should be as close as possible from the Pareto front.
- A good (uniform) distribution of the non dominated points in the objective space should be assessed.
- The extent of its representation in the objective space should be maximized, i.e single-objective non dominated solutions should be part of the Pareto set approximation.

By increasing the mesh and frame size parameters for new promising elements, DMulti-MADS enables a larger exploration in the zone around these new points if they are selected as poll centers in the following iterations. The `updateList` procedure considers the following points as promising:

- The ones that dominate a portion of the actual list of non dominated points L^k (Line 2 of Algorithm 3). The images of these points are closer to the Pareto front.
- The ones which improve the extent of the Pareto set approximation in the objective space (Line 4 of Algorithm 3), i.e. that reach a better value for at least one of the objectives.

On the contrary, the `updateList` procedure does not consider as promising new points that fill the approximated Pareto front, i.e new non dominated points that neither dominate the current points or extend the approximated Pareto front (Line 4 of Algorithm 3). The procedure keeps their frame size parameters constant. One may hope to find new non dominated points that locally improve the density of the Pareto front approximation around these new points. Figure 6 illustrates these concepts.

Let emphasize that the convergence analysis requires that new elements added to the list L^k must have a frame size parameter $\Delta \geq \Delta^k$. The `updateList` procedure satisfies these requirements.

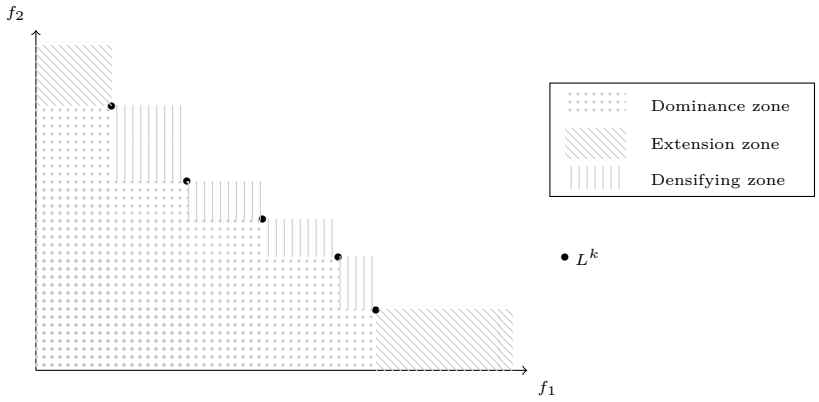


Figure 6: Zone of interests relatively to a set L^k for a biobjective minimization problem in the objective space.

4.3 Choice of the current incumbent x^k

Contrary to the DMS algorithm [21], the choice of the incumbent point x^k at iteration k is less flexible, since the convergence analysis depends on it. More precisely, at iteration k, element $(x^k,\ \Delta^k)$ of the list L^k must verify

$$(x^k, \Delta^k) \in \left\{ (x, \Delta) \in L^k : \Delta = \max_{j=1,2,\ldots,|L^k|} \Delta^j \right\}.$$

As new evaluated points at iteration k are initialized with the Δ^k value, it is possible to have several elements of L^h satisfying the above condition for $h \geq k$. One can ask how to choose the current incumbent point x^k at iteration k among the ones which have maximum frame size parameters.

Following the recommendations of [21], a first approach should be to take the first element of the list L^k with maximum frame size parameter as the current incumbent and add all new non dominated points at the end of the list to diversify the search. To fill gaps into the Pareto front approximation, a second approach consists in choosing elements with the maximum frame size parameter in the least-dense zone of the Pareto front approximation.

Audet et al. [9] consider the distance between three consecutive points in biobjective optimization; the point in the middle is taken as the current incumbent. The crowding distance [22] extends this result to more objectives. For each objective, values of the non dominated points are sorted in ascendant order. The crowding distance for a given point is the sum of the normalized distance between this point and its two adjacent neighbors according to each objective. Its computational cost is in $\mathcal{O}(m \times |L^k| \times \log(|L^k|))$. But the crowding distance does not consider the extreme points of the current approximated Pareto front. Based on these remarks and the work of [9], a new way to select a current incumbent point x^k with frame size parameter Δ^k is proposed, as described in Algorithm 4.

The `selectCurrentIncumbent` procedure firstly stores elements of the list L^k with maximum frame size parameters into a temporary list $L^{\max}$. If L^k possesses two elements with the same

Algorithm 4 `selectCurrentIncumbent`(L^k)

Let $L^{\max} := \left\{ (x, \Delta) \in L^k : \Delta = \max_{(x', \Delta') \in L^k} \Delta' \right\}$
if $|L^{\max}| = 1$ **then**
  return (x, Δ) with $L^{\max} = \{(x, \Delta)\}$.
else if $|L^{\max}| = 2$ and $|L^k| = 2$ **then**
  Let $j_0 \in \arg\max_{j=1,2} \max_{i=1,2,\ldots,m} f_i(x^j)$.
  return (x^{j_0}, Δ^{j_0}).
else
  Let $j_0 \in \arg\max_{j=1,2,\ldots,|L^{\max}|} \max_{i=1,2,\ldots,m} \gamma_i(x^j)$.
  return (x^{j_0}, Δ^{j_0}).
end if

Figure 7: A procedure to select the current incumbent at iteration k taking into account the spacing between elements of the iterate list L^k in the objective space.

maximal frame size parameter value, the procedure selects as current incumbent the one with the higher objective value among all objectives. By exploring the region around this incumbent, one can expect to find non dominated points with lowest objective values and then close to the Pareto front. If $|L^{\max}| \geq 2$ and L^k has more than two elements, the element of the list $L^{\max}$ in the least dense zone of the current Pareto set approximation L^k in the objective space is selected, according to our new distance-based indicator γ_i for $i = 1, 2, \ldots, m$.

For each objective $i = 1, 2, \ldots, m$, $L^k = \{(x^1, \Delta^1), (x^2, \Delta^2), \ldots, (x^{|L^k|}, \Delta^{|L^k|})\}$ is ordered such that

$$f_i(x^1) \leq f_i(x^2) \leq \ldots \leq f_i(x^{|L^k|}).$$

γ_i corresponds to the scaled distance between three consecutive points according to objective i for $i = 1, 2, \ldots, m$. It is then defined, for $j = 1, 2, \ldots, |L^k|$, by

$$\gamma_i(x^j) = \begin{cases} 2 \, \dfrac{f_i(x^2) - f_i(x^1)}{f_i(x^{|L^k|}) - f_i(x^1)} & \text{if } j = 1, \\ 2 \, \dfrac{f_i(x^{|L^k|}) - f_i(x^{|L^k|-1})}{f_i(x^{|L^k|}) - f_i(x^1)} & \text{if } j = |L^k|, \\ \dfrac{f_i(x^{j+1}) - f_i(x^{j-1})}{f_i(x^{|L^k|}) - f_i(x^1)} & \text{otherwise.} \end{cases}$$

If x^j is the first or last element of the sorted list L^k, the double scaled distance between this point and its closest neighbor for objective i is considered.

The point which is chosen as the current incumbent at iteration k is the one with the maximal

frame size parameter in the least dense zone according to $\gamma = \max_{i=1,2,\ldots,m} \gamma_i$. Figure 8 illustrates this distance-based indicator for two objectives.

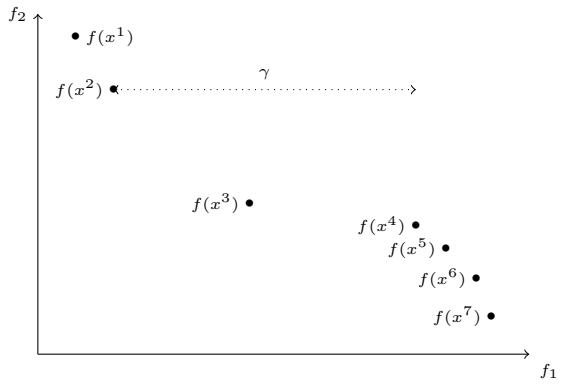


Figure 8: An example of the γ distance-based indicator in biobjective optimization. γ corresponds to the largest scaled distance between three consecutive points according to one objective i for $i = 1, 2, \ldots, m$.

