Handling of constraints in multiobjective blackbox optimization *

Jean Bigeon† Sébastien Le Digabel‡ Ludovic Salomon§

Abstract

This work proposes the integration of two new constraint-handling approaches into the blackbox constrained multiobjective optimization algorithm DMulti-MADS, an extension of the Mesh Adaptive Direct Search (MADS) algorithm for single-objective constrained optimization. The constraints are aggregated into a single constraint violation function which is used either in a two-phase approach, where research of a feasible point is prioritized if not available before improving the current solution set, or in a progressive barrier approach, where any trial point whose constraint violation function values are above a threshold are rejected. This threshold is progressively decreased along the iterations. As in the single-objective case, it is proved that these two variants generate feasible and/or infeasible sequences which converge either in the feasible case to a set of local Pareto optimal points or in the infeasible case to Clarke stationary points according to the constraint violation function. Computational experiments show that these two approaches are competitive with other state-of-the-art algorithms.

Key words. Multiobjective optimization, derivative-free optimization, blackbox optimization, constrained optimization, Clarke analysis.

1 Introduction

This work considers the following constrained multiobjective problem

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

where $\Omega = \{x \in \mathcal{X} : c_j(x) \leq 0,\ j \in \mathcal{J}\} \subset \mathbb{R}^n$ is the *feasible decision set*, and $\mathcal{X}$ a subset of $\mathbb{R}^n$. $\mathbb{R}^n$ and $\mathbb{R}^m$ are respectively designed as the *decision space* and the *objective space*. The functions $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ for $i = 1, 2, \ldots, m$ and $c_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ for $j \in \mathcal{J}$ are the outputs of a program seen as a blackbox. In this context, no gradient is available nor cannot be approximated and one cannot make any assumption on the structure of the problem (differentiability, continuity, convexity) in absence of analytical expressions for the objective and the constraint functions. Many engineering applications,

*GERAD and Département de mathématiques et génie industriel, Polytechnique Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec, Canada H3C 3A7. This work is supported by the NSERC RGPIN-2018-05286 Discovery Grant and an IVADO fellowship.

†www.univ-nantes.fr/jean-bigeon.

‡www.gerad.ca/Sebastien.Le.Digabel.

§www.gerad.ca/Ludovic.Salomon.

which involve several, costly and conflicting objectives, over a given set of constraints, fit into this framework (see for example [2, 29, 31, 49]). For more general information on derivative-free methods, the reader is referred to [10, 24, 39].

The description of the feasible decision set Ω enables the modeller to distinguish different types of constraints [41]. The set $\mathcal{X}$ is the set of *unrelaxable constraints*, which cannot be violated along the optimization process (e.g. strict bound constraints). The constraints in Ω constitute the set of *relaxable and quantifiable constraints*, that can be violated during the optimization, i.e. the blackbox will execute and the constraints outputs can be aggregated as a measure of violation of the constraints. Finally, *hidden constraints* constitute the set of points in the decision space for which the blackbox does not return any value, typically when the blackbox fails to execute. Allowing the f_i and c_j functions to take infinity values refers to this last type of constraints.

Furthermore, in a multiobjective optimization context, due to the conflict between different objectives, a solution is not always optimal for all criteria. The goal is then to provide the set of best trade-off solutions to the decision maker [19, 22, 46].

In single-objective optimization, many algorithms have been proposed to solve blackbox constrained optimization problems: direct search methods via the use of a filter [8, 6], a merit function [37] or an augmented Lagrangian [38], a derivative-free linesearch algorithm coupled with a penalty function [44], or quadratic model-based approaches (see for example [5, 16, 30]). The reader can refer to [39, Section 7] for a more thorough review.

Evolutionary algorithms [19] are popular methods to tackle constrained multiobjective optimization blackbox problems. Firstly investigated in the context of bound-constrained or unconstrained blackbox optimization, researchers have adapted some of them to take into account inequality constraints (see [50, 51] for more details). However, these methods are mostly stochastic heuristics. They practically require an important number of evaluations to perform. For example, the authors in [50] suggest a budget comprised between 2×10^5 and 5×10^5 function evaluations in their experiments, which can be impracticable when evaluations are too costly. Surrogate models remove this limitation by substituting true blackboxes by less expansive surrogates, such as radial basis functions (see [47]), or Gaussian processes (see [32]).

Recently, researchers have proposed extensions of convergence-based deterministic single-objective methods to multiobjective constrained optimization. Among the first ones, BiMADS [13, 13] and MultiMADS [14] are scalarization-based algorithms. These two algorithms reformulate the multiobjective optimization problem into a succession of single-objective subproblems. Each of them is solved by the single-objective constrained blackbox MADS algorithm [7, 8]. Practically, it can be difficult to correctly allocate the total budget of evaluations between all the subproblems, potentially resulting in a loss of evaluations required to improve the diversity and density of the current solution set.

The Direct MultiSearch (DMS) [26], its variants [20, 25, 28] and Derivative-Free MultiObjective (DFMO) [45] algorithms consider a different approach. They all keep a list of current feasible best non-dominated solutions that they improve along the iterations. DMS extends single-objective direct search algorithms to constrained multiobjective optimization. It rejects non-feasible points via the use of an extreme barrier function approach, i.e. non feasible points are assigned infinity values. This approach does not exploit knowledge of constraint violations values, which could potentially help to improve the solution set. Furthermore, DMS imposes the use of a feasible starting point, which is not practically available (too costly) in a real engineering context. DFMO extends a derivative-free linesearch algorithm to constrained multiobjective optimization. By aggregating constraints with the objective functions via the use of a penalty function, DFMO reduces the initial constrained multiobjective optimization problem to a simple bound constrained multiobjective optimization problem, easier to solve. However, its convergence assumptions are a bit restrictive in a blackbox optimization context, i.e. constraints functions and objective functions must be Lipschitz continuous. On the contrary, DMS requires that

objective functions should satisfy locally Lipschitz continuity. Besides, penalty function approaches can be sensitive to the scale of constraints (not always available in a blackbox context) and their penalty parameters values.

Based on these remarks, this work proposes two other ways to handle blackbox constraints, based on the DMulti-MADS algorithm [17, 17]. DMulti-MADS is an extension of the MADS algorithm to multiobjective optimization, strongly inspired by the DMS and BiMADS algorithms. It possesses stronger convergence properties than DMS. At the same time, experiments have shown its competitiveness according to state-of-the-art solvers on synthetic bound-constrained problems [17]. Similarly to DMS, the first version of DMulti-MADS, described in [17] requires a feasible starting point. The two extensions described below remove this limitation. The first one is an extension of the two-phase MADS algorithm described in [9] to constrained multiobjective optimization, named DMulti-MADS-TEB. The second version is an extension of the MADS algorithm with progressive barrier [8] to constrained multiobjective optimization, designed as DMulti-MADS-PB. Contrary to a penalty function approach, this last method

- is less sensitive to the scale of the outputs of the blackbox as it does not aggregate the constraints with the objective function;
- allows to explore around several incumbent points and not just one;
- is proved to be have stronger convergence properties than DFMO.

This work is organized as follows. Section 2 provides a summary of multiobjective optimization concepts. Section 3 introduces the core elements of the DMulti-MADS algorithm. Section 4 describes the DMulti-MADS-TEB and DMulti-MADS-PB variants to handle blackbox constraints. Main convergence results are detailed in Section 5. Finally, experiments are conducted in Section 6 on synthetic benchmarks and three real engineering applications in comparison with other state-of-the-art solvers.

2 Pareto dominance and optimal solutions in multiobjective optimization

This section summarizes some notation and concepts of multiobjective optimization. In order to characterize optimal solutions, one needs the concept of *Pareto dominance* [46].

Definition 2.1. Given two feasible decision vectors x^1 and x^2 in Ω,

- $x^1 \preceq x^2$ (x^1 *weakly dominates* x^2) if and only if $f_i(x^1) \leq f_i(x^2)$ for $i = 1, 2, \ldots, m$.
- $x^1 \prec x^2$ (x^1 *dominates* x^2) if and only if $f_i(x^1) \leq f_i(x^2)$ for $i = 1, 2, \ldots, m$ and there exists at least an index $i_0 \in \{1, 2, \ldots, m\}$ such that $f_{i_0}(x^1) < f_{i_0}(x^2)$.
- $x^1 \sim x^2$ (x^1 and x^2 are *incomparable*) if and only if x^1 does not dominate x^2 and x^2 does not dominate x^1.

With this definition, one is able to characterize locally optimal solutions and global optimal solutions in a multiobjective context.

Definition 2.2. A feasible decision vector $x^\star \in \Omega$ is said to be (globally) *Pareto optimal* if it does not exist any other decision vector $x \in \Omega$ which dominates $x^\star$.

Definition 2.3. A feasible decision vector $x^\star \in \Omega$ is said to be *locally Pareto optimal* if it exists a neighbourhood $\mathcal{N}(x^\star)$ of $x^\star$ such that there does not exist any other decision vector $x \in \mathcal{N}(x^\star) \cap \Omega$ which dominates $x^\star$.

The set of all Pareto optimal solutions in Ω is called the *Pareto set* denoted by $\mathcal{X}_p$ and its image by the objective function is designed as the *Pareto front* denoted by $\mathcal{Y}_P \subseteq \mathbb{R}^m$. Any set of locally Pareto optimal solutions is called a *local Pareto set*. Ideally, one would wish to find the entire Pareto set and consequently the entire Pareto front. But the Pareto set may be composed of an infinite number of solutions. In practice, an algorithm tries to find a representative set of nondominated points, denoted as a *Pareto set approximation* [52] (its mapping by the objective function f is designed as a *Pareto front approximation*). In the best case, a Pareto set approximation should be a subset of the Pareto set or a locally Pareto set, but this condition is not always satisfied.

Several objective vectors, i.e. points in the objective space, play an important role in multiobjective optimization as bounds on the Pareto front. The *ideal objective vector* [46] $y^I \in \mathbb{R}^m$ bounds the Pareto front from below and is defined as

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

From each component of the ideal objective vector, one can obtain information on the *extreme points of the Pareto set*, i.e. the elements of the Pareto set and solutions of each single-objective problem $\min_{x\in\Omega} f_i(x)$ for $i = 1, 2, \ldots, m$. The *nadir objective vector* [46] $y^{\mathrm{N}} \in \mathbb{R}^m$ provides an upper bound on the Pareto front. It is defined as

$$y^{\mathrm{N}} = \left(\max_{x\in\mathcal{X}_P} f_1(x), \max_{x\in\mathcal{X}_P} f_2(x), \max_{x\in\mathcal{X}_P} f_2(x), \ldots, \max_{x\in\mathcal{X}_P} f_m(x) \right)^\top .$$

3 The DMulti-MADS algorithm

DMulti-MADS [17] is a direct search iterative method designed to solve constrained multiobjective black-box optimization problems. It is an extension of the MADS [7] algorithm to multiobjective optimization, strongly inspired by the DMS [26] and BiMADS algorithms [13]. The notations and following definitions are taken from [8, 10].

Definition 3.1. At iteration k, the *set of feasible incumbent solutions* is defined as

$$F^k = \left\{ \arg\min_{x\in V^k} \{f(x) : x \in \Omega\} \right\}$$

where $V^k \subset \mathcal{X}$ is the set of trial points which have been evaluated by the start of iteration k.

Thus, all points in V^k satisfy the set of unrelaxable constraints $\mathcal{X}$. $V^0 \neq \emptyset \subset \mathcal{X}$ is then the set of starting points provided by the user. DMulti-MADS keeps an *iterate list of best feasible incumbents* found until iteration k, denoted as L_F^k and defined as

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

where Δ^l is the *frame size parameter* associated to the lth non-dominated point x^l of the list L_F^k. As L_F^k keeps only feasible non-dominated points, it is possible that $|F^k| \neq |L_F^k|$.

At the beginning of each iteration k, DMulti-MADS selects an element (x^k, Δ^k) of the list L_F^k as the current *feasible frame center*, and generates a finite number of new candidates. To ensure the convergence properties, all generated candidates during iteration k must belong to the mesh M^k defined by

$$M^k = \bigcup_{x\in V^k} \{x + \delta^k D z : z \in \mathbb{N}^{n_D}\} \subset \mathbb{R}^n$$

where $\delta^k > 0$ is the *mesh size parameter*; $D = GZ \in \mathbb{R}^{n \times n_D}$ is a matrix whose columns form a positive spanning set for $\mathbb{R}^n$ (see [10, Chapter 6] or [24, Chapter 2]) for some non-singular matrix $G \in \mathbb{R}^{n \times n}$ and some integer matrix $Z \in \mathbb{Z}^{n \times n_D}$. Note that G, Z and D do not depend on the iteration indexes. Generally, G and Z are chosen such as $G = I_n$ and $Z = [I_n \ -I_n] = D$, with I_n the identity matrix of dimensions $n \times n$. Furthermore, the current incumbent selection must satisfy at least the following condition:

$$(x^k, \Delta^k) \in \left\{(x, \Delta) \in L_F^k : \tau^{w^+} \Delta_{\max}^k \leq \Delta \leq \Delta_{\max}^k\right\}$$

where $\tau \in \mathbb{Q} \cap (0, 1)$ is the *frame size adjustment parameter*, $w^+ \in \mathbb{N}$ a fixed integer and $\Delta_{\max}^k$ the *maximum frame size parameter* at iteration k defined as

$$\Delta_{\max}^k = \max_{(x,\Delta) \in L_F^k} \Delta.$$

The mesh size parameter δ^l and frame size parameter Δ^l associated to the lth non-dominated point x^l of L_F^k are linked to each other such that $0 < \delta^l \leq \Delta^l$. When a subsequence of one of them goes to 0, so does the other. Typically, the following relation $\delta^l = \min\left\{\Delta^l, (\Delta^l)^2\right\}$ meets these requirements.