5 Convergence analysis of the DMulti-MADS algorithm

The previous section describes the DMulti-MADS algorithm. This section is devoted to its convergence analysis, inspired by the works of [21, 29]. Basically, the following result is shown: under mild assumptions, DMulti-MADS produces (at the limit) a set of locally stationary points of the constrained multiobjective problem. As in the classical analysis of MADS [3], the objective is to show there exists a subsequence of mesh size and frame size parameters converging to zero. However, contrary to [21], the analysis distinguishes between the Pareto set approximation and its elements. To guarantee this condition, the following assumption is required given a feasible starting point x^0.

Assumption 5.1. *The level set $L(x^0) = \bigcup_{i=1}^{m} \{x \in \Omega : f_i(x) \leq f_i(x^0)\}$ is compact. Each component f_i of the objective function f is bounded from above and below for $i = 1, 2, \ldots, m$.*

5.1 Preliminaries

This subsection is dedicated to the analysis of the convergence of the mesh and frame size parameters, based on the works of [3, 21, 37].

Theorem 5.1. *Let Assumption 5.1 hold. Then DMulti-MADS generates a sequence of Pareto set approximations iterates satisfying*

$$\liminf_{k \to \infty} \delta^k = 0 \text{ and } \liminf_{k \to \infty} \Delta^k = 0.$$

Proof. Suppose by contradiction that there exists a strictly positive lower bound on the mesh size parameter δ^k for all $k \geq 0$. As in single-objective optimization [3], all the points evaluated by DMulti-MADS lie on the mesh. Since the intersection of the mesh and a compact set, as assumed by Assumption 5.1, is a finite set, then only a finite number of points can be added to the Pareto set approximation. There remains to prove that the DMulti-MADS algorithm cannot cycle among these points.

When an element of the Pareto set approximation is removed, it is because it is dominated by a new point. By transitivity, it cannot be added to the Pareto set approximation again. At each successful iteration, at least one non-dominated point is added to the current Pareto set approximation. At each unsuccessful iteration, at most one non-dominated point is added to the current Pareto set approximation. As the number of points that can be added is finite, the number of times the maximal mesh size parameter of the approximated Pareto set can be increased or kept constant is finite. With the rule update, this contradicts the existence of strictly positive lower bound on the mesh size parameter.

The second part of the theorem is based on the fact that $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$, which concludes the proof. □

Let highlight that when the iteration k is marked as unsuccessful, the maximal mesh size parameter of the current Pareto set approximation is not always decreased. Indeed, new non dominated points can be added to the Pareto set approximation even though the algorithm does not find new points dominating the current incumbent x^k. Furthermore, the Pareto set approximation can contain elements with the same mesh size parameter.

The following corollary puts into relief a statistical relation between the elements of the list L^k. For a given subsequence of Pareto set approximations, the mesh and frame size parameters converge in average towards zero.

Corollary 5.1.1. *Let Assumption 5.1 hold. Let* $\{L^k\}_{k\in\mathbb{N}}$ *with*

$$L^k = \left\{(x^{j_k}, \Delta^{j_k}) : x^{j_k} \in \Omega \text{ and } \Delta^{j_k} > 0,\ j = 1, 2, \ldots, |L^k|\right\}$$

be the sequence of current Pareto set approximations generated by the DMulti-MADS algorithm. Then

$$\lim_{k\to\infty} \inf \bar{\delta}^k = 0 \text{ and } \lim_{k\to\infty} \inf \bar{\Delta}^k = 0$$

with

$$\bar{\delta}^k = \frac{1}{|L^k|}\sum_{j=1}^{|L^k|} \delta^{j_k} \text{ and } \bar{\Delta}^k = \frac{1}{|L^k|}\sum_{j=1}^{|L^k|} \Delta^{j_k}.$$

Proof. Since $\delta^k = \max_{j=1,2,\ldots,|L^k|} \delta^{j_k}$ for $k = 0, 1, 2, \ldots$, one has $\bar{\delta}^k \leq \delta^k$ for all $k \in \mathbb{N}$. Theorem 5.1 states that there exists a subset of indexes $k \in K$ such that $\{\delta^k\}_{k\in K}$ converges to zero. By the squeeze theorem, one gets $\lim_{k\in K} \bar{\delta}^k = \lim_{k\in K} \delta^k = 0$. The proof with the frame size parameter is equivalent. □

Intuitively, this corollary claims that either the cardinality of L^k converges to infinity or all the mesh and frame size parameters converge to zero for a given set of indexes $k \in K$.

Theorem 5.1 analyzes the convergence of mesh size parameters relatively to the list of non dominated points L^k. One has to go deeper to analyze the behavior of mesh size and frame size parameters of specific elements of the list to prove convergence of the DMulti-MADS algorithm to stationary points. To do that, the concept of linked sequence, taken from [29] is introduced.

Definition 5.1. Let $\{L^k\}_{k\in\mathbb{N}}$ with $L^k = \left\{(x^j, \Delta^j) : x \in \Omega \text{ and } \Delta^j > 0, j = 1, 2, \ldots, |L^k|\right\}$ be the sequence of current Pareto set approximations generated by the DMulti-MADS algorithm. A linked sequence is defined as a sequence $\{(x^{j_k}, \Delta^{j_k})\}$ such that for any $k = 1, 2, \ldots$, the pair $(x^{j_k}, \Delta^{j_k}) \in L^k$ is generated at iteration $k-1$ of DMulti-MADS from the pair $(x^{j_{k-1}}, \Delta^{j_{k-1}}) \in L^{k-1}$.

The relation between the pair $(x^{j_k}, \Delta^{j_k}) \in L^k$ and $(x^{j_{k-1}}, \Delta^{j_{k-1}}) \in L^{k-1}$ is precised below.

1. Successful iteration : the algorithm generates at least one point that dominates the current incumbent point x^{k-1}. All non dominated points at iteration $k-1$ which are not dominated at the end of iteration $k-1$ are inserted in the Pareto set approximation L^k.

   Then one has:

   - $\forall (x^{j_k}, \Delta^{j_k}) \in L^k \setminus L^{k-1}$,

     $$x^{j_k} = x^{k-1} + \delta^{k-1} D z^{k-1} \text{ for some } z^k \in \mathbb{N}^p \text{ and } \Delta^{jk} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}.$$

   - $\forall (x^{j_k}, \Delta^{j_k}) \in L^k \cap L^{k-1}$,

     $$x^{j_k} = x^{j_{k-1}} \text{ and } \Delta^{j_k} = \Delta^{j_{k-1}}.$$

2. Unsuccessful iteration : The algorithm does not generate a point that dominates the current incumbent point x^{k-1}. However, it is possible that the algorithm finds new non dominated points which are inserted into the Pareto set approximation.

   Then one has:

   - $\forall (x^{j_k}, \Delta^{j_k}) \in L^k \setminus L^{k-1}$,

     $$x^{j_k} = x^{k-1} + \delta^{k-1} D z^{k-1} \text{ for some } z^k \in \mathbb{N}^p \text{ and } \Delta^{jk} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}.$$

   - $\forall (x^{j_k}, \Delta^{j_k}) \in L^k \cap L^{k-1} \setminus \{(x^{k-1}, \Delta^{k-1})\}$,

     $$x^{j_k} = x^{j_{k-1}} \text{ and } \Delta^{j_k} = \Delta^{j_{k-1}}.$$

   - $\forall (x^{j_k}, \Delta^{j_k}) \in \{(x^{k-1}, \Delta^{k-1})\}$,

     $$x^{j_k} = x^{j_{k-1}} \text{ and } \Delta^{j_k} = \tau\Delta^{j_{k-1}}.$$

     Note that the current incumbent at iteration $k-1$ is not always the same at iteration k.

As proved below, linked sequences generate subsequence of points whose mesh size and frame size parameters converge to 0.