Each iteration is decomposed into two steps: the *search* and the *poll*. The search is an optional and flexible step which enables the user to design any strategy as long as the proposed trial points belong to the mesh M^k and their number is finite. A common strategy is the use of surrogate models, proposed for example in [20]. The finite set of points used in the search is denoted by S^k.

The poll is more rigorously defined, as the convergence analysis depends on it. The trial points involved in this step, named the poll set and denoted by P^k, must satisfy some specific requirements. More precisely, the construction of P^k involves the use of the current incumbent x^k and its associated frame size Δ^k and mesh size δ^k parameters to obtain a positive spanning set $\mathbb{D}_\Delta^k$. Each column of $\mathbb{D}_\Delta^k$ must be a nonnegative integer combination of the directions in D; the distance from the current incumbent x^k to a poll point must be bounded by a multiple of the frame size parameter Δ^k. Note that the relation between δ^k and Δ^k given above meets these requirements. Formally, P^k is described as

$$P^k = \{x^k + \delta^k d : d \in \mathbb{D}_\Delta^k\} \subset M^k.$$

All new candidates generated during the search and the poll are assigned a frame size parameter value larger or equal to the frame size parameter of the current feasible frame center.

If a new generated candidate dominates the current feasible incumbent, the iteration is marked as a *success*. Otherwise, it is a *failure* and the frame size parameter (and so the mesh size parameter) of the current feasible frame center is decreased. The iteration can be opportunistic, meaning that as soon as it is successful, the remaining candidates (if they exist) are not evaluated. In all cases, the iterate list L_F^k is filtered to keep only best non-dominated feasible points found until the end of this iteration.

More details can be found in [17].

4 Handling of constraints with DMulti-MADS

This section details several strategies to handle constraints with DMulti-MADS. The set of quantifiable and relaxable constraints is given by $\Omega = \{x \in \mathcal{X} : c_j(x) \leq 0, \ j \in \mathcal{J}\}$. A relaxable constraint can be violated during the optimization and still returns meaningful outputs for the blackbox. A quantifiable constraint provides a measure of violation of feasibility. All other types of constraints (unrelaxable, hidden, non quantifiable), if present are added to $\mathcal{X}$.

4.1 The constraint violation function

Exploiting constraints to guide the algorithm towards optimal solutions requires a way to quantify constraint violations. The strategies described below rely on the *constraint violation function* $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ used in [8] and defined by

$$h(x) = \begin{cases} \sum_{j \in \mathcal{J}} \left(\max\{c_j(x), 0\}\right)^2 & \text{if } x \in \mathcal{X}, \\ +\infty & \text{otherwise.} \end{cases}$$

With this definition, x belongs to Ω if and only if $h(x) = 0$, and $0 < h(x) < +\infty$ when x is infeasible but belongs to $\mathcal{X} \setminus \Omega$. The use of a squared function instead of common ℓ^1 norm enables some conservation of first-order smoothness properties.

4.2 The extreme barrier (EB)

Similarly to DMS [26], the original version of the DMulti-MADS algorithm [17] treats constraints via the use of an extreme barrier approach. It replaces the objective function f by

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

In other terms, all infeasible points are assigned an infinite value. This approach requires a feasible starting point, which is not always available in an engineering context. To allow the use of an infeasible starting point, this work proposes a *Two-phase Extreme Barrier (TEB)* approach, in the continuation of [9]. When starting from an infeasible point, the new strategy, called DMulti-MADS-TEB, performs a single-objective minimization of the h constraint violation function using the MADS algorithm. As soon as a feasible point is found, DMulti-MADS-TEB moves to the second phase, which is the minimization of 1 from the feasible point found in the first phase.

Although this approach is simple, its performance has never been investigated in the context of deterministic multiobjective derivative-free optimization. It also shares some convergence properties with MADS and DMulti-MADS, summarized in Section 5. Note that this strategy can be applied to any multiobjective blackbox algorithm.

4.3 The progressive barrier (PB)

This subsection introduces the DMulti-MADS-PB extension of the single-objective MADS-PB algorithm [8] for multiobjective derivative-free optimization.

4.3.1 Feasible and infeasible incumbents

Similarly to the MADS-PB algorithm [8], DMulti-MADS-PB constructs two sets of incumbent solutions from V^k. F^k still denotes the set of feasible incumbent solutions. To define the set of infeasible incumbent solutions, one needs to extend the notion of dominance for infeasible solutions, as it is required in the design of filter algorithms [33, 34].

Definition 4.1 (Dominance relation for constrained multiobjective optimization)**.** In the context of constrained multiobjective optimization, $x^1 \in \mathcal{X}$ is said to dominate $x^2 \in \mathcal{X}$ if

- Both points are feasible and $x^1 \in \Omega$ dominates $x^2 \in \Omega$, denoted as $x^1 \prec_f x^2$.
- Both points are infeasible and $f_i(x^1) \leq f_i(x^2)$ for $i = 1, 2, \ldots, m$ and $h(x^1) \leq h(x^2)$ with at least one strict inequality, denoted as $x^1 \prec_h x^2$.

This extension of the dominance relation is different from the definition proposed in [47]. Indeed, in this work, feasible and infeasible points are never compared, and the dominance relation takes into account both objective function values and the constraint violation function values. Another extension of dominance to constrained optimization appears in [35], but as in the previous case, it allows the comparison of feasible and infeasible points. Note that if $m = 1$, the dominance relation reduces into the dominance relation of MADS-PB [8].

With this dominance relation, one can define the set of infeasible nondominated points.

Definition 4.2. At iteration k, the *set of infeasible nondominated points* is defined as

$$U^k = \left\{ x \in V^k \setminus \Omega : \text{ there is no } y \text{ such that } y \prec_h x \right\}.$$

As for the MADS-PB algorithm, DMulti-MADS-PB relies on a nonnegative barrier threshold $h^k_{\max}$, set at each iteration k, to construct the set of infeasible incumbent solutions.

Definition 4.3. At iteration k, the *set of infeasible incumbent solutions* is

$$I^k = \left\{ \arg\min_{x \in U^k} \left\{ f(x) : 0 < h(x) \leq h^k_{\max} \right\} \right\}.$$

All evaluated points having a value of h above $h^k_{\max}$ are automatically rejected by the algorithm. Furthermore, the barrier threshold is nonincreasing with the iteration number k. Its value at each iteration is detailed in Section 4.3.3.

Figure 1 illustrates these definitions. Note that I^k is not a singleton. The images of fourteen trial points generated at the beginning of iteration k, i.e. V^k, in the "augmented" objective space (a triobjective space with two objectives f_1, f_2 and the constraint violation function h) for a biobjective minimization optimization problem are represented. The set of feasible incumbent solutions, indicated by black bullets, contains four elements. Two other generated points are equally visible, but each of them is dominated by a feasible incumbent solution. These six generated trial points belong to the biobjective space. The set of infeasible non-dominated points contains six elements, identified by black lozenges and diamonds. Among them, only three qualify to be infeasible incumbent solutions. Indeed, one element among the others is above the threshold value $h^k_{\max}$. The two other ones are dominated by at least one solution of I^k in terms of f objective values. Two elements of U^k dominate the two last remaining trial points, marked by $\times$ symbols. Notice that all elements among I^k and F^k could have been generated before iteration $k-1$, by definition of V^k.

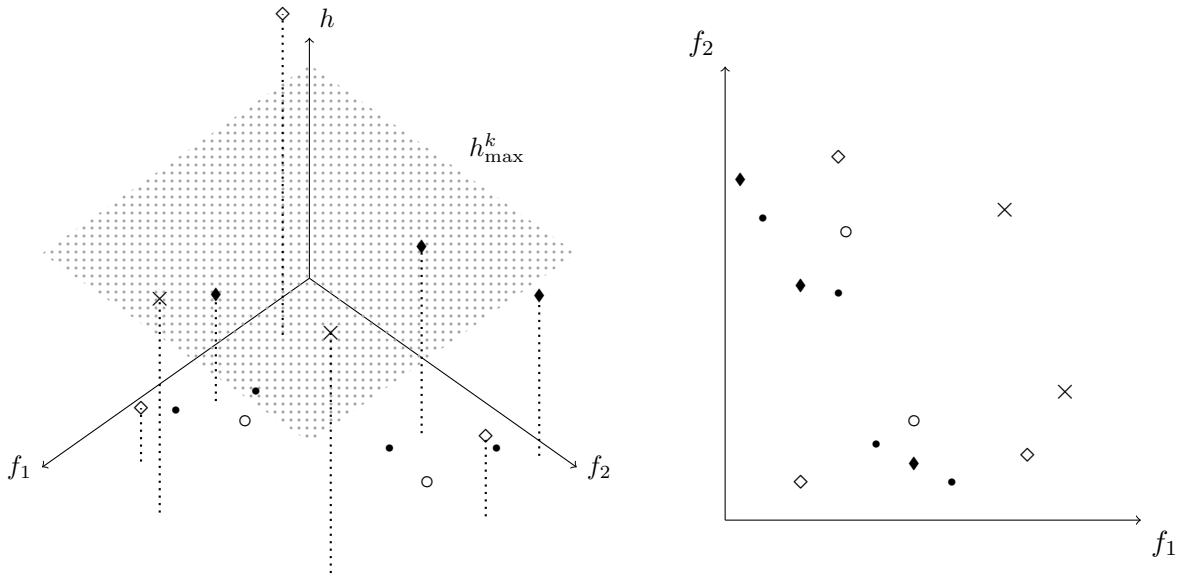
From the sets F^k and I^k, DMulti-MADS constructs two lists of incumbent solutions, the iterate list of best feasible incumbents found until iteration k,

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

and the *iterate list of best infeasible incumbents* found until iteration k

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

Each element of both lists possesses its own associated frame size parameter Δ^l.



- • Set of feasible non dominated points F^k
- ♦ Set of infeasible incumbent solutions I^k
- ◇ Subset of the set of infeasible non-dominated points $U^k \setminus I^k$
- ○ Feasible dominated points
- × Infeasible dominated points

Figure 1: An example of feasible and infeasible incumbent solutions at iteration k for a biobjective minimization problem in the "augmented" objective space (a triobjective space with the two objectives f_1, f_2 and the constraint violation function h). On the left, a 3D view; on the right, the projection on the biobjective space.

4.3.2 An iteration of the DMulti-MADS-PB algorithm

As for the single-objective optimization MADS-PB algorithm [8], the search and the poll which constitute the two steps of an iteration for DMulti-MADS-PB are organized around two iterate incumbents at iteration k: a feasible one $(x_F^k, \Delta_F^k) \in L_F^k$ and an infeasible one $(x_I^k, \Delta_I^k) \in L_I^k$. However, as the frame size parameters associated to the feasible incumbent x_F^k and infeasible incumbent x_I^k can be distinct, it is necessary to adapt the definition of the mesh M^k. At iteration k, M^k is defined as

$$M^k = \begin{cases} \bigcup\limits_{x \in V^k} \{x + \delta_F^k Dz : z \in \mathbb{N}^{n_D}\} & \text{if } L_F^k \neq \emptyset; \\ \bigcup\limits_{x \in V^k} \{x + \delta_I^k Dz : z \in \mathbb{N}^{n_D}\} & \text{otherwise,} \end{cases}$$

where $\delta_F^k > 0$ and $\delta_I^k > 0$ are respectively the mesh size parameters associated to the feasible and infeasible incumbents x_F^k and x_I^k defined as $\delta_F^k = \min\left\{\Delta_F^k, (\Delta_F^k)^2\right\}$ and $\delta_I^k = \min\left\{\Delta_I^k, (\Delta_I^k)^2\right\}$. In other terms, the configuration of the mesh M^k at iteration k is primarily based on the selection of the feasible frame center if this last one exists.

It is then possible to adapt the definition of the poll set P^k. At iteration k, P^k is defined as

$$P^k = \begin{cases} P^k(x_F^k, \Delta_F^k) & \text{for some } (x_F^k, \Delta_F^k) \in L_F^k \text{ if } L_I^k = \emptyset, \\ P^k(x_I^k, \Delta_I^k) & \text{for some } (x_I^k, \Delta_I^k) \in L_I^k \text{ if } L_F^k = \emptyset, \\ P^k(x_F^k, \Delta_F^k) \cup P^k(x_I^k, \Delta_F^k) & \text{for some } (x_F^k, \Delta_F^k) \in L_F^k \text{ and } (x_I^k, \Delta_I^k) \in L_I^k, \text{ otherwise,} \end{cases}$$

where $P^k(x, \Delta^k) = \{x + \delta^k d : d \in \mathbb{D}_\Delta^k\} \subset M^k$ represents the poll set centered at x at iteration k with $\delta^k = \min\left\{\Delta^k, (\Delta^k)^2\right\}$.

Figure 2 illustrates a construction of the poll set P^k when both the feasible frame center x_F^k and the infeasible frame center x_I^k exist. Here, $\Omega \subset \mathbb{R}^2$. All poll candidates belong to one of the frames generated by x_F^k or x_I^k of size $\Delta_F^k > 0$ (this is not mandatory as long as the definition of the poll holds). The set P^k is the union of the sets $P^k(x_F^k, \Delta_F^k) = \left\{p^1, p^2, p^3, p^4\right\}$ and $P^k(x_I^k, \Delta_I^k) = \left\{p^5, p^6\right\}$. Section 4.3.4 gives more implementation details on the construction of the poll set.

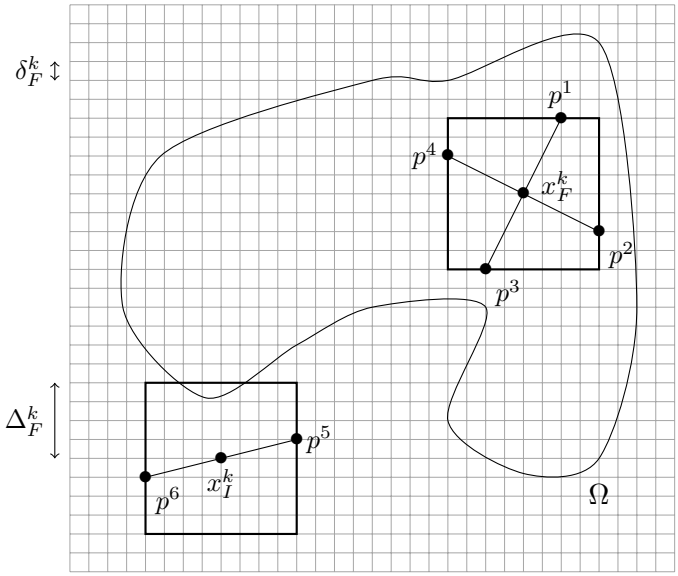


Figure 2: Example of a poll set $P^k = \left\{p^1, p^2, p^3, p^4, p^5, p^6\right\}$ for $\Omega \subset \mathbb{R}^2$ when both x_F^k and x_I^k exist (inspired by [8]).

It remains to address the choice of the feasible and infeasible frame centers at iteration k. In the case of the MADS-PB algorithm, the set of feasible and infeasible incumbent solutions are often singletons (or composed of points which have the same objective function value and the same h-constrained value). Their selection is then unambiguous.

When L_F^k possesses at least one element, the choice of the current feasible frame center must satisfy the same condition as described in Section 3. Practically, to get a good Pareto front approximation, it is also recommended to take into consideration the gap between the different non-dominated solutions found until iteration k, as it is done in [17].

If L_F^k is empty, the infeasible frame center must satisfy

$$(x_I^k, \Delta_I^k) \in \left\{(x, \Delta) \in L_I^k : \Delta_{h_{\min}}^k \leq \Delta\right\}$$

where $\Delta^k_{h_{\min}}$ is defined as

$$(x^k_{h_{\min}}, \Delta^k_{h_{\min}}) \in \arg \min_{(x,\Delta)\in L^k_I} h(x).$$

The idea behind this selection criterion is to prioritize exploration along the least infeasible point with the best objective values hoping to find a "good" feasible point. At the same time, this selection criterion allows to explore some potentially interesting regions of the objective space. Intuitively, the infeasible current best incumbents associated with a frame size parameter value superior to the least current infeasible point are the ones which have not yet been explored or are promising, due to the update procedure, detailed in Section 4.3.3. Several infeasible incumbents can satisfy this criterion. Then, following the selection procedure described in [17], this work proposes Algorithm 1 to take into account the density of the set of best infeasible incumbents in the objective space.

There remains the case where L^k_F and L^k_I are both non-empty. A first approach would be to independently select the feasible and infeasible frame centers, based for example on a spacing criterion to densify the set of best feasible and best infeasible current solutions. However, this strategy does not exploit the "dominance" order which exists between both sets. More precisely, one could hope that exploring around a carefully chosen infeasible incumbent leads to the generation of a new feasible point which significantly improves the set of current feasible solutions. The proposed approach is inspired by the works of [42].

At iteration k, considering the non-empty iterate list of feasible incumbents L^k_F, this work introduces the function $\psi_{L^k_F} : \mathcal{X} \to \mathbb{R}$ given as

$$\begin{aligned}\psi_{L^k_F}(x) &= \Phi_{L^k_F}(f(x)) \\ &= \begin{cases} \min\limits_{(x^F,\Delta)\in L^k_F} \sum\limits_{i=1}^{m} \left[f_i(x^F) - \min\left\{f_i(x), f_i(x^F)\right\}\right] & \text{if there is no } (x^F,\Delta) \in L^k_F \text{ such that} \\ & f_i(x^F) \leq f_i(x) \text{ for } i = 1, 2, \ldots, m; \\ -\min\limits_{(x^F,\Delta)\in L^k_F} \sum\limits_{i=1}^{m} \left[f_i(x) - \min\left\{f_i(x), f_i(x^F)\right\}\right] & \text{otherwise.} \end{cases}\end{aligned}$$

The level sets of $\Phi_{L^k_F}$ are depicted in Figure 4. Note that all potential feasible decision vectors which are not dominated by a current feasible incumbent solution of L^k_F are given a positive $\psi_{L^k_F}$ value. All dominated feasible decision vectors correspond to a negative $\psi_{L^k_F}$ value.

The current infeasible frame center is then chosen as the element of L^k_I which maximizes the $\psi_{L^k_F}$ function, i.e.

$$(x^k_I, \Delta^k_I) \in \arg \max_{(x,\Delta)\in L^k_I} \psi_{L^k_F}(x).$$

Intuitively, exploring around an infeasible frame center with a large positive value can lead to the generation of a feasible point which significantly improves the current Pareto front approximation. If the selected infeasible frame center possesses a negative value, it is "close" to the non-dominated zone relative to the current Pareto front approximation. An exploration around it can still improve the current feasible set.

4.3.3 Update of the mesh parameter at the end of an iteration

At the end of the search and the poll at iteration k, DMulti-MADS-PB has evaluated a finite number of candidates on the mesh M^k. The cache V^{k+1} is then the union of the cache V^k at the beginning of iteration k and all the candidates evaluated during iteration k. As for MADS-PB [8], the values of f and

Algorithm 1 `selectCurrentInfeasibleIncumbent`(L_I^k)

Let $L_I^{select} := \left\{(x,\Delta) \in L_I^k : \Delta_{h_{\min}}^k \leq \Delta\right\}$ with $\Delta_{h_{\min}}^k = \arg\min_{(x,\Delta)\in L_I^k} h(x)$.

if $|L_I^{select}| = 1$ **then**
  return (x,Δ) with $L_I^{select} = \{(x,\Delta)\}$.
else if $|L_I^{select}| = 2$ and $|L_I^k| = 2$ **then**
  Let $l_0 \in \arg\max_{l=1,2} \max_{i=1,2,\ldots,m} f_i(x^l)$.
  return (x^{l_0}, Δ^{l_0}).
else
  for $i = 1, 2, \ldots, m$ **do**
    Order $L_I^k = \left\{(x^1,\Delta^1),(x^2,\Delta^2),\ldots,(x^{|L_I^k|},\Delta^{|L_I^k|})\right\}$ such that
    $f_i(x^1) \leq f_i(x^2) \leq \ldots \leq f_i(x^{|L^k|})$.
    for $l = 1, 2, \ldots, |L_I^k|$ **do**
      Compute $\gamma_i(x^l)$ defined as

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

    end for
  end for
  Let $l_0 \in \arg\max_{l=1,2,\ldots,|L_I^{select}|} \max_{i=1,2,\ldots,m} \gamma_i(x^l)$.
  return (x^{l_0}, Δ^{l_0}).
end if

Figure 3: A procedure to select the current incumbent at iteration k taking into account the spacing between elements of the iterate list of best infeasible incumbents L_I^k in the objective space, inspired by [17].

h stored in V^{k+1} for DMulti-MADS-PB determine the way the threshold value $h_{\max}^{k+1}$ (see (1)) and the mesh and frame size parameters of the elements of iterate lists of feasible and infeasible incumbents L_F^{k+1} and L_I^{k+1} are updated.

Similarly to MADS-PB [8], this work uses the concept of *dominating*, *improving* and *unsuccessful* iteration. A dominating iteration occurs when DMulti-MADS-PB generates a trial point which dominates a current frame incumbent. An improving iteration is not dominating but improves the feasibility of the infeasible frame center. Otherwise, the iteration is unsuccessful. More precisely,

- Iteration k is said to be dominating whenever a trial point $x^t \in V^{k+1}$ dominates one frame incum-

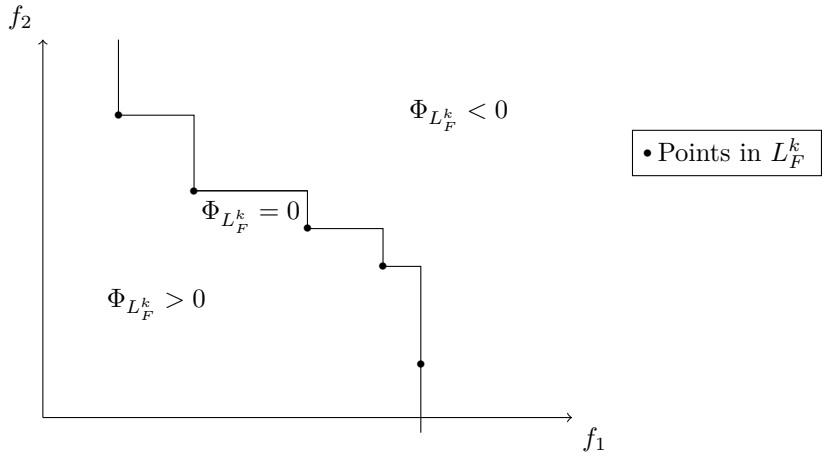


Figure 4: Level sets in the objective space of $\Phi_{L_F^k}$ for a biobjective minimization problem.

bent, i.e.

$$h(x^t) = 0 \text{ and } x^t \prec_f x_F^k \text{ or } h(x^t) > 0 \text{ and } x^t \prec_h x_I^k$$

is found.

- Iteration k is said to be improving if it is not dominating, but generates a trial point $x^t \in V^{k+1}$ which satisfies

$$0 < h(x^t) < h(x_I^k) \text{ and there exists } i_0 \in \{1, 2, \ldots, m\} \text{ such that } f_{i_0}(x_I^k) < f_{i_0}(x^t).$$

- Iterations which are neither dominating nor improving are labelled as unsuccessful. It happens when every trial point $x^t \in V^{k+1}$ is such that

$$h(x^t) = 0 \text{ and } x^t \not\prec_f x_F^k, \text{ or } h(x^t) = h(x_I^k) \text{ and } x^t \not\prec_h x_I^k \text{ or } h(x^t) > h(x_I^k).$$

These three cases are described in Figure 5.

All points generated during iteration k are given a frame size parameter $\Delta \geq \Delta^k$ where Δ^k is the frame size parameter associated to M^k. More precisely, for any trial element (x^t, Δ) generated during iteration k,

$$(x^t, \Delta) = \begin{cases} (x^t, \tau^{-1}\Delta^k) & \text{if } h(x^t) = 0 \text{ and there exists at least } x \in F^k \text{ such that } x \prec_f x^t, \text{ or} \\ (x^t, \tau^{-1}\Delta^k) & \text{if } h(x^t) = 0 \text{ and for } i = 1, 2, \ldots, m, f_i(x^t) \leq \min_{x \in F^k} f_i(x) \text{ with at least an} \\ & \text{index } i_0 \in \{1, 2, \ldots, m\} \text{ such that } f_{i_0}(x^t) < \min_{x \in F^k} f_{i_0}(x), \text{ or} \\ (x^t, \tau^{-1}\Delta^k) & \text{if } h(x^t) > 0 \text{ and there exists at least } x \in I^k \text{ such that } x^t \prec_h x, \text{ or} \\ (x^t, \tau^{-1}\Delta^k) & \text{if } 0 < h(x^t) \leq \max_{x \in I^k} h(x) \text{ and for } i = 1, 2, \ldots, m, f_i(x^t) \leq \min_{x \in I^k} f_i(x) \\ & \text{with at least one index } i_0 \in \{1, 2, \ldots, m\} \text{ such that } f_{i_0}(x^t) < \min_{x \in I^k} f_{i_0}(x), \\ (x^t, \Delta^k) & \text{otherwise;} \end{cases}$$

where $\tau \in (0, 1) \cap \mathbb{Q}$ is the frame size adjustment parameter chosen by the user. Thus, all candidates which dominate one of the points in L_F^k or L_I^k or improve the extent of the objectives values covered by at