Theorem 5.2. *Let Assumption 5.1 hold. Let* $\{L^k\}_{k\in\mathbb{N}}$ *with*

$$L^k = \left\{(x^j, \Delta^j) : x^j \in \Omega \text{ and } \Delta^j > 0,\ j = 1, 2, \ldots, |L^k|\right\}$$

be the sequence of current approximated Pareto sets generated by the DMulti-MADS algorithm. Then every linked sequence $\{(x^{j_k}, \Delta^{j_k})\}$ *is such that*

$$\liminf_{k\to\infty} \delta^{j_k} = 0 \text{ and } \liminf_{k\to\infty} \Delta^{j_k} = 0.$$

Proof. $\forall k \in \mathbb{N}, 0 \leq \delta^{j_k} \leq \delta^k$. Using Theorem 5.1 and the squeeze theorem, one gets

$$\liminf_{k\to\infty} \delta^{j_k} = \liminf_{k\to\infty} \delta^k = 0.$$

As $\delta^{j_k} = \min\left\{\Delta^{j_k}, \left(\Delta^{j_k}\right)^2\right\}$, it results that

$$\liminf_{k\to\infty} \Delta^{j_k} = 0.$$

□

Note that due to the update strategy, the mesh and frame size parameters for a linked sequence $\{(x^{j_k}, \Delta^{j_k})\}$ can only decrease when there exists an index $k \in \mathbb{N}$ such that $x^k = x^{j_k}$ and iteration k is marked as unsuccessful.

5.2 Refining subsequences and directions

The theory of classical direct search methods consists in analyzing the behavior at limit points of unsuccessful iterates. The concept of refining subsequences in the context of multiobjective optimization, previously introduced in the works of [3], is adapted.

Definition 5.2. A subsequence $\{x_k\}_{k\in K}$ of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if $\{\delta^k\}_{k\in K}$ converges to 0. The limit point $\hat{x}$ of a refining subsequence $\{x_k\}_{k\in K}$ is said to be a refining point.

By Assumption 5.1 and Theorem 5.2, every linked sequence produced by the DMulti-MADS algorithm contains a refining subsequence.

Theorem 5.3. *Let Assumption 5.1 hold. Let* $\{L^k\}_{k\in\mathbb{N}}$ *with*

$$L^k = \left\{(x^j, \Delta^j) : x^j \in \Omega \text{ and } \Delta^j > 0,\ j = 1, 2, \ldots, |L^k|\right\}$$

be the sequence of current Pareto set approximations generated by the DMulti-MADS algorithm. Then every linked sequence $\{(x^{j_k}, \Delta^{j_k})\}$ *is such that* $\{x^{j_k}\}_{k\in K}$ *is a refining subsequence.*

This theorem is stronger than the one proposed by [21] where the DMS algorithm generates at least one refining subsequence.

As the DMulti-MADS convergence analysis is based on the study of generalized directional derivatives along certain limits directions at refined points, the concept of a refining direction [3] is introduced.

Definition 5.3. Given a refining subsequence $\{x^{j_k}\}_{k\in K}$ and its corresponding refining point $\hat{x}$, a direction d is said to be a refining direction if and only if there exists an infinite subset $K' \subseteq K$ such that $d^k \in \mathbb{D}_\Delta^k$ with $x^{j_k} + \delta^{j_k} d^k \in \Omega$ and $\lim_{k\in K'} \frac{d^k}{\|d^k\|} = \frac{d}{\|d\|}$.

5.3 Tangent cones and generalized derivatives

The main convergence result of DMulti-MADS is that a limit point of a refining subsequence of a linked sequence generated by the algorithm is Pareto-Clarke stationary. It requires some concepts linked to stationarity in the context of nonsmooth constrained multiobjective optimization.

Classical theory of direct-search methods in the context of constrained single-objective optimization makes use of the hypertangent cone, which is a generalization of the tangent cone at x, i.e the set of directions that point inside Ω . Definition and notations are taken from [3].

Definition 5.4. A vector $d \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in the closure of Ω if for every sequence $\{y^k\}$ of elements of Ω that converges to x and for every sequence of positive real numbers $\{t^k\}$ converging to zero, there exists a sequence of vectors $\{w^k\}$ converging to d such that $y^k + t^k w^k \in \Omega$.

The set of all Clarke tangent vectors to Ω to x is called the Clarke tangent cone at x and is denoted by $T_\Omega^{Cl}(x)$. The interior of this cone is designed as the hypertangent cone.

Definition 5.5. A vector $d \in \mathbb{R}^n$ is said to be be a hypertangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point $x \in \Omega$ if and only if there exists a scalar $\epsilon > 0$ such that

$$y + tw \in \Omega, \ \forall y \in \Omega \cap B_\epsilon(x), \ w \in B_\epsilon(x) \text{ and } 0 < t < \epsilon$$

where $B_\epsilon(x)$ is the open ball centered at x of radius ϵ.

The set of all hypertangent directions vectors to Ω at x is called the hypertangent cone to Ω at x, and is denoted by $T_\Omega^H(x)$.

Note that the Clarke tangent cone can be considered as the closure of the hypertangent cone. The convergence analysis requires the assumption that the objective function f is locally Lipschitz continuous in Ω, i.e. each of its components f_i, for $i = 1, 2, \ldots, m$, is locally Lipschitz continuous in Ω. Assuming this assumption is satisfied, the Clarke-Jahn generalized derivatives [15] of each function f_i along directions d in the hypertangent cone to Ω at x exist and are defined by

$$f_i^o(x; d) = \limsup_{\substack{y \to x, \ y \in \Omega \\ t \searrow 0, \ y + td \in \Omega}} \frac{f_i(y + td) - f_i(y)}{t}, \ i = 1, 2, \ldots, m.$$

Audet et al [3] show that the directions v in the Clarke tangent cone can be expressed by taking the limit, i.e.

$$f_i^o(x; v) = \lim_{\substack{d \in T_\Omega^H(x) \\ d \to v}} f_i^o(x; d), \ i = 1, 2, \ldots, m.$$

Stationarity conditions for the DMulti-MADS algorithm can now be defined.

Definition 5.6. Let f be Lispchitz continuous near a point $\hat{x} \in \Omega$. $\hat{x}$ is said to be a Pareto-Clarke critical point of f in Ω for all directions $d \in T_\Omega^{Cl}(\hat{x})$ if there exists $i = i(d) \in \{1, 2, \ldots, m\}$ such that $f_i^o(\hat{x}; d) \geq 0$.

If the objective function f is differentiable, this definition can be reformulated using the gradient of each component of the objective function f.

Definition 5.7. Let f be differentiable at a point $\hat{x} \in \Omega$. $\hat{x}$ is said to be a Pareto-Clarke KKT critical point of f in Ω for all directions $d \in T_\Omega^H(\hat{x})$ if there exists $i = i(d) \in \{1, 2, \ldots, m\}$ such that $\nabla f_i(\hat{x})^\top d \geq 0$.

5.4 Convergence results

The main convergence results of the DMulti-MADS algorithm can now be given. It states that every limit point of a refining subsequence generated by DMulti-MADS is Pareto-Clarke optimal under the condition that the set of refining directions is dense in the unit sphere. Thus, every limit point of a refining subsequence of a linked sequence generated by DMulti-MADS is Pareto-Clarke optimal. The proof follows the classical theory of direct-search methods and in particular of the DMS algorithm.

Theorem 5.4. *Let $\{x^{j_k}\}_{k\in K}$ be a refining subsequence converging to $\hat{x} \in \Omega$ and a refining direction $d \in T_\Omega^H$ for $\hat{x}$. Assume that f is Lipschitz continuous near $\hat{x}$. Then there exists $i = i(d) \in \{1, 2, \ldots, m\}$ such that $f^o_{i(d)}(\hat{x}; d) \geq 0$.*

Proof. Let $\{x^{j_k}\}_{k\in K}$ be a refining subsequence converging to a refined point $\hat{x} \in \Omega$ and $d \in T_\Omega^H(\hat{x})$ a refining direction for $\hat{x}$. By definition of a refining direction, there exists an infinite subsequence K' of the set of indices K of unsuccessful iterations, with poll directions $d^k \in \mathbb{D}^k_\Delta$ such that $x^{j_k} + \delta^k d^k \in \Omega$ and $\lim_{k\in K'} \frac{d^k}{\|d^k\|} = \frac{d}{\|d\|}$ for all $k \in K'$. Let ν be the Lipschitz constant of f near $\hat{x}$.

For $i \in \{1, 2, \ldots, m\}$, one has

$$
\begin{aligned}
f_i^o\left(\hat{x}; \frac{d}{\|d\|}\right) &= f_i^o\left(\hat{x}; \frac{d}{\|d\|}\right) + \limsup_{k\in K'} \frac{\nu\delta^k\|d^k\| \left\| \frac{d^k}{\|d^k\|} - \frac{d}{\|d\|} \right\|}{\delta^k\|d^k\|} \\
&\geq f_i^o\left(\hat{x}; \frac{d}{\|d\|}\right) + \limsup_{k\in K'} \frac{\left| f_i(x^{j_k} + \delta^k d^k) - f_i(x^{j_k} + \delta^k\|d^k\| \frac{d}{\|d\|}) \right|}{\delta^k\|d^k\|} \\
&\geq \limsup_{k\in K'} \frac{f_i(x^{j_k} + \delta^k\|d^k\| \frac{d}{\|d\|}) - f_i(x^{j_k})}{\delta^k\|d^k\|} + \limsup_{k\in K'} \frac{\left| f_i(x^{j_k} + \delta^k d^k) - f_i(x^{j_k} + \delta^k\|d^k\| \frac{d}{\|d\|}) \right|}{\delta^k\|d^k\|} \\
&\geq \limsup_{k\in K'} \frac{f_i(x^{j_k} + \delta^k d^k) - f_i(x^{j_k} + \delta^k\|d^k\| \frac{d}{\|d\|}) + f_i(x^{j_k} + \delta^k\|d^k\| \frac{d}{\|d\|}) - f_i(x^{j_k})}{\delta^k\|d^k\|} \\
&= \limsup_{k\in K'} \frac{f_i(x^{j_k} + \delta^k d^k) - f_i(x^{j_k})}{\delta^k\|d^k\|}.
\end{aligned}
$$

As $\{x^{j_k}\}_{k\in K}$ is a refining subsequence, each $k \in K' \subseteq K$ corresponds to an unsuccessful iteration, and $x^{j_k} + \delta^k d^k \in \Omega$ for $d^k \in \mathbb{D}^k_\Delta$ does not dominate x^{j_k}. One can find for $k \in K'$ a component of the objective function of index $i(k)$ such that $f_{i(k)}(x^{j_k} + \delta^k d^k) - f_{i(k)}(x^k) \geq 0$. As the objective function has a finite number of components, one can consider a subset of iteration indexes $K'' \subset K'$ such that there exists at least one index $i = i(d)$ such that:

$$
f_i^o\left(\hat{x}; \frac{d}{\|d\|}\right) \geq \limsup_{k\in K''} \frac{f_i(x^{j_k} + \delta^k d^k) - f_i(x^{j_k})}{\delta^k\|d^k\|} \geq 0.
$$

□

If f is strictly differentiable at a refining point, one can state the following corollary.

Corollary 5.4.1. *Let $\{x^{j_k}\}_{k\in K}$ be a refining subsequence converging to $\hat{x} \in \Omega$ and a refining direction $d \in T_\Omega^H$ for $\hat{x}$. Assume that f is strictly differentiable at $\hat{x}$. Then there exists an $i \in \{1, 2, \ldots, m\}$ such that $\nabla f_i(\hat{x})^\top d \geq 0$.*

Proof. It comes from the fact that when f is strictly differentiable at a point $x \in \Omega$, $f_i^o(x; d) = \nabla f_i(x)^\top d$ for $d \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$ (see [15]). □

Assuming that the set of refining directions is dense in the hypertangent cone at a refining point $\hat{x}$, one can state the following theorem, which complies with the DMS algorithm convergence analysis.

Theorem 5.5. *Let $\{x^{j_k}\}_{k\in K}$ be a refining subsequence converging to $\hat{x} \in \Omega$. Assume that f is Lipschitz continuous near $\hat{x}$ and $T_\Omega^H(\hat{x}) \neq \emptyset$. If the set of refining directions is dense for $\hat{x}$ in $T_\Omega^{Cl}(\hat{x})$, then $\hat{x}$ is a Pareto-Clarke critical point. In addition, if f is strictly differentiable at $\hat{x}$, then $\hat{x}$ is a Pareto-Clarke KKT critical point.*

Proof. As proved in [3], given a direction v in the Clarke tangent cone, one has

$$f_i^o(\hat{x}; v) = \lim_{\substack{d\in T_\Omega^H(\hat{x}) \\ d\to v}} f_i^o(\hat{x}; d), i = 1, 2 \ldots, m.$$

As the set of refining directions is dense for $\hat{x}$ in $T_\Omega^{Cl}(\hat{x})$, there exists a sequence of refining directions $\{d_r\}_{r\in R} \in T_\Omega^H(\hat{x})$ for $\hat{x}$ such that $\lim_{r\in R} d_r = v$. Since the number of components of the objective function is finite, considering a subset $R' \subseteq R$ of indexes, one gets $v = \lim_{r\in R'} d_r$ with $f_{i(v)}^o(\hat{x}; d_r) \geq 0$ for all $r \in R'$ by Theorem 5.4. Passing at the limit concludes the proof. The second statement of the theorem can be deduced easily. □

6 Computational experiments

This section is devoted to the numerical experiments of DMulti-MADS on bound-constrained multiobjective problems taken from [21]. The first part introduces the considered test problems and solvers. The second part presents an extension of the classical data profiles [30] for multiobjective blackbox optimization, based on the hypervolume indicator [38]. Several variants of DMulti-MADS are then compared using this tool in a third part. In the last part, the performance of DMulti-MADS is analysed versus other state-of-the-art solvers. The DMulti-MADS code is freely available at `https://github.com/bbopt/DMultiMadsEB`.

6.1 Bound-constrained problems and algorithms tested

All algorithms are tested on the benchmark set of multiobjective optimization problems taken from [21]. It is composed of 100 problems, with a number of variables $n \in [\![1, 30]\!]$ and a number of objective functions $m \in \{2, 3, 4\}$. It has 69 problems with $m = 2$, 29 problems with $m = 3$, and 2 problems with $m = 4$. A modeling of these problems in `AMPL` can be found at `www.mat.uc.pt/dms/`. Their implementations coded in `Matlab` and `C++` can be found at `https://github.com/bbopt/DMultiMadsEB`. In the numerical experiments, the following solvers are tested:

- BiMADS (Bi-objective Mesh Adaptive Direct Search) [9] tested only for $m = 2$ objectives – `www.gerad.ca/nomad/`.
- DMS (Direct MultiSearch) [21], version `0.3.` – `www.mat.uc.pt/dms/`
- MOIF (MultiObjective Implicit Filtering) [16], version `0.1` – `www.iasi.cnr.it/~liuzzi/DFL/`.
- NSGA-II (Non Dominated Sorting Algorithm II) [22]; implemented in the `Pymoo` Library [12], version `0.3.2` – `pymoo.org/`.

BiMADS, DMS and MOIF are deterministic algorithms, whereas NSGA-II is a stochastic solver. All numerical results can be found at `https://github.com/bbopt/DMultiMadsEB`.

6.2 Data profiles for multiobjective blackbox optimization

In single-objective optimization, data profiles [30] enable the user to assess the performance of a method on a set of problems for a given budget of function evaluations. Assume one wants to solve $\min_{x\in\Omega} f(x)$ where f is a single-objective function and Ω the set of constraints. Let $\mathcal{P}$ be the set of problems and $\mathcal{A}$ the set of considered algorithms. A data profile associated to a solver $a \in \mathcal{A}$ is a cumulative distribution function which returns the percentage of problems in $\mathcal{P}$ solved by $a \in \mathcal{A}$ for a given budget of group of function evaluations $k \in \mathbb{N}$, i.e.

$$d_a(k) = \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} : N_{a,p} \leq k(n_p + 1)\}| \tag{1}$$

where $N_{a,p}$ is the number of functions evaluations required by solver $a \in \mathcal{A}$ to solve the problem $p \in \mathcal{P}$ and n_p the dimension of the problem $p \in \mathcal{P}$. By convention, if a problem has not been solved given a maximum budget of function evaluations, then $N_{a,p} = +\infty$. The $n_p + 1$ term on the right part of the inequality in Equation (1) is added based on the assumption that a problem with higher dimension requires more function evaluations to be solved than a problem with lower dimension. $n_p + 1$ is equally the number of points needed to construct a simplex gradient in $\mathbb{R}^{n_p}$.