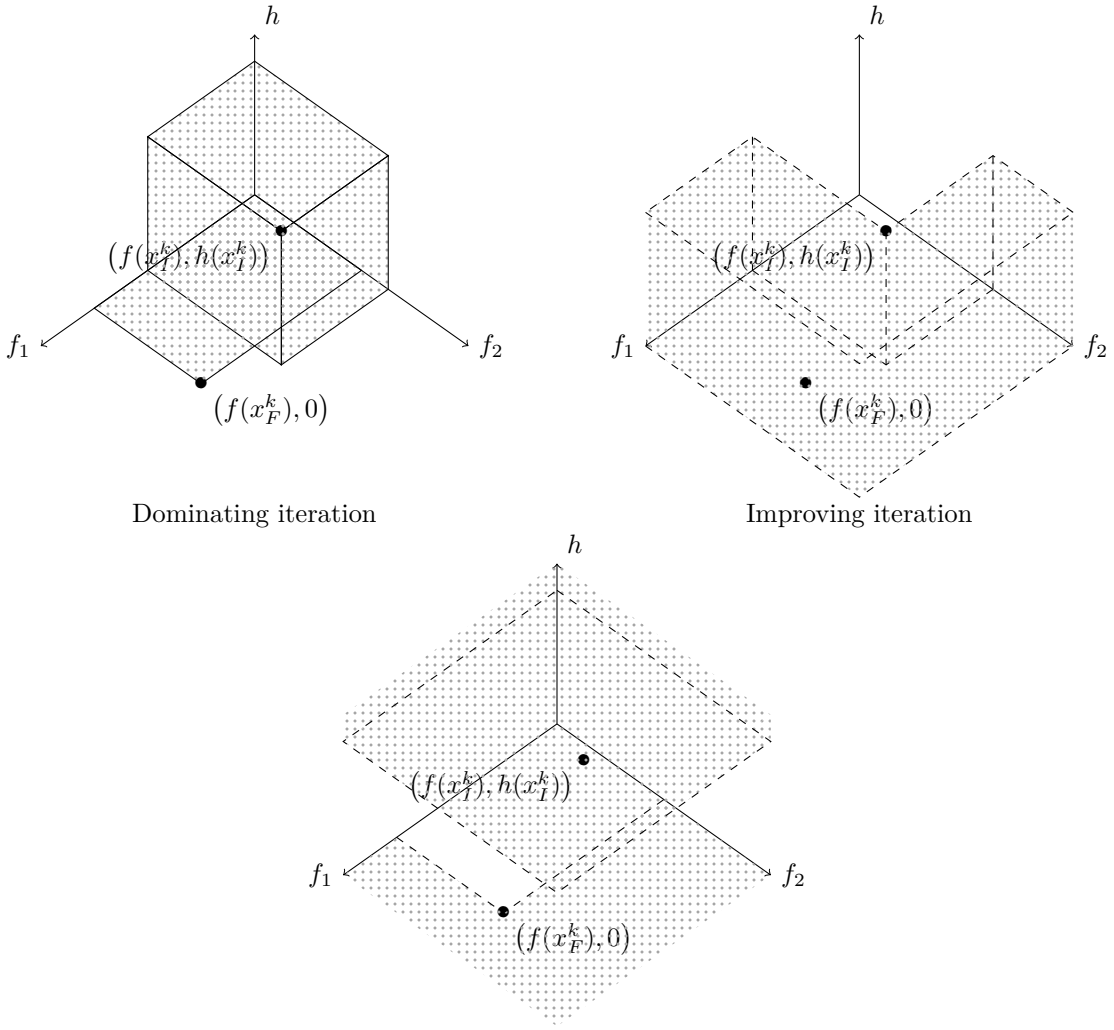


Figure 5: Iterations cases for DMulti-MADS-PB.

least one of the iterate list have their associated frame size parameter increased. When L_F^k is empty and no feasible point has been generated at iteration k, these candidates are likely to be potential frame center candidates at iteration k+1. If L_F^k is not empty, the update of the frame size parameter associated to a new feasible generated point is similar to the one proposed in the original DMulti-MADS-EB algorithm [17]. If the iteration is labelled as unsuccessful, no generated point at the end of iteration k dominates at least one of the frame center incumbents. In this case, DMulti-MADS-PB replaces $(x_{center}^k, \Delta_{center}^k)$ by $(x_{center}^k, \tau\Delta^k)$ with $\tau \in (0,1) \cap \mathbb{Q}$ and $x_{center}^k \in \{\{x_F^k\}, \{x_I^k\}, \{x_F^k, x_I^k\}\}$ relatively to the emptiness of the iterate lists L_F^k or L_I^k. If the iteration is improving, the frame size parameters associated to the existing frame center incumbents keep the same value as in iteration k.

Figure 6 illustrates the frame update rules for a biobjective minimization problem in the "augmented"

objective space (a triobjective space with two objectives f_1, f_2 and the constraint violation function h). All candidates whose image is outside combined gray areas are affected a frame size parameter $\Delta := \Delta^k$.

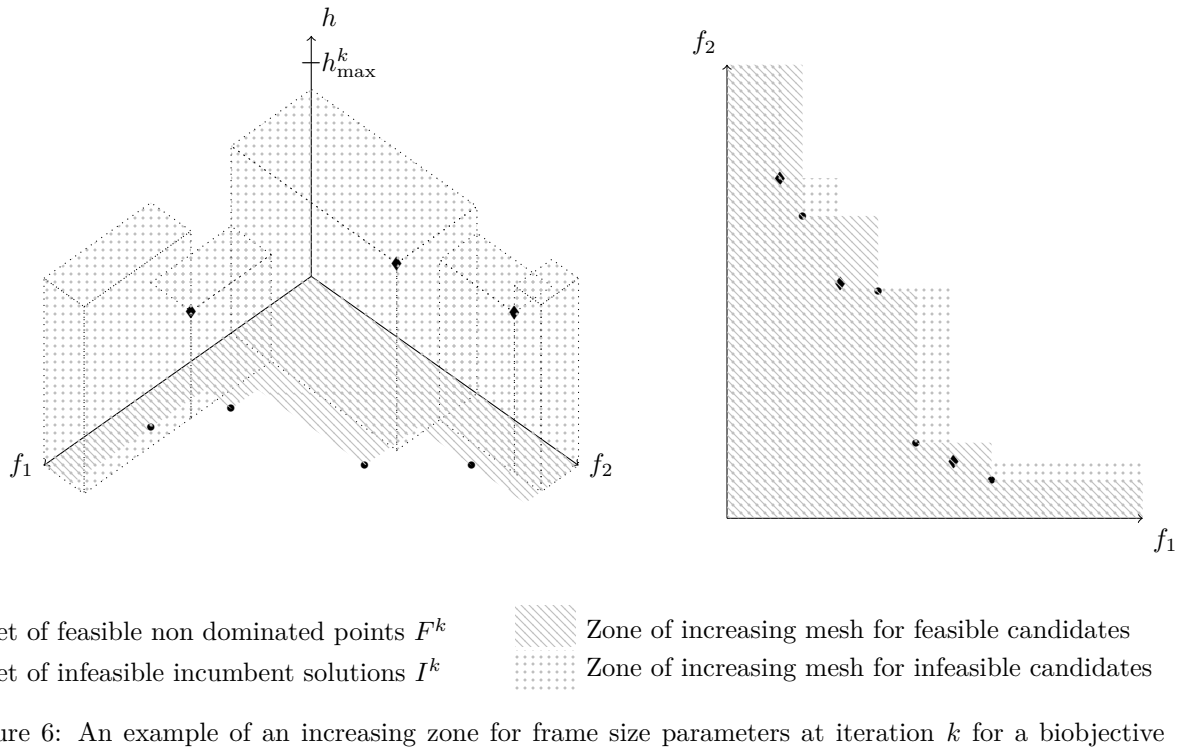


Figure 6: An example of an increasing zone for frame size parameters at iteration k for a biobjective minimization problem. On the left, a 3D view of the "augmented" objective space (a triobjective space with the two objectives f_1, f_2 and the constraint violation function h); on the right, projection on the biobjective space.

The threshold barrier is then updated according to the following rules:

$$h^{k+1}_{\max} := \begin{cases} \max\limits_{x^t \in V^{k+1}} \left\{ h(x^t) : h(x^t) < h(x^k_I) \right\} & \text{if iteration } k \text{ is improving,} \\ h(x^k_I) & \text{if } h(x^k_I) = \max\limits_{x \in I^k} h(x), \\ \max\limits_{x^t \in V^{k+1}} \left\{ h(x^t) : h(x^k_I) \leq h(x^t) < \max\limits_{x \in I^k} h(x) \right\} & \text{otherwise.} \end{cases} \tag{1}$$

The threshold update rule guarantees in the case where an iteration is considered as not improving that the set I^k will change if the infeasible frame incumbent does not possess the maximum violation function value h among the elements of I^k at iteration k. Another consequence (similar to MADS-PB [8]) is that $h^k_{\max}$ is nonincreasing with iteration k and that if $I^k \neq \emptyset$, $I^q \neq \emptyset$ for all iteration indexes $q \geq k$.

Note that even if an iteration is marked as unsuccessful, the algorithm can still generate new feasible non-dominated points or infeasible non-dominated points below the value $\max_{x \in I^k} h(x)$, which may be used as frame incumbents in some next iteration.

Remark. It is also possible to set the update rules of the threshold $h^{k+1}_{\max}$ according to the $h(x^k_I)$ barrier value. Nonetheless, in some preliminary experiments, it has been observed that this approach prevents

the algorithm to explore some parts of the objective space, potentially interesting to greatly improve the current feasible solution set.

Finally, the iterate lists L_F^k and L_I^k are filtered to add new non-dominated points generated during iteration k and remove potential resulting dominated elements.

Algorithm 2 summarizes the different steps of the DMulti-MADS-PB algorithm.

Algorithm 2 The DMulti-MADS-PB algorithm for constrained optimization

---

Initialisation : Given a finite set of points $V^0 \subset \mathcal{X}$, choose $\Delta^0 > 0$, $D = GZ$ a positive spanning set matrix, $\tau \in (0,1) \cap \mathbb{Q}$ the frame size adjustment parameter, and $w^+ \in \mathbb{N}$ a fixed integer parameter. Define the frame trigger parameter $\rho > 0$ (optional). Initialize the lists $L_F^0 = \left\{(x_F^l, \Delta^0),\ l = 1, 2, \ldots, |L_F^0|\right\}$ and $L_I^0 = \left\{(x_I^l, \Delta^0),\ l = 1, 2, \ldots, |L_I^0|\right\}$ for some $(x_F^l, x_I^l) \in V^0$.

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

Selection of the current infeasible frame centers. Select feasible and/or infeasible elements of respective iterate lists L_F^k and L_I^k as described in [17] and Algorithm 1. Define the current frame size parameter Δ^k according to the associated frame size parameters of the feasible incumbent element (x_F^k, Δ_F^k) and/or infeasible current incumbent element (x_I^k, Δ_I^k). Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$.

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

Search (optional) : Evaluate f and h at a finite set of points $S^k \subset \mathcal{X}$ on the mesh $M^k = \bigcup_{x \in V^k} \{x + \delta^k D z : z \in \mathbb{N}^{n_D}\}$. Set $L^{add} := \{(x, \Delta^k) : x \in S^k\}$.

If an improving or dominating success criterion is satisfied, the search may terminate. In this case, skip the poll and go to the parameter update step.

Poll : Select a positive spanning set $\mathbb{D}_\Delta^k$. Evaluate f and h on the poll set $P^k \subset M^k$ as defined in Subsection 4.3.2. Set $L^{add} := L^{add} \cup \left\{(x, \Delta^k) : x \in P^k\right\}$. If an improving or dominating criterion is satisfied, the poll may terminate opportunistically.

Parameter update : Define V^{k+1} as the union of V^k and all new candidates evaluated in $\mathcal{X}$ during the search and the poll. Classify the iteration as dominating, improving or unsuccessful. Update $h_{\max}^{k+1}$ according to Section 4.3.3. Remove points above the threshold from L_I^k. Update iterate lists L_F^{k+1} and/or L_I^{k+1} by adding new non-dominated points from L^{add} with their updated associated frame center $\Delta \in \{\Delta^k, \tau^{-1}\Delta^k\}$, as explained in Section 4.3.3. Remove new dominated points from L_F^k and/or L_I^k.

If the iteration is unsuccessful, replace (if they exist) the frame center elements (x_F^k, Δ_F^k) and (x_I^k, Δ_I^k) respectively by (x_F^k, Δ^{k+1}), (x_I^k, Δ^{k+1}) with $\Delta^{k+1} := \tau\Delta^k$.

end for

Figure 7: A summary of the DMulti-MADS-PB algorithm, inspired by [17].

Remark. When $m = 1$, the classification of the different type of iterations used in the DMulti-MADS-PB context is equivalent to the one used for the MADS-PB algorithm [8]. There also exists many configurations of iteration classifications criteria such that the generalization of the MADS-PB algorithm for multiobjective optimization and the convergence properties still hold. For example, one can declare an iteration as dominating when a trial point changes the set I^k. Practically, not all of them have the same performance. The definitions used below correspond to the most efficient variant observed on some preliminary experiments.

4.3.4 A frame center selection rule for the DMulti-MADS-PB algorithm

The constrained single-objective MADS-PB algorithm uses a classification of its two frame centers to practically improve the performance of the poll step. More precisely, the two frame centers are ordered, based on their objective values into *primary* and *secondary* poll centers. MADS-PB concentrates more efforts (based on the number of poll directions) on the primary poll center than the secondary poll center [8]. Inspired by this strategy, this subsection proposes an extension of the so-called *frame center selection rule* to constrained multiobjective optimization.

As in the single-objective case, DMulti-MADS-PB executes the poll around at least two frame centers. When $L_F^k = \emptyset$ or $L_I^k = \emptyset$, there is only one frame center, designed as the *primary frame center*. A complete set of poll points can be evaluated based on a positive spanning set $\mathbb{D}_\Delta^k$ composed of at least of $n+1$ directions (more details for the construction of $\mathbb{D}_\Delta^k$ can be found in [1, 11]).

When L_F^k and L_I^k are both non-empty, polling is done around a feasible and an infeasible frame centers. DMulti-MADS-PB orders these two frame centers into a *primary* frame center and a *secondary* frame center. This ordering is based on an user-supplied parameter $\rho > 0$, called the *frame trigger parameter*.

Recall that if L_F^k and L_I^k are nonempty, the selection of the infeasible frame center is done based on the $\psi_{L_F^k} : \mathcal{X} \to \mathbb{R}$ function parametrized by L_F^k, defined in Section 4.3.2. The following frame center selection rule is then proposed.

Definition 4.4 (frame center selection rule)**.** Let $\rho > 0$ provided by the user and suppose that $L_F^k \neq \emptyset$ and $L_I^k \neq \emptyset$. Let $(x_F^k, \Delta_F^k) \in L_F^k$ the feasible current incumbent and $(x_I^k, \Delta_I^k) \in L_I^k$ the infeasible current incumbent. If $\psi_{L_F^k}(x_I^k) - \rho\, \xi(L_F^k) > 0$, where $\xi(L_F^k)$ is given by

$$\xi(L_F^k) = \sum_{i=1}^{m} \mu \left(\max_{(x,\Delta)\in L_F^k} f_i(x), \min_{(x,\Delta)\in L_F^k} f_i(x) \right)$$

with $\mu : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ defined as

$$\mu(a,b) = \begin{cases} |a-b| & \text{if } a \neq b, \\ |a| & \text{otherwise;} \end{cases}$$

then the primary poll center is chosen as x_I^k and the secondary poll center is chosen as x_F^k, otherwise the primary poll center is chosen as x_F^k and the secondary poll center is chosen as x_I^k.

As for the single-objective MADS-PB algorithm, DMulti-MADS-PB puts more effort on the primary frame center than on the secondary frame center. The implementation of the poll strategy in this work follows the one developed in [11]: $n+1$ directions are used for the primary frame center and 2 directions for the secondary frame center by taking the negative of the first one.

If there exists at least one element $(x, \Delta) \in L_F^k$ such that $f_i(x) \leq f_i(x_I^k)$ for $i = 1, 2, \ldots, m$, then x_F^k will be chosen as the primary poll center. Figure 8 illustrates the zone in the biobjective space where I^k elements must be to be considered as potential primary poll centers. One could hope that putting more effort on the infeasible frame center in this case should enable it to reach a better part of the feasible decision region [8].

Remark. If $m = 1$, the frame center selection rule is equivalent to $f(x_I^k) < f(x_F^k) - \rho|f(x_F^k)|$, used in [5]. In this work, this rule was privileged to the original one in [5] $f(x_F^k) - f(x_I^k) > \rho$, as it takes into account the scale of the objective function values. A corresponding frame center selection rule extension to multiobjective optimisation would have been $\psi_{L_F^k}(x_I^k) - m\rho > 0$.

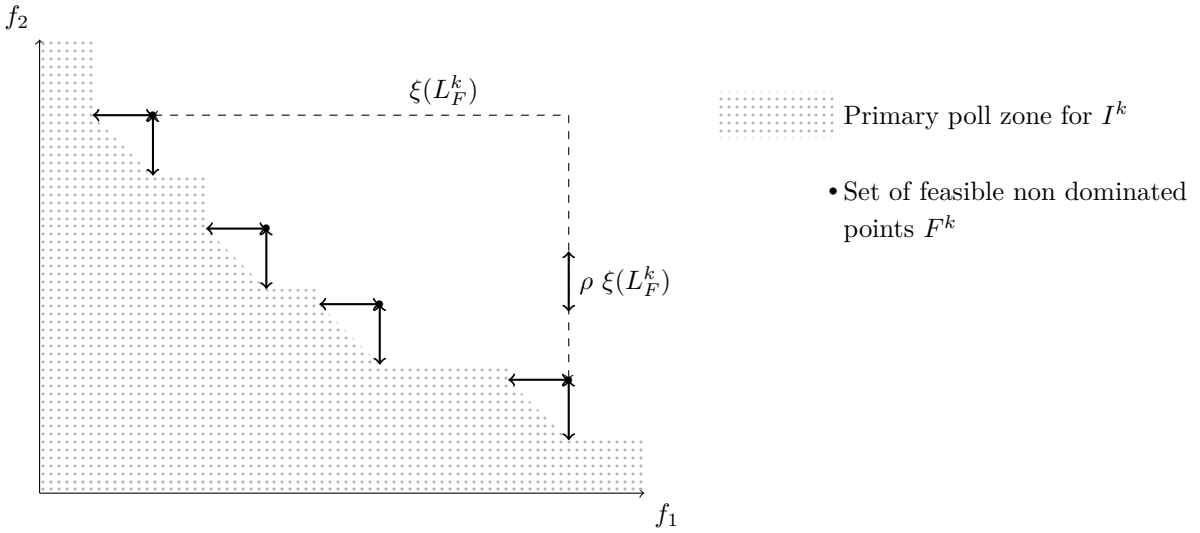


Figure 8: Representation of the selection of I^k frame incumbent as primary poll in the objective space for a biobjective minimization problem.

5 Convergence analysis

This section is devoted to the convergence analysis of the DMulti-MADS-TEB and DMulti-MADS-PB algorithms, inspired by [8, 17]. This work makes use of the following assumptions, taken from [8]:

Assumption 5.1. *There exists some point x^0 in the user-provided set V^0 such that $x^0 \in \mathcal{X}$ and $f(x^0)$ and $h(x^0)$ are both finite.*

Assumption 5.2. *All iterates considered by the algorithm lie in a bounded set.*

If Assumption 5.1 is not satisfied, DMulti-MADS cannot start. Assumption 5.2 is ensured if one imposes the existence of a bounded set in $\mathbb{R}^n$ containing V^k for all $k \in \mathbb{N}$. As V^k for $k \in \mathbb{N}$ is always composed of points satisfying the unrelaxable constraints, it is sufficient to guarantee that the set of unrelaxable constraints is itself a bounded set. For example, many engineering problems possess bound variables constraints, which cannot be violated.

As in single-objective MADS-PB algorithm [8], combining assumptions 5.1 and 5.2 and the structure of the mesh M^k enables to show that $\liminf_{k\to+\infty} \Delta^k = \liminf_{k\to+\infty} \delta^k = 0$ (see for example [26, Theorem A.1]). The classical convergence analysis of direct search methods focuses on subsequences of generated frame centers for which corresponding mesh size and frame size parameters converge to zero. The following notations and definitions are adapted from [8].

Let $\mathbb{U} \subset \mathbb{N}$ the set of unsuccessful iterations indexes. The poll generates one or several trial points around at least one of the two feasible and infeasible incumbents. If $k \in \mathbb{U}$ and the poll is executed around the feasible current frame center x_F^k, this last one is designed as a *feasible minimal frame center*. Otherwise, if $k \in \mathbb{U}$ and the poll is executed around the infeasible current frame center x_I^k, this last one is designed as an *infeasible minimal frame center*. From the rest of this work, these subsequences of frame centers are investigated separately. Note that for the DMulti-MADS-TEB variant, studying a

subsequence of infeasible minimal frame centers means that the algorithm does not manage to find a feasible point.

Definition 5.1. A subsequence $\{x^k\}_{k\in K}$ of DMulti-MADS frame centers, for some infinite subset of indexes $K \subseteq \mathbb{U}$ 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 is called a *refining point*.

Definition 5.2. Given a corresponding refining subsequence $\{x^k\}_{k\in K}$ and its refining point $\hat{x}$, a direction d is said to be a *refining direction* if and only if there exists an infinite subset of indexes $K' \subseteq K$ such that $d^k \in \mathbb{D}_\Delta^k$ with $x^k + \delta^k d^k \in \mathcal{X}$ and $d = \lim_{k\in K'} \frac{d^k}{\|d^k\|}$.

The convergence analysis also requires some mathematical tools from nonsmooth analysis. The following definitions are taken from [8].

Definition 5.3. 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 converge 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 cones to Ω at x is the Clarke tangent cone to Ω at x denoted as $T_\Omega^{Cl}(x)$. The DMulti-MADS analysis in a general constrained optimization context makes use of the hypertangent cone [48], which is the interior of the Clarke tangent cone, defined as:

$$T_\Omega^H(x) = \{d \in \mathbb{R}^n : \exists\ \epsilon > 0 \text{ such that } y + tw \in \Omega, \text{ for all } y \in \Omega \cap B_\epsilon(x), w \in B_\epsilon(d), \text{ and } 0 < t < \epsilon\}$$

where $B_\epsilon(x)$ is the open ball of radius $\epsilon > 0$ centered at x.

The DMulti-MADS analysis also requires that the objective function f is locally Lipschitz continuous in $\mathcal{X}$, i.e. each of its components f_i for $i = 1, 2, \ldots, m$ is locally Lipschitz continuous in $\mathcal{X}$. If this condition is satisfied, the Clarke-Jahn generalized derivatives [21] of f_i at $x \in \mathcal{X}$ in the direction $d \in \mathbb{R}^n$ exist and are defined by

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

This work can then introduce the main stationary conditions.

Definition 5.4. Let f be Lipschitz continuous near a point $\hat{x} \in \Omega$. $\hat{x}$ is a Pareto-Clarke critical point of f in Ω if for all directions $d \in T_\Omega^{Cl}(\hat{x})$, there exists $i(d) \in \{1, 2, \ldots, m\}$ such that $f_i^o(d)(\hat{x}; d) \geq 0$.

With the additional assumption that f is equally strictly differentiable at $\hat{x}$ (i.e. the corresponding Clarke generalized is a singleton containing only the gradient of one objective component at $\hat{x}$), the previous stationary result can be reformulated.

Definition 5.5. Let f be strictly differentiable at a point $\hat{x} \in \Omega$. $\hat{x}$ is a Pareto-Clarke-KKT critical point of f in Ω if for all directions $d \in T_\Omega^{Cl}(\hat{x})$, there exists $i(d) \in \{1, 2, \ldots, m\}$ such that $\nabla f_{i(d)}(\hat{x})^\top d \geq 0$.

As in the single-objective case [8], this work divides the convergence analysis into two cases: the study of subsequences of feasible minimal frame centers and the study of subsequences of infeasible minimal frame centers. For each case, the following methodology is used:

1. Prove that a subsequence of mesh size parameters and frame size parameters converges to zero.
2. Determine a particular subsequence of iterate points associated to the previous subsequence of parameters, i.e. a so-called *refined subsequence*.
3. This subsequence of iterate points converges to a refined point. Prove that this point satisfies some stationary properties.

5.1 Feasible case: results for f

As in [17], one wants to show that starting from a set of feasible points, DMulti-MADS produces at the limit locally stationary points for the constrained multiobjective optimization problem. To do that, this work proves the existence of finer refining subsequences, as it is done in [17]. The following analysis is a summary of the convergence analysis developed in [17] and covers the two variants DMulti-MADS-TEB and DMulti-MADS-PB.

Theorem 5.1. *Let Assumptions 5.1 and 5.2 hold and suppose DMulti-MADS generates a sequence of feasible iterates lists* $\{L_F^k\}$ *with* $L_F^k = \{(x^j, \Delta^j), j = 1, 2 \ldots, |L_F^k|\}$. *Then*

$$\lim_{k\to+\infty} \inf \delta_{\max}^k = \lim_{k\to+\infty} \inf \Delta_{\max}^k = 0.$$

Proof. Combining assumptions 5.1, 5.2, and the selection criterion of the feasible frame center with the structure of the mesh has been shown to be enough to ensure $\lim_{k\to+\infty} \inf \delta_{\max}^k = \lim_{k\to+\infty} \inf \Delta_{\max}^k = 0$ (see [17, Theorem 5.1] for more details). □

This work wants to prove the convergence of specific elements of the feasible iterate list generated by DMulti-MADS to stationary points. The concept of a feasible linked list, adapted from [17, 45], is then introduced.

Definition 5.6. Suppose DMulti-MADS generates the sequence of feasible iterate lists $\{L_F^k\}_{k\geq k_0}$ with $L_F^k = \{(x^l, \Delta^l), l = 1, 2, \ldots, |L_F^k|\}$ and $k_0 \in \mathbb{N}$ the iteration index such that $k_0 \in \arg\min \left\{k \in \mathbb{N} : F^k \neq \emptyset\right\}$. A *feasible linked sequence* is defined as a sequence $\{(x^{l_k}, \Delta^{l_k})\}$ such that there exists an iteration index $\ell_0 \geq k_0$ such that for any $k = \ell_0 + 1, \ell_0 + 2, \ldots$, the pair $\left\{(x^{l_k}, \Delta^{l_k})\right\} \in L_F^k$ is generated at iteration $k-1$ of DMulti-MADS from the pair $(x^{l_{k-1}}, \Delta^{l_{k-1}}) \in L_F^{k-1}$.