The definition of a convergence test to claim a problem has been solved is a critical phase in the construction of a data profile. For single-objective optimization, let x^b be the best feasible point found by all algorithms on a given problem, and x^e be the best feasible point found by a given algorithm on this problem after e evaluations. Then the problem is said to be solved by this algorithm with accuracy $\varepsilon_\tau > 0$ if

$$f(x^0) - f(x^e) \geq (1 - \varepsilon_\tau)(f(x^0) - f(x^b)),$$

where x^0 is the feasible initial starting point.

Several works describe the construction of data profiles for multiobjective blackbox optimization, based on the use of quality indicators (see [2, 28] for surveys on quality indicators). Since the works of [21], which to the best of our knowledge, introduced data and performance profiles for multiobjective blackbox optimization, many researchers have adopted this framework to assess the performance of their methods [16, 29, 32]. However, these frameworks rely on spread and cardinality metrics, which are not Pareto compliant [2] with the dominance order for multiobjective optimization. The use of Pareto compliant quality indicators such as the hypervolume indicator is

addressed in [29] for the construction of performance profiles [30] for multiobjective blackbox optimization. Nonetheless, performance profiles possess the following drawbacks: they are sensitive to the number of considered solvers [21] and are more difficult to interpret than the data profiles [14].

In this work, a new extension of data profiles for multiobjective optimization is proposed, which relies on the hypervolume indicator. Note that the use of the hypervolume indicator in data profiles is not new, as it is done in [13]. Some similarities between this work and [13] are present. The main differences are highlighted:

- This work is more detailed in the description of the integration of the hypervolume indicator into the convergence criterion, specifically the scaling of the objective vectors and the positioning of the reference point.
- Variability of stochastic algorithms is included in this work based on the research of [35, 36].

The hypervolume indicator [38] represents the volume of the space in the objective space dominated by a Pareto front approximation Y_N and delimited above by an objective vector $r \in \mathbb{R}^m$ such that $\forall y \in Y_N, y < r$. An illustration of the hypervolume indicator is shown in Figure 9.

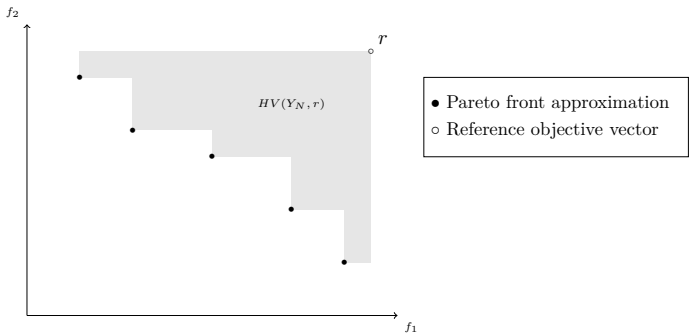


Figure 9: Illustration of the hypervolume indicator (HV) for a biobjective minimization problem, delimited above by the reference objective vector $r \in \mathbb{R}^2$. The higher, the better.

The hypervolume indicator enjoys many properties: it is Pareto compliant with the dominance ordering, its interpretation is simple and it can serve as a metric for convergence, cardinality, spread and extension of a Pareto front approximation [2, 28]. On the contrary, its computation is exponential in the number of objectives [2, 28]. Practically, there exists several libraries (see for example [25]) which can compute the hypervolume indicator value in less than some milliseconds on modern machines for a small number of objectives ($m \in \{1, 2, 3, 4\}$). Indeed, one has to keep in mind that the construction time of the data profiles is not important in these experiments.

To build the convergence test for the multiobjective optimization case, one needs to consider a Pareto front reference Y^p for the problem $p \in \mathcal{P}$. From this Pareto front reference, let extract the approximated ideal objective vector

$$\tilde{y}^I = \left(\min_{y \in Y^p} y_1, \min_{y \in Y^p} y_2, \ldots, \min_{y \in Y^p} y_{m_p} \right),$$

the approximated Nadir objective vector

$$\tilde{y}^N = \left(\max_{y \in Y^p} y_1, \max_{y \in Y^p} y_2, \ldots, \max_{y \in Y^p} y_{m_p} \right),$$

and the reference objective vector

$$r_p = 1.1 \times \tilde{y}^N$$

where m_p is the number of objectives considered in problem $p \in \mathcal{P}$.

Let Y^e be the Pareto front approximation found after e evaluations by a given deterministic algorithm. To avoid privileging an objective function against another, a transformation T is applied to the Pareto front reference, the Pareto front approximation and the reference objective vector. This transformation is defined by: $\forall y \in Y^e \cup Y^p \cup \{r_p\}$,

$$T(y) = \begin{cases} (y - \tilde{y}^I) \oslash (\tilde{y}^N - \tilde{y}^I) & \text{if } \tilde{y}^I \neq \tilde{y}^N \\ y - \tilde{y}^I & \text{otherwise} \end{cases}$$

where $\oslash$ is the element-wise divisor operator. Thus all objective vectors are scaled and translated such that $T(\tilde{y}^I) = 0_{\mathbb{R}^{m_p}}$ and $T(\tilde{y}^N) = 1_{\mathbb{R}^{m_p}}$ if $\tilde{y}^N$ exists. $T(r_p)$ is the objective vector composed of 1.1 values in $\mathbb{R}^{m_p}$. Note that this translation does not modify the dominance ordering, i.e. $y^1 \leq y^2$ implies $T(y^1) \leq T(y^2)$ with y^1, y^2 two objective vectors. Finally, the multiobjective problem $p \in \mathcal{P}$ is said to be solved by this algorithm with accuracy $\varepsilon_\tau > 0$ if

$$\frac{HV\left(T(Y^e), T(r_p)\right)}{HV\left(T(Y^p), T(r_p)\right)} \geq 1 - \varepsilon_\tau$$

where $HV(Y_N, r)$ is the hypervolume indicator value of the Pareto front approximation delimited from above by the reference objective vector r. By convention, the hypervolume indicator does not consider elements of Y_N which do not dominate r. If all elements of Y_N do not dominate r, then $HV(Y_N, r) = 0$.

Given a problem $p \in \mathcal{P}$, the Pareto front reference Y^p is constructed using the best feasible non dominated points found by all considered solvers on this problem for a maximum budget of evaluations. More precisely, the Pareto front reference is computed by removing the dominated points found in the union of the Pareto front approximations generated by the set of solvers on Problem $p \in \mathcal{P}$ once the budget of functions evaluations is exhausted.

Stochastic algorithms are commonly used to tackle blackbox multiobjective optimization problems. To include them into the data profiles framework, one can consider that different instances of a given problem obtained by different random seeds constitute different problems. This augmented set of problems can be used to construct classical data profiles as explained above. The authors in [14] adopt this approach. However, this approach does not enable to visualize the variability of stochastic algorithms on a given set of problems $\mathcal{P}$. This work proposes another approach, inspired by [35, 36].

Consider a stochastic algorithm $a \in \mathcal{A}$ and a set of problems $\mathcal{P}$. Assume the algorithm has generated after e evaluations different Pareto front approximations $Y^e_{a_1,p}, Y^e_{a_2,p}, \ldots, Y^e_{a_q,p}$ corresponding to $q = |\mathcal{I}_a|$ different instances for all the problems $p \in \mathcal{P}$. For each instance, compute the respective hypervolume values $h^e_{a_1,p}, h^e_{a_2,p}, \ldots, h^e_{a_q,p}$ as described previously. A specific launch

$a_j \in \mathcal{I}_a$ of the stochastic algorithm $a \in \mathcal{A}$ is said to solve the problem $p \in \mathcal{P}$ with accuracy $\varepsilon_\tau > 0$ if:

$$\frac{h^e_{a,p}}{HV(T(Y^p), T(r_p))} \geq 1 - \varepsilon_\tau$$

where Y^p is the Pareto front reference of Problem p and r_p its associated reference point.