For the DMulti-MADS-PB variant algorithm, the following cases can occur:

1. Dominating iteration: either the algorithm generates at least one point which dominates the feasible frame center x_F^{k-1}, or it generates some infeasible points which have triggered the dominating success condition in the infeasible case.
   - $\forall (x^{l_k}, \Delta^{l_k}) \in L_F^k \setminus L_F^{k-1}$,
     $$x^{l_k} = x^{k-1} + \delta^{k-1} D z^{k-1} \text{ for some } z^{k-1} \in \mathbb{N}^{n_D} \text{ and } \Delta^{l_k} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}$$
     with $x^{k-1} \in \{x_F^{k-1}, x_I^{k-1}\}$.
   - $\forall (x^{l_k}, \Delta^{l_k}) \in L_F^k \cap L_F^{k-1}$,
     $$x^{l_k} = x^{l_{k-1}} \text{ and } \Delta^{l_k} = \Delta^{l_{k-1}}.$$
2. Improving iteration: the algorithm may generate some new feasible non-dominated points without dominating the feasible frame incumbent.

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

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

  with $x^{k-1} \in \{x_F^{k-1}, x_I^{k-1}\}$.
- $\forall (x^{l_k}, \Delta^{l_k}) \in L_F^k \cap L_F^{k-1}$,

$$x^{l_k} = x^{l_{k-1}} \text{ and } \Delta^{l_k} = \Delta^{l_{k-1}}.$$

3. Unsuccessful iteration: the algorithm may generate some new feasible non-dominated points without dominating the feasible frame incumbent.
   - $\forall (x^{l_k}, \Delta^{l_k}) \in L_F^k \setminus L_F^{k-1}$,

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

     with $x^{k-1} \in \{x_F^{k-1}, x_I^{k-1}\}$.
   - $\forall (x^{l_k}, \Delta^{l_k}) \in (L_F^k \cap L_F^{k-1}) \setminus \{(x_F^{k-1}, \Delta_F^{k-1})\}$,

$$x^{l_k} = x^{l_{k-1}} \text{ and } \Delta^{l_k} = \Delta^{l_{k-1}}.$$

   - $\forall (x^{l_k}, \Delta^{l_k}) \in \{(x_F^{k-1}, \Delta_F^{k-1})\}$,

$$x^{l_k} = x_F^{k-1} \text{ and } \Delta^{l_k} = \tau \Delta^{k-1}.$$

Similar relations can be drawn for the DMulti-MADS-TEB variant algorithm : note that for all $k > k_0$, no point at iteration k can be generated from an infeasible point at iteration $k-1$.

One can then prove that feasible linked sequences contain a feasible refining subsequence. The original proof can be found in [17], but for better understanding, it is restated below.

Theorem 5.2. *Let assumptions 5.1 and 5.2 hold and suppose DMulti-MADS generates the sequence of feasible iterate lists $\{L_F^k\}_{k \geq k_0}$ with $L_F^k = \{(x^l, \Delta^l), l = 1, 2, \ldots, |L_F^k|\}$ and $k_0 \in \mathbb{N}$ the iteration index such that $k_0 \in \arg\min \left\{k \in \mathbb{N} : F^k \neq \emptyset\right\}$. Then every feasible linked sequence $\{(x^{l_k}, \Delta^{l_k})\}$ contains a refining subsequence $\{x^{l_k}\}_{k \in K}$ for some infinite subset of indexes $K \subset \mathbb{U}$.*

Proof. $\forall k \geq k_0$, $0 \leq \Delta^{l_k} \leq \Delta_{\max}^k$. By combining Theorem 5.1 and the squeeze theorem, one gets

$$\lim_{k \to +\infty} \inf \Delta^{l_k} = \lim_{k \to +\infty} \inf \Delta_{\max}^k = 0,$$

which implies by definition the existence of a refining feasible subsequence within $\{(x^{l_k}, \Delta^{l_k})\}$. □

The analysis which follows is similar to [17].

Theorem 5.3. *Let assumptions 5.1 and 5.2 hold and suppose DMulti-MADS generates a feasible refining subsequence $\{x_F^k\}_{k \in K}$, with $x_F^k \in F^k$, converging to a refining point $\hat{x}_F \in \Omega$. Assume that f is Lipschitz continuous near $\hat{x}_F$. If $d \in T_\Omega^H(\hat{x}_F)$ is a refining direction for $\hat{x}_F$, then there exists an objective index $i(d) \in \{1, 2, \ldots, m\}$ such that $f_{i(d)}^o(\hat{x}_F; d) \geq 0$.*

Proof. Let $\{x_F^k\}_{k\in K}$, with $x_F^k \in F^k$, be a refining subsequence converging to a feasible refining point $\hat{x}_F \in \Omega$ and $d = \lim_{k\in K'} \frac{d^k}{\|d^k\|} \in T_\Omega^H(\hat{x}^F)$ a refining direction for $\hat{x}_F$, where $K' \subseteq K$ is an infinite subsequence of some infinite subset of unsuccessful iteration indexes, with poll directions $d^k \in \mathbb{D}_\Delta^k$ such that $x_F^k + \delta^k d \in \Omega$. Denote by $\nu \geq 0$ the Lipschitz constant of f near $\hat{x}_F$.

Then, for $i \in \{1, 2, \ldots, m\}$, the inequality

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

is satisfied.

$\{x_F^k\}_{k\in K}$ being a refining subsequence, the infinite subset of indexes $K' \subseteq K$ corresponds to unsuccessful iterations. Consequently, the point $x_F^k + \delta^k d \in \Omega$ does not dominate x_F^k. One can then find an infinite subsequence of indexes $K'' \subset K'$ such that there exists an index $i(d) \in \{1, 2, \ldots, m\}$ satisfying

$$f_{i(d)}(\hat{x}_F; d) \geq \limsup_{k\in K''} \frac{f_{i(d)}(x_F^k + \delta^k d) - f_{i(d)}(x_F^k)}{\delta^k \|d^k\|} \geq 0.$$

□

When the set of refining directions is dense in a non-empty hypertangent cone at Ω, Pareto Clarke stationarity is ensured, similarly to the analysis conducted in [17, 26].

Theorem 5.4. *Let assumptions 5.1 and 5.2 and suppose DMulti-MADS generates a feasible refining subsequence $\{x_F^k\}_{k\in K}$, with $x_F^k \in F^k$, converging to a refining point $\hat{x}_F \in \Omega$. Assume that f is Lipschitz continuous near $\hat{x}_F$ and $T_\Omega^H(\hat{x}_F) \neq \emptyset$. If the set of refining directions is dense for $\hat{x}_F$ in $T_\Omega^{Cl}(\hat{x}_F)$, then $\hat{x}_F$ is a Pareto-Clarke critical point of (MOP).*

Proof. The authors in [7] prove than for any direction v in the Clarke tangent cone,

$$f_i^0(\hat{x}_F; v) = \lim_{\substack{d \in T_\Omega^H(\hat{x}_F) \\ d \to v}} f_i^o(\hat{x}_F; d) \text{ for } i = 1, 2, \ldots, m.$$

By hypothesis, the set of refining directions is dense for $x_F \in \Omega$ in $T_\Omega^{Cl}(\hat{x}_F)$. Then there exists a sequence of refining directions $\{d_r\}_{r\in R} \in T_\Omega^H$ for $\hat{x}_F$ such that $\lim_{r\in R} d_r = v$. Since the number of components of the objective function is finite, one can find a subsequence $\{d_r\}_{r\in R'}$ with $R' \subseteq R$ such that $v = \lim_{r\in R'} d_r$ and $f_{i(v)}(\hat{x}_F; v) \geq 0$ by Theorem 5.3 for all indexes $r \in R'$. Passing at the limit concludes the proof. □

5.2 Infeasible case: results for h

In this subsection, the goal is to analyse refining subsequences of infeasible points according to the h violation function as in [8] for the single-objective constrained case. Two cases can occur. The refining point $\hat{x}_I$ of an infeasible refining subsequence satisfies $h(x_I) = 0$. In this case, it means that the feasible set is non-empty, and that $\hat{x}_I$ is a global minimum for the single-objective problem $\min_{x \in \mathcal{X}} h(x)$. Otherwise, this work proves than $\hat{x}_I$ satisfies some stationarity results relatively to h. Note that the DMulti-MADS-TEB variant generates an infeasible refining subsequence if and only if it starts from an infeasible point belonging to V^0 and generates no feasible points along the iterations.

Contrary to the feasible case, this work does not characterize particular infeasible sequences of points within the sequence of infeasible frame incumbents.

Theorem 5.5. *Let assumptions 5.1 and 5.2 hold and suppose DMulti-MADS generates a refining subsequence $\{x_I^k\}_{k \in K}$, with $x_I^k \in I^k$, converging to a refining point $\hat{x}_I \in X$. Assume that h is Lipschitz continuous near $\hat{x}_I$. If $d \in T_{\mathcal{X}}^H(\hat{x}_I)$ is a refining direction for $\hat{x}_I$, then $h^o(\hat{x}_I; d) \geq 0$.*

Proof. The proof is similar to that of 5.3, h and $\mathcal{X}$ playing respectively the roles of f_i for a fixed objective index $i \in \{1, 2, \ldots, m\}$ and Ω. □

The next theorem's proof is identical to 5.4.

Theorem 5.6. *Let assumptions 5.1 and 5.2 hold and suppose DMulti-MADS generates a refining subsequence $\{x_I^k\}_{k \in K}$, with $x_I^k \in I^k$, converging to a refining point $\hat{x}_I \in \mathcal{X}$. Assume that h is Lipschitz continuous near $\hat{x}_I$ and $T_{\mathcal{X}}^H(\hat{x}_I) \neq \emptyset$. If the set of refining directions is dense in $T_{\mathcal{X}}^{Cl}(\hat{x}_I)$, then $\hat{x}_I$ is a Clarke stationary point for*

$$\min_{x \in \mathcal{X}} h(x).$$

Proof. The authors in [7] prove than for any direction v in the Clarke tangent cone,

$$h^0(\hat{x}_I; v) = \lim_{\substack{d \in T_{\mathcal{X}}^H(\hat{x}_I) \\ d \to v}} h^o(\hat{x}_I; d).$$

By hypothesis, the set of refining directions is dense for $\hat{x}_I \in \mathcal{X}$ in $T_{\mathcal{X}}^{Cl}(\hat{x}_I)$. Then there exists a sequence of refining directions $\{d_r\}_{r \in R} \in T_{\mathcal{X}}^H$ for $\hat{x}_I$ such that $\lim_{r \in R} d_r = v$. By Theorem 5.5, for all $r \in R$, $h^o(\hat{x}_I; d) \geq 0$. Passing at the limit concludes the proof. □

6 Computational experiments

This section is devoted to the computational experiments of DMulti-MADS on constrained multiobjective problems. The first part presents the considered solvers. The second part is dedicated to the comparison of all solvers and DMulti-MADS variants on a set of analytical problems using data profiles for multiobjective optimization [17]. The last part shows comparison of solvers on "real" engineering problems using convergence profiles.

To assess the performance of different algorithms, this work relies on the use of data profiles for multiobjective optimization [17] and convergence profiles. Both tools require the definition of a convergence test for a given computational problem, based on the hypervolume indicator [53].

The hypervolume indicator represents the volume of the objective space dominated by a Pareto front approximation Y_N and delimited from above by a reference point $r \in \mathbb{R}^m$ such that for all $y \in Y_N$,

$y_i < r_i$ for $i = 1, 2, \ldots, m$. The hypervolume possesses many useful properties : Pareto compliant with the dominance ordering, it can capture many properties of a Pareto front approximation as spread, cardinality, convergence to the Pareto front, or extension [4, 43].

The convergence test requires a Pareto front approximation reference Y^p for a given problem $p \in \mathcal{P}$, where $\mathcal{P}$ is the set of considered problems, from which the approximated ideal objective vector

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

and the approximated nadir objective vector

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

are extracted, with m_p the number of objectives of problem $p \in \mathcal{P}$. Y^p is constructed using the set of best non dominated points found by all algorithms on problem $p \in \mathcal{P}$ for a maximal budget of evaluations.

Assuming Y^e is a Pareto front approximation generated after e evaluations by a given deterministic solver for problem p, a scaling and translating transformation is applied to this last one defined by: $\forall y \in Y^e \cup Y^p \cup \{\tilde{y}^{N,p}\}$,

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

where $\oslash$ is the element wise-divisor operator. Note that this transformation conserves the dominance order relation. The computational problem is said to be solved by the algorithm with tolerance $\varepsilon_\tau > 0$ if

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

where $HV(Y_N, r)$ is the hypervolume indicator value of the volume dominated by the Pareto front approximation Y_N and delimited above by the reference point $r \in \mathbb{R}^m$. All elements of Y_N which are dominated by $r \in \mathbb{R}^m$ are removed during the computation of the hypervolume indicator. If no element of Y_N dominates r, then $HV(Y_N, r) = 0$.

Data profiles show the proportion of all computational problems solved by an algorithm in function of the number of groups of $n+1$ evaluations required to build a gradient simplex in $\mathbb{R}^n$. In these experiments, stochastic solvers are also considered. In this case, data profiles are modified to take into account their performance variability, as described in [17].

6.1 Tested solvers and variants of DMulti-MADS

The following constrained solvers are considered:

- the deterministic solver `NOMAD` [40] which implements the BiMADS algorithm (Bi-objective Mesh Adaptive Direct Search) [13] tested only for $m = 2$ objectives - `www.gerad.ca/nomad/`;
- the deterministic solver DFMO (Derivative-Free Multi Objective) [45] - `http://www.iasi.cnr.it/~liuzzi/DFL/`;
- the stochastic heuristic solver NSGA-II (Non Dominating Sorting Algorithm II) [27]; a constrained version is implemented in the `Pymoo` Library [18] version `0.4.2.2` - `https://pymoo.org`.

For the BiMADS algorithm, two variants based on `NOMAD 3.9.1` are considered. The first uses the default settings of the MADS algorithm, detailed in [11, 15, 12, 23]. The second deactivates models and other heuristics such that BiMADS relies only on the MADS algorithm with $n+1$ directions, a speculative search and an opportunistic polling strategy, for a fairer comparison with DMulti-MADS. DFMO and NSGA-II are used with their default settings. NSGA-II uses an initial population with 100 elements.

In these experiments, this work considers another variant of DMulti-MADS for constrained multiobjective optimization based on the penalty approach used in [45]. More specifically, given the constrained multiobjective problem (MOP), the authors of [45] introduce the following penalty functions

$$Z_i(x;\varepsilon) = f_i(x) + \frac{1}{\epsilon}\sum_{j\in\mathcal{J}} \max\{0, c_i(x)\},\ i = 1, 2, \ldots, m$$

where $\epsilon > 0$ is an external parameter and consider the following multiobjective problem

$$(MOP_p) : \min_{x\in\mathcal{X}} Z(x) = (Z_1(x;\epsilon), Z_2(x;\epsilon), \ldots, Z_m(x;\epsilon))^\top .$$

The DMulti-MADS-Penalty variant uses the DMulti-MADS-TEB variant on the modified (MOP_p) multiobjective problem. The external parameter $\epsilon > 0$ is set to the default value proposed by [45]. Note that this approach has already been used by these authors to compare DMS (which cannot start from infeasible points) and DFMO on constrained multiobjective problems [45]. As the first strategy proposed to handle constraints with convergence results, it is natural to see if this approach performs well compared to the two new variants proposed in this paper.

For all constrained variants of DMulti-MADS, a speculative search strategy is implemented as in [17] for one or both feasible and infeasible current incumbents if they exist, combined with a polling strategy with $n+1$ directions [11]. The implementation of the mesh follows a granular mesh strategy [12]. All variants stop as soon as one component of the mesh size vector is below 10^{-9} or after running out of evaluations budget. All variants use an opportunistic strategy: as soon as a new candidate dominates at least one current incumbent, the iteration stops. All variants also apply a spread selection with parameter value $w^+ = 1$. For the DMulti-MADS-PB variant, the trigger parameter is set to $\rho = 0.1$. When DMulti-MADS-TEB switches from the first phase to the second phase, the frame and mesh size parameters of the generated feasible points are not resettled to their respective initial values Δ^0 and δ^0.

Finally, the implementation of the progressive barrier in `NOMAD 3.9.1` for the BiMADS algorithm diverges from the description given in [8] for efficiency gains. The DMulti-MADS-PB algorithm variant equally incorporates these modifications for a fairer comparison with the implementation of `NOMAD 3.9.1`. Precisely, the threshold $h^k_{\max}$ is updated according to the set $U^{k+1} \subseteq V^{k+1}$, which enables it to decrease faster. Furthermore, in the implementation, an iteration k is considered as improving if the algorithm generates a point $x^t \in V^{k+1} \setminus V^k$ satisfying improving conditions. Note that the convergence properties for infeasible refining subsequences still hold. However, it may exist a point $x \in \cup_{k\in\mathbb{N}} V^k$ with $0 < h(x) < h(\hat{x}^I)$ where $\hat{x}^I \in \mathcal{X}$ is an infeasible refining point.

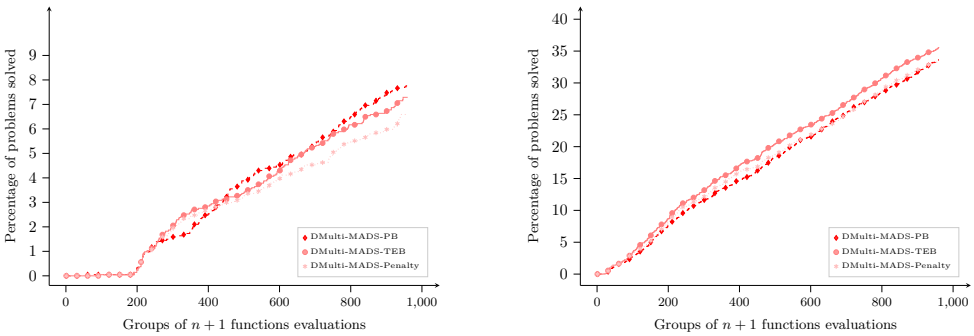
The code used for experiments can be found at `https://github.com/bbopt/DMultiMadsPB`.

6.2 Comparing solvers on synthetic benchmarks

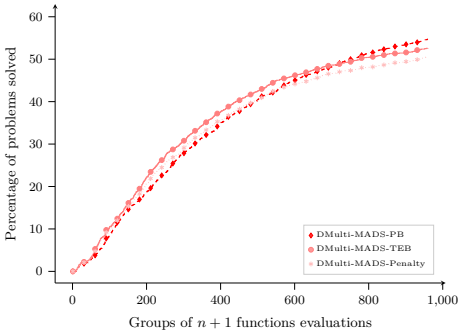
In this subsection, this work considers a set of 214 analytical multiobjective optimization problems proposed by [45], with $n \in [3, 30]$, $m \in \{2, 3, 4\}$ and $|\mathcal{J}| \in [3, 30]$. Among them, 103 problem possess $m = 2$ objectives.

In a first part, this work compares the three variants of DMulti-MADS on this set of problems. The three of them use a maximum budget of $30,000$ evaluations. For each problem, the three variants start

from the same set of initial points, using the linesearch strategy described in [26]. Each variant on each problem executes 10 replications by changing the random seed which controls the generation of polling directions.



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



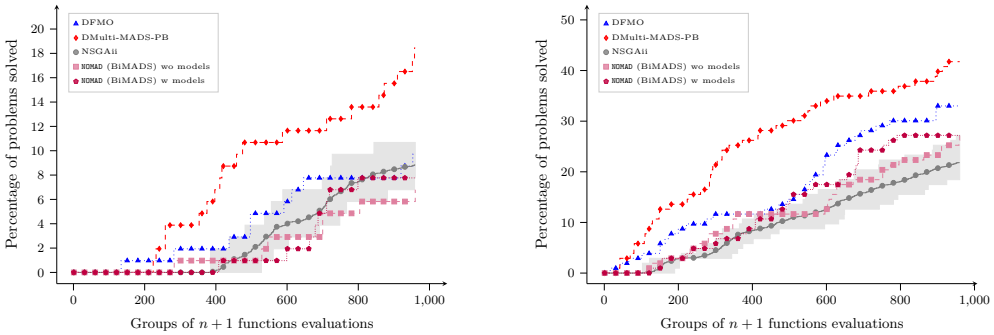
(c) $\varepsilon_\tau = 10^{-1}$

Figure 9: Data profiles obtained from 10 replications from 214 multiobjective analytical problems taken from [45] for DMulti-MADS-PB, DMulti-MADS-TEB and DMulti-MADS-Penalty with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

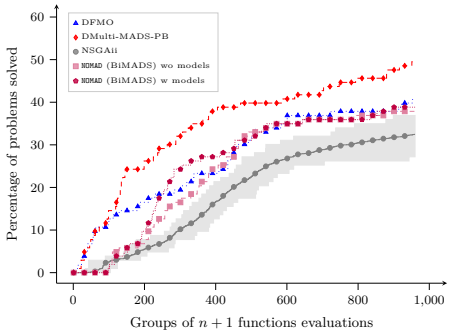
The data profiles given in Figure 9 show that for the three tolerance values considered, DMulti-MADS-Penalty solves slightly less problems than the two other variants introduced in this work. One can equally observe than DMulti-MADS-PB performs better for a medium to high budget of evaluations for the lowest tolerance $\varepsilon_\tau = 10^{-2}$. For the largest tolerance $\varepsilon_\tau = 10^{-1}$, DMulti-MADS-PB solves more problems for a high budget of evaluations. However, for medium tolerance, the performance of DMulti-MADS-PB is similar to DMulti-MADS-Penalty. A closer look at the considered problems shows than in this case, it is better to firstly look for feasible solutions than to explore the infeasible decision space. It then gives an advantage to DMulti-MADS-TEB over the two other variants. For the rest of this subsection, only DMulti-MADS-PB and DMulti-MADS-TEB are kept, as they are more performant.

For the comparison with the other algorithms, the same maximum budget of $30,000$ function evaluations is kept. Practically, for NSGA-II, the total number of population generations is fixed to 300, with a

fixed population size equal to 100. For each problem, the deterministic solvers start from the same initial points using the linesearch strategy [26]. For each problem, NSGA-II is run 30 times with different seeds to capture stochastic behavior and analyze its performance variation.



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



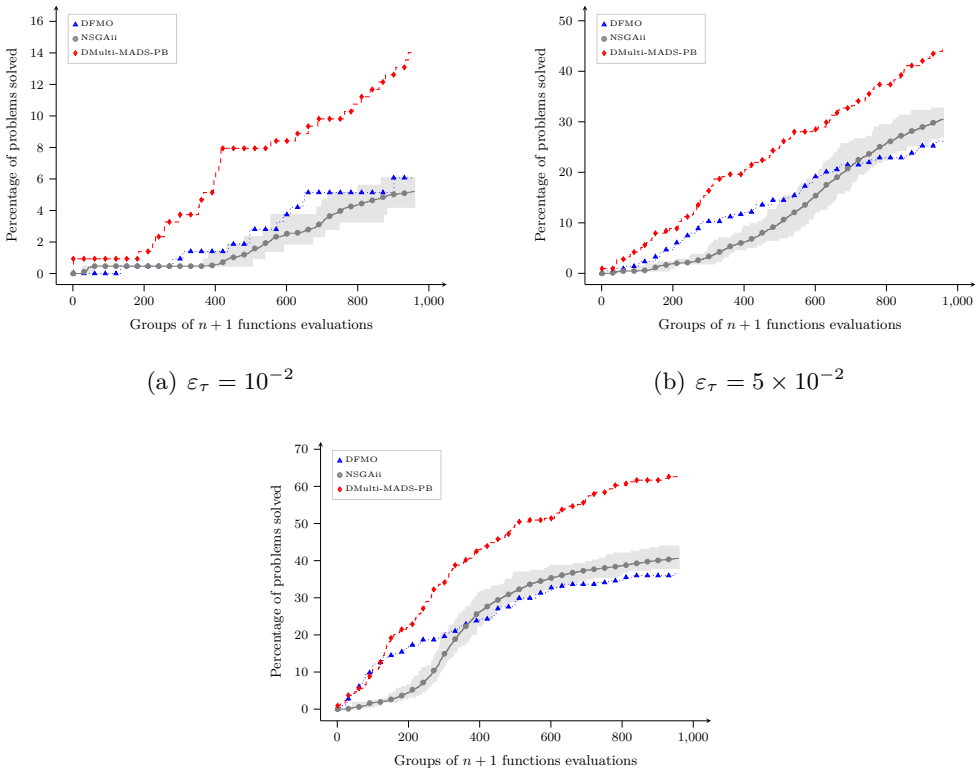
(c) $\varepsilon_\tau = 10^{-1}$

Figure 10: Data profiles using NOMAD (BiMADS), DFMO, DMulti-MADS-PB and NSGA-II obtained on 103 biojective analytical problems from [45] with 30 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

From Figures 10 and 11, one can see that DMulti-MADS-PB outperforms the other solvers on this set of analytical functions, for all tolerances considered. The same conclusions can be drawn for DMulti-MADS-TEB. From these figures, one can also observe than DFMO displays better performance on biojective problems than for the whole set (see Figure 10).

6.3 Comparing solvers on real engineering benchmarks

In this subsection, this work considers three multiobjective optimization problems: the biobjective SOLAR8 and SOLAR9 design problems and the triobjective STYRENE design problem [3, 14]. These three applications are more costly to solve than the analytical benchmarks considered in the previous subsection. The use of data profiles to compare solvers on these problems is then difficult to put into practice.



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

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

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

Figure 11: Data profiles using DFMO, DMulti-MADS-PB and NSGA-II obtained on 214 multiobjective analytical problems from [45] with 30 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5\times10^{-2}, 10^{-1}\}$.