With this convergence test, one can associate a unique data profile d_{a_j} for each of the launches $a_j \in \mathcal{I}_a$ of the stochastic algorithm $a \in \mathcal{A}$ (designed as an *operational characteristic* in [35, 36]). The *average data profile* of the stochastic algorithm $a \in \mathcal{A}$ is then defined as

$$\bar{d}_a : k \in \mathbb{N} \mapsto \frac{1}{|\mathcal{I}_a|} \sum_{j=1}^{|\mathcal{I}_a|} d_{a_j}(k).$$

Similarly, the *lower bound* and *upper bound data profiles* of the stochastic algorithm $a \in \mathcal{A}$ are respectively defined as

$$d^l_a : k \in \mathbb{N} \mapsto \min_{1 \leq j \leq |\mathcal{I}_a|} d_{a_j}(k)$$

and

$$d^u_a : k \in \mathbb{N} \mapsto \max_{1 \leq j \leq |\mathcal{I}_a|} d_{a_j}(k).$$

These two data profiles delimit the variations of the performance of a given stochastic algorithm for a given set of instances.

6.3 Comparing different variants of DMulti-Mads

The aim of the tests presented in this section is to understand the impact of the different algorithmic of Dmulti-MADS on its computational efficiency. The experiments focus on three parameters:

- The *success iteration condition*: for Dmulti-MADS, an iteration is said to be successful if the algorithm generates a new point which dominates the current poll center. In order to compare with the DMS success strategy, a DMS strategy version of the DMulti-MADS algorithm is implemented. Specifically, an iteration of the DMS strategy version is marked as a success if a new non dominated point is found. The selection of the poll center and the update step of the new points remain the same; in case of a success, the poll center mesh size and frame size parameters remain constant. Note that with these conditions, it is possible to prove the same convergence results as for the DMulti-MADS algorithm.
- The *choice of the current incumbent*. Two selection strategies are considered. The first one picks the first element of the iterate list with the maximum mesh size and frame size parameters. The second one selects the poll center according to Algorithm 4, which includes the spread of the current non dominated points in its selection criterion.
- The *opportunistic strategy*. If the opportunistic strategy is activated, the iteration is stopped as soon as the algorithm finds a new point which triggers the success condition. Otherwise, the iteration continues until the end of the poll step.

These 8 variants are implemented in `Julia` and can be found at `https://github.com/bbopt/DMultiMadsEB`. For all variants, a speculative search strategy is implemented on the model of [3], as follows: considering the incumbent point x_k at iteration $k > 0$ generated during iteration $h < k$ with incumbent poll center x_h, one can build the target direction [6] $w_k = x_k - x_h$. If no failure iteration was ever observed at x_k, the search point $s_k = x_k + w_k$ is firstly evaluated before executing the poll step. Moreover, our implementation of the DMulti-MADS exploits the target direction w_k to reduce the number of polling directions to $n + 1$ directions without models as described in [6]. Preliminary tests show that this approach is more efficient than the classical OrthoMADS strategy [1]. This implementation is also the standard one in the `NOMAD` software when all models are deactivated for single objective optimization.

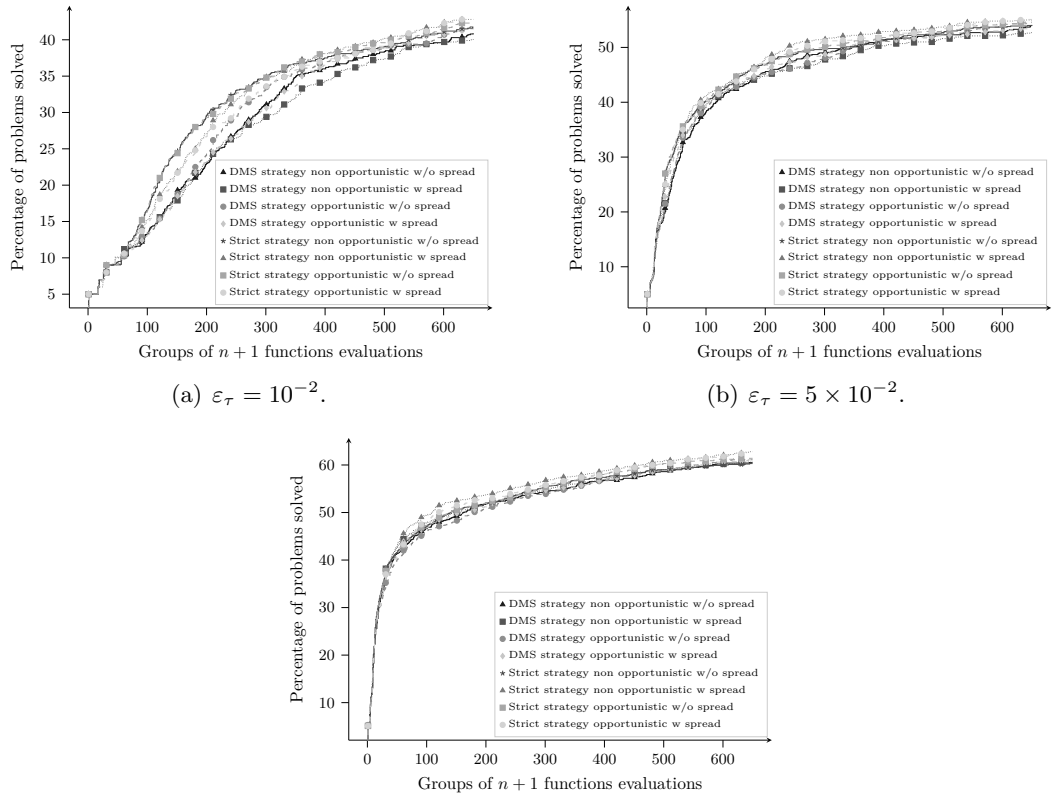
The mesh is implemented using the granular mesh strategy devised in [8]. All variants stop as soon as one component of the mesh size vector is below 10^{-9} or reach a maximum number of $20,000$ evaluations. For each problem, the variants start from the same set of initial points using the linesearch starting strategy exposed in [21]. For each problem and each variant, 10 replications are run by changing the random seed which controls the generation of the polling directions.

Figure 10 shows the data profiles for the whole set of variants with tolerance accuracies $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$. The main information that one can draw from these graphs is the fact that all strict success strategies outperform the DMS success strategies in most of the cases. For the highest tolerance accuracy $\varepsilon_\tau = 10^{-1}$ in Figure 10(c), the strict success strategy without opportunistic polling and spreading performs worse than all the DMS success strategies. But all remaining strict strategy variants are still better than the DMS success strategy variants.

To select the most efficient strategy variant among the strict strategies variants, Figure 11 shows the data profiles for all strict success strategies variants for a tolerance accuracy $\varepsilon_\tau = 10^{-2}$. Similar plots are obtained for tolerance accuracies $\varepsilon_\tau \in \{5 \times 10^{-2}, 10^{-1}\}$. For readability, we do not show them here. From this graph, one can observe than for a lower budget of evaluations (i.e. inferior to $400(n + 1)$ evaluations), strict success strategies variants without spreading solve slightly more problems than strategies with spread. However, for a larger budget of evaluations, using the spreading strategy in combination with the strict success strategy, performs better than without the spreading strategy. When using the spreading strategy coupled with the strict strategy, evaluating points opportunistically does not bring considerable advantages. Similar observations can be done based on Figure 10(a). From the rest of these experiments, we then keep the strict success strategies variants with and without the opportunistic stopping criteria as the most efficient DMulti-MADS strategies.

6.4 Comparing DMulti-MADS with other algorithms

This section presents the comparison of the two best Dmulti-MADS variants coded in `Julia` with other multiobjective derivative-free solvers BiMADS, DMS, MOIF and NSGA-II. The DMS and MOIF solvers have been used with their default settings as described in [16, 21]. Two variants of BiMADS, based on `NOMAD` version 3.9.1, are considered. The first uses the default settings of the MADS algorithm as implemented in `NOMAD` with state-of-the-art search step heuristics (see [6, 11, 18] for more details). The second one deactivates the search step heuristics such that the settings are equivalent to the DMulti-MADS implementation for a fairest comparison. Specifically, the number of poll directions is set to $n + 1$, with a speculative search step strategy enabled and an opportunistic polling strategy. For both variants, all single-objective runs terminate when the



(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

(c) $\varepsilon_\tau = 10^{-1}$.