To assess the performance of solvers on these problems, an adaptation of convergence profiles (see [10, Appendix A] for a description) to multiobjective optimization is proposed. Convergence profiles for multiobjective optimization make use of the normalized hypervolume value, presented at the beginning of Section 6 and given by:

$$\frac{HV\left(T(Y^e), T(\tilde{y}^{N,p})\right)}{HV(T(Y^p), T(\tilde{y}^{N,p}))}$$

where Y^p is the Pareto front approximation reference for problem p, Y^e the Pareto front approximation generated after e evaluations by a given solver on an instance of problem p, T a scaling and translating transformation applied and $\tilde{y}^{N,p}$ the approximated nadir objective vector of Y^p.

Convergence profiles for multiobjective optimization on a given problem p visualize the evolution of the normalized hypervolume indicator for a given solver against the number of evaluations used. Consequently, a normalized hypervolume value equal to 1 means that the solver has solved the problem p. A normalized hypervolume equal to 0 means that the solver has not generated points which dominate the approximated nadir objective vector of the Pareto front approximation reference.

6.3.1 Comparing solvers on the SOLAR8 and SOLAR9 design problems

SOLAR8 and SOLAR9 are two biobjective optimization problems derived from a numerical simulator coded in C++ of a solar plant with a molten salt heat storage system [36]. The simulation is composed of three steps. The heliostats field captures sun rays which are transmitted to a central cavity receiver. The sun energy is given to the thermal storage which converts it to thermal energy. This last one activates the powerblock, which triggers a steam turbine, generating electrical power output. Numerical simulations intervene all along the different phases of the process, which make it impossible to provide gradients. For more details, the reader can refer to [36]. The simulator can be found at `https://github.com/bbopt/solar`.

For the two considered problems, a blackbox evaluation can take more than 10 seconds (on a machine with 8 Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz 16G RAM). Experiments equally reveal the presence of hidden constraints. Tables 12 and 13 describe the objectives, constraints and starting points used for each problem.

Constraints/Objectives	Description of constraints and objectives
$-f_1$	Maximize heliostat field performance (absorbed energy)
f_2	Minimize cost of field, tower and receiver
Heliostat design constraints	Four constraints related to the dimensions of the heliostat field
Receiver constraints	Three constraints related to the design of the receiver
Energy constraints	Two constraints which depend on the energy production
Variables	Description and type
Heliostats field	Nine variables related to the dimensions of the heliostats field Eight real and one integer
Heat transfer loop	Four variables related to the design of the heat transfer system Three real and one integer
Starting point (infeasible)	$(11.0, 11.0, 200.0, 10.0, 10.0, 2650, 89.0, 0.5, 8.0, 36, 0.30, 0.020, 0.0216)$

Figure 12: Objectives, constraints, variables and starting point of the SOLAR8 problem.

SOLAR8 and SOLAR9 both possess integer decision variables. In the experiments, the only solver which can treat integer variables is `NOMAD` (BiMADS). Consequently, for the other solvers, all integer variables are fixed to their starting values along the optimization. For the SOLAR8 problem, three variants of `NOMAD` (BiMADS): two for which integer variables are fixed and one which treat mixed integer (MI) problems. For this last variant, the algorithmic parameters are chosen by default.

Remark. For SOLAR9, `NOMAD` (BiMADS) completely outperforms the other algorithms when it can modify integer variables. After investigation, this behaviour is not related to the performance of the algorithm, but the initial choice of the integer variables. However, for the sake of reproducibility, these values are kept.

All deterministic algorithms are allocated a maximal budget of $5,000$ evaluations and start from the same infeasible point for each problem. NSGA-II does not take starting points as arguments. To compare it with the others, NSGA-II is run 10 times to capture stochastic behaviour, with a population size fixed to 100 and a total number of generations equal to 50.

From Figure 14(a), one can see that DMulti-MADS-PB performs better than the other algorithms on SOLAR8. When looking at the Pareto front plottings (Figure 14(b)), one can note that DMulti-MADS-PB captures a portion of the Pareto front on the top left. DMulti-MADS-TEB is slightly better than DMulti-MADS-Penalty and compares well in terms of performance with `NOMAD` (BiMADS) when allowing

Constraints/Objectives	Description of constraints and objectives
f_1	Minimize production costs
$-f_2$	Maximize energy production
Heliostats design constraints	Four constraints related to the dimensions of the heliostat field
Heat storage constraints	Four constraints relative to the molten salt heat thermic/pressure storage system
Receiver design constraints	Two constraints which depend on the tube size and diameter receiver
Steam constraints	Five constraints related to steam temperature, power output and steam design.
Variables	**Description and type**
Heliostats field	Nine variables related to the dimensions of the heliostats field Eight real and one integer
Heat transfer loop	Nineteen variables related to the design of the heat transfer system Fourteen real and five integer
Powerblock	One variable: type of turbine; integer
Starting point (infeasible)	$(9.0, 9.0, 150.0, 6.0, 8.0, 1000, 45.0, 0.5, 5.0, 900.0,$ $9.0, 9.0, 0.30, 0.20, 560.0, 500, 0.30, 0.0165, 0.018, 0.017,$ $10.0, 0.0155, 0.016, 0.20, 3, 12000, 1, 2, 2)$

Figure 13: Objectives, constraints, variables and starting point of the SOLAR9 problem.

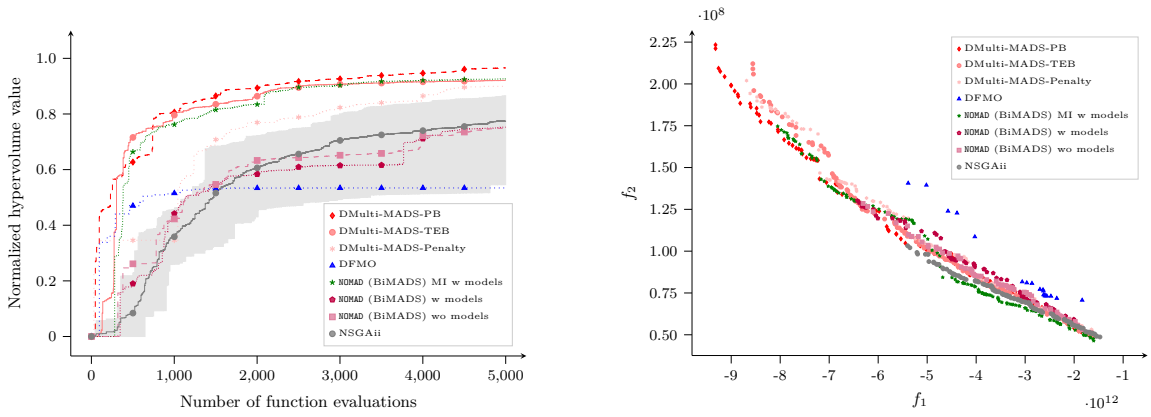


Figure 14: (a) On the left, convergence profiles for the SOLAR8 problem using DFMO, DMulti-MADS, NOMAD (BiMADS) and NSGA-II with 10 different runs of NSGA-II for a maximal budget of 5,000 evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of SOLAR8 for DFMO, DMulti-MADS, NOMAD (BiMADS) and an instance of NSGA-II in the objective space.

the use of mixed integer variables. DFMO does not perform well on this problem, due to the different scales on the constraints included in the penalty objective function, which impacts its efficiency.

Figure 15(a) shows the convergence profiles for the SOLAR9 problem for different solvers. On this problem, NOMAD (BiMADS) are the most efficient, even if DMulti-MADS-PB catches it for the last evaluations. As shown on Figure 15(b), the extent of the Pareto front approximation reference is low, which

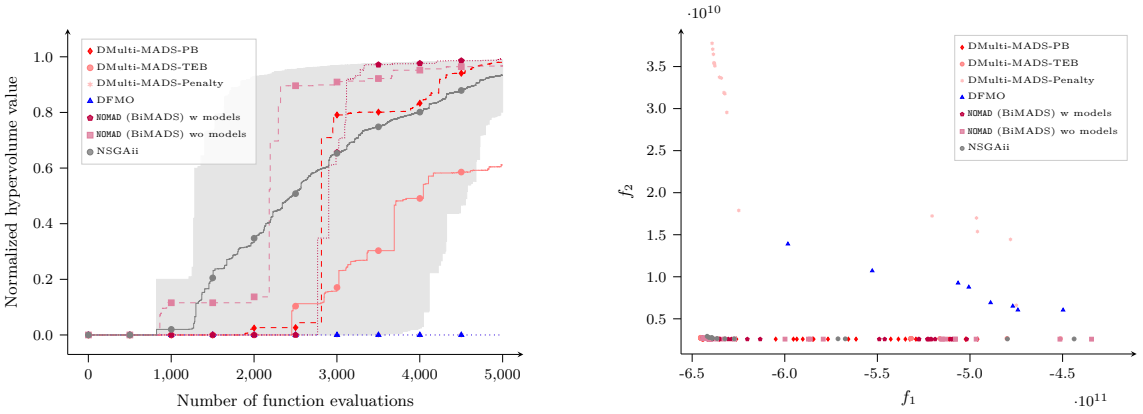


Figure 15: (a) On the left, convergence profiles for the SOLAR9 problem using DFMO, DMulti-MADS, NOMAD (BiMADS) and NSGA-II with 10 different runs of NSGA-II for a maximal budget of 5,000 evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of SOLAR9 for DFMO, DMulti-MADS, NOMAD (BiMADS) and an instance of NSGA-II in the objective space.

favours scalarization-based approaches such as BiMADS. This problem also illustrates the default of penalty-based approaches against other methods. As the constraint functions possess different amplitudes, the penalized optimization problem differs from the original, which explains why DFMO and DMulti-MADS-Penalty fail.

6.3.2 Comparing solvers on the STYRENE design problem

STYRENE is a triobjective optimization problem related to the production of styrene, as described in [3, 14]. Styrene production process is composed of four steps: reactants preparation, catalytic reactions, a first distillation to recover styrene and a second one to recover benzene. The second distillation equally involves the recycling of unreacted ethylbenzaline, reintroduced into the styrene production as an initial reactant. The proposed triobjective optimization problem, based on a numerical implementation coded in C++, aims at maximizing the net present value associated to the process (f_1), the purity of produced styrene (f_2), and the overall ethylbenzene conversion into styrene (f_3). This application possesses eight bounded variables, and nine general inequality constraints related to the chemical process (e.g. environmental regulations), or costs (e.g. investment). More details can be found in [14].

A simulation takes around 1 second to run, starting from a feasible point (on a machine with 8 Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz 16G RAM). This problem has hidden constraints. Even when starting from a feasible point, the simulation can sometimes fail to produce a finite numerical value.

A maximal budget of 20,000 evaluations is allocated for all deterministic solvers, which all start from the same feasible point. This experiment does not consider NOMAD (BiMADS), as it only treats biojective problems. NSGA-II is run 10 times, with a population size fixed to 100, and a maximal number of generations equal to 200.

Figure 16(a) shows the convergence profiles obtained for the STYRENE design triobjective problem. This figure shows that DMulti-MADS-PB performs better than the other solvers, followed by DMulti-MADS-TEB. From Figure 16(b), one can observe that DMulti-MADS-PB captures more parts of the

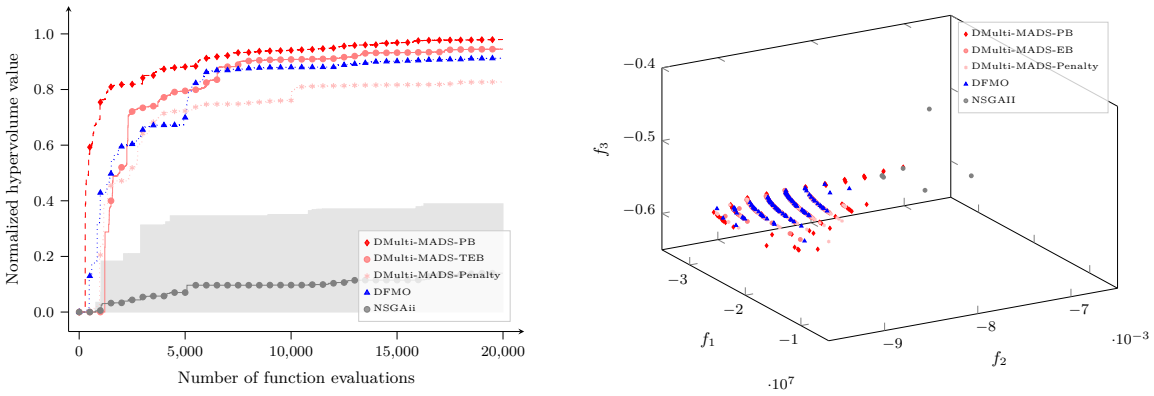


Figure 16: (a) On the left, convergence profiles for the STYRENE design problem using DFMO, DMulti-MADS, and NSGA-II with 10 different runs of NSGA-II for a maximal budget of 20,000 evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of STYRENE for DFMO, DMulti-MADS, and an instance of NSGA-II in the objective space.

Pareto front reference than all the other methods. Finally, even when taking into account variability, NSGA-II is less efficient than all the other solvers on this problem.

7 Discussion

This work proposes two extensions of the DMulti-MADS