Figure 10: Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [21] for all DMulti-MADS variants with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

mesh or step size parameter is below a threshold value (see [9] for specific details). All these deterministic solvers have a maximum budget of evaluations equal to $20,000$ and start from the same set of initial points as described before.

For NSGA-II, the population size is fixed to 100 points, with a total number of generations equal to 200, which is equivalent to a budget of $20,000$ blackbox evaluations. 50 instances of this stochastic solver are considered, corresponding to 50 different random seeds.

Figure 12 presents the data profiles obtained by the different solvers for the whole set of multiobjective problems. As BiMADS is a biobjective method, it is then not presented. For the lowest tolerance $\varepsilon_\tau = 10^{-2}$, the DMS solver is better than the others for a small to medium budget of evaluations. The two variants of DMulti-MADS outperform it when the allowed budget of evaluations is high 12(a) but dominate the MOIF and NSGA-II solvers for any considered budget. From Figures 12(b) and 12(c), one can observe that the two DMulti-MADS variants outperform all

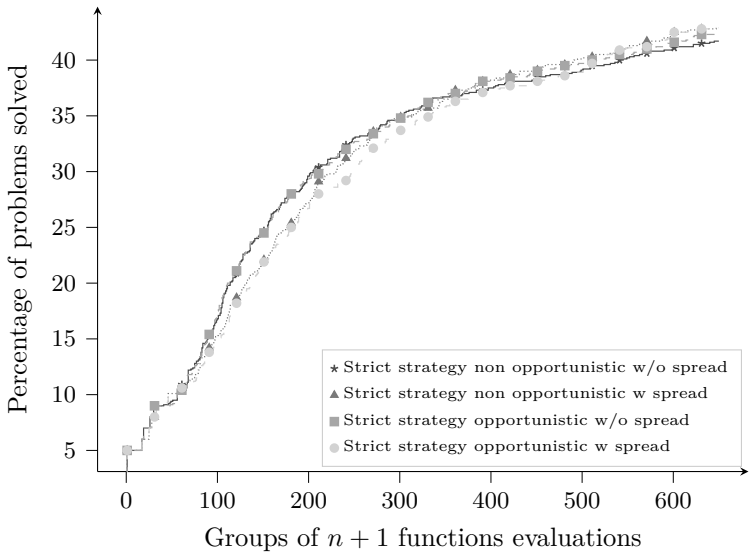


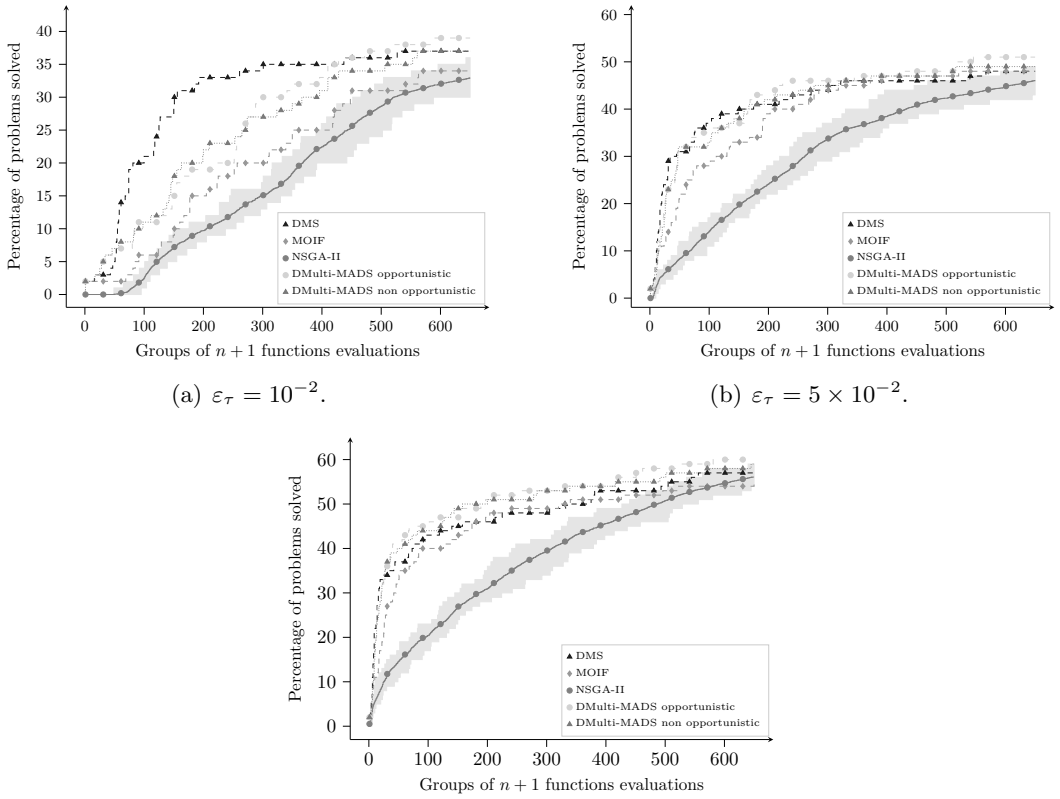
Figure 11: Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [21] for all strict success strategies variants of DMulti-MADS with tolerance $\varepsilon_\tau = 10^{-2}$.

the other solvers for a moderate to high budget of evaluations. Choosing an opportunistic strategy can increase the global performance of the DMulti-MADS algorithm on this set of problems.

From the data profiles obtained on the set of problems for $m = 2$ on Figure 13, where `NOMAD` (BiMADS) is added, similar results can be observed. The DMS solver is better than the others solvers with a small to moderate budget of blackbox evaluations but gets outperformed by the DMulti-MADS variants when one chooses a high budget of evaluations, for the lowest tolerance. For higher tolerances, Figures 13(b) and 13(c) illustrate the fact that the DMulti-MADS variant with opportunistic strategy is the dominating algorithm for a moderate to high budget of evaluations.

7 Conclusion

This work proposes a new extension of the MADS algorithm to multiobjective optimization, inspired by the works of [9, 21]. Contrary to the BIMADS and MultiMADS methods, the DMulti-MADS algorithm does not solve a succession of single-objective parameterized formulations. It directly updates a current list of non dominated points which gets closer to the Pareto front. This enables a better management of a given budget of evaluations to explore the feasible objective set. As in single-objective optimization, each iteration is built around a search and poll step. Theoretically, it is proved under mild assumptions that DMulti-MADS generates a succession of sequence of points whose stationary points are locally Pareto optimal. This convergence result is stronger than the proof presented in [21] which guarantees that the DMS algorithm is able to converge to at least one locally Pareto optimal point. However, the flexibility to choose the poll center, as it is the



(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

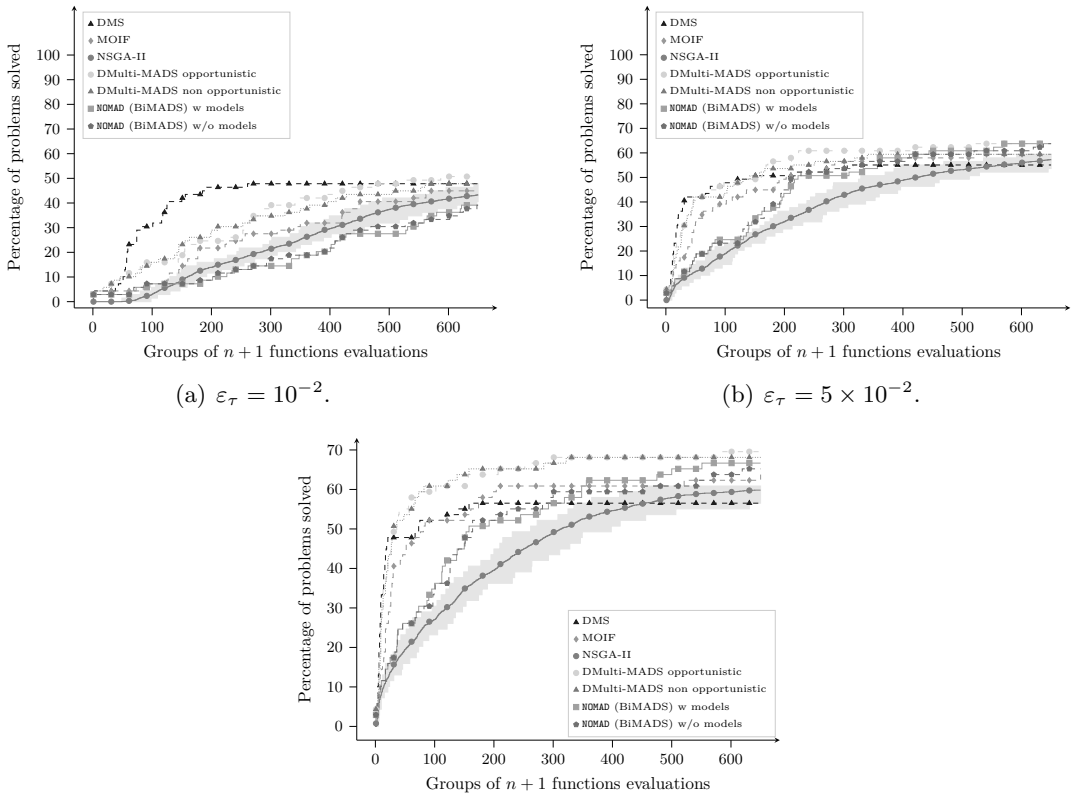
(c) $\varepsilon_\tau = 10^{-1}$.

Figure 12: Data profiles using DMS, DMulti-MADS, MOIF and NSGA-II obtained on 100 multiobjective optimization problems (69 with $m = 2$, 29 with $m = 3$ and 2 with $m = 4$) from [21] with 50 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

case for DMS, is lost. Computational results show that DMulti-MADS is competitive compared to other state-of-the-art blackbox multiobjective optimization techniques.

The selection mechanism of the poll center is a central part of the strong convergence properties of DMulti-MADS. Future research directions could adapt the convergence analysis of DMulti-MADS to other multiobjective optimization derivative-free methods with a posteriori preferences of articulations [16, 29, 33]. Indeed, all these methods look for improvements of a list of non dominated points possessing their own optimization parameter (trust-region radius, line step, and stepsize). Maximum optimization parameter selection could be tested on these methods.

In addition, many extensions could be implemented to improve performance of DMulti-MADS: search strategies assisted by surrogate models [18] or global search strategies [20], parallelism and so on. An integration of this algorithm in `NOMAD` is also planned.



(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

(c) $\varepsilon_\tau = 10^{-1}$.

Figure 13: Data profiles using NOMAD (BiMADS), DMS, DMulti-MADS, MOIF and NSGA-II obtained on 69 biobjective optimization problems from [21] with 50 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

Acknowledgments. The authors would like to thank Professor Ana Luísa Custódio (Universidade Nova de Lisboa) for interesting discussions on derivative-free multiobjective optimization.

References

[1] M.A. Abramson, C. Audet, J.E. Dennis, Jr., and S. Le Digabel. OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. SIAM Journal on Optimization, 20(2):948–966, 2009.

[2] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon. Performance indicators in multiobjective optimization. Technical Report G-2018-90, Les cahiers du GERAD, 2018.

[3] C. Audet and J.E. Dennis, Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM Journal on Optimization, 17(1):188–217, 2006.

[4] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009.

[5] C. Audet and W. Hare. Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, Cham, Switzerland, 2017.

[6] C. Audet, A. Ianni, S. Le Digabel, and C. Tribes. Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms. SIAM Journal on Optimization, 24(2):621–642, 2014.

[7] C. Audet, S. Le Digabel, and C. Tribes. Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization. Optimization and Engineering, 17(2):333–358, 2016.

[8] C. Audet, S. Le Digabel, and C. Tribes. The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables. SIAM Journal on Optimization, 29(2):1164–1189, 2019.

[9] C. Audet, G. Savard, and W. Zghal. Multiobjective Optimization Through a Series of Single-Objective Formulations. SIAM Journal on Optimization, 19(1):188–210, 2008.

[10] C. Audet, G. Savard, and W. Zghal. A mesh adaptive direct search algorithm for multiobjective optimization. European Journal of Operational Research, 204(3):545–556, 2010.

[11] C. Audet and C. Tribes. Mesh-based Nelder-Mead algorithm for inequality constrained optimization. Computational Optimization and Applications, 71(2):331–352, 2018.

[12] J. Blank and K. Deb. PYMOO - Multi-objective Optimization in Python. https://pymoo.org.

[13] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar. Quantitative performance assessment of multiobjective optimizers: The average runtime attainment function. In Heike Trautmann, Günter Rudolph, Kathrin Klamroth, Oliver Schütze, Margaret Wiecek, Yaochu Jin, and Christian Grimme, editors, Evolutionary Multi-Criterion Optimization, pages 103–119, Cham, 2017. Springer International Publishing.

[14] D. Brockhoff, T.D. Tran, and N. Hansen. Benchmarking numerical multiobjective optimizers revisited. In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, pages 639–646, New York, NY, USA, 2015. ACM.

[15] F.H. Clarke. Optimization and Nonsmooth Analysis. John Wiley & Sons, New York, 1983. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.

[16] G. Cocchi, G. Liuzzi, A. Papini, and M. Sciandrone. An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints. Computational Optimization and Applications, 69(2):267–296, 2018.

[17] Y. Collette and P. Siarry. Optimisation multiobjectif. Eyrolles, 2002.

[18] A.R. Conn and S. Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optimization Methods and Software, 28(1):139–158, 2013.

[19] A.R. Conn, K. Scheinberg, and L.N. Vicente. Introduction to Derivative-Free Optimization. MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[20] A. L. Custódio and J. F. A. Madeira. Multiglods: global and local multiobjective optimization using direct search. Journal of Global Optimization, 72(2):323–345, October 2018.

[21] 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.

[22] 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.

[23] Kalyanmoy Deb and Kaisa Miettinen. Multiobjective optimization: interactive and evolutionary approaches, volume 5252. Springer Science & Business Media, 2008.

[24] M. Ehrgott. Multicriteria Optimization, Volume 491 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, 2nd edition, 2005.

[25] C.M. Fonseca, L. Paquete, and M. Lopez-Ibanez. An improved dimension-sweep algorithm for the hypervolume indicator. In 2006 IEEE International Conference on Evolutionary Computation, pages 1157–1163. IEEE, 2006.

[26] M.S. Hasanoglu and M. Dolen. Multi-objective feasibility enhanced particle swarm optimization. Engineering Optimization, 50(12):2013–2037, 2018.

[27] J. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation, 10(1):50–66, 2006.

[28] M. Li and X. Yao. Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey. ACM Computing Surveys, 52(2):26:1–26:38, 2019.

[29] G. Liuzzi, S. Lucidi, and F. Rinaldi. A Derivative-Free Approach to Constrained Multiobjective Nonsmooth Optimization. SIAM Journal on Optimization, 26(4):2744–2774, 2016.

[30] J.J. Moré and S.M. Wild. Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

[31] J. Müller. SOCEMO: Surrogate Optimization of Computationally Expensive Multiobjective Problems. INFORMS Journal on Computing, 29(4):581–596, 2017.

[32] R.G. Regis. Multi-objective constrained black-box optimization using radial basis function surrogates. Journal of Computational Science, 16:140–155, 2016.

[33] J. Ryu and S. Kim. A derivative-free trust-region method for biobjective optimization. SIAM Journal on Optimization, 24(1):334–362, 2014.

[34] S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Mathematical Programming, 87(3):543–560, May 2000.

[35] Y.D. Sergeyev, D.E. Kvasov, and M.S. Mukhametzhanov. Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms. Mathematics and Computers in Simulation, 141:96–109, November 2017. New Trends in Numerical Analysis: Theory, Methods, Algorithms and Applications - NETNA 2015 (dedicated to Professor F.A. Costabile on his 70th birthday) held in Falerna (CZ), Italy during June 18–20, 2015.

[36] Y.D. Sergeyev, D.E. Kvasov, and M.S. Mukhametzhanov. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. Scientific Reports, 8(1):453, January 2018.

[37] V. Torczon. On the convergence of pattern search algorithms. SIAM Journal on Optimization, 7(1):1–25, 1997.

[38] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms — a comparative case study. In Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, Parallel Problem Solving from Nature — PPSN V, pages 292–301, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[39] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, 7(2):117–132, April 2003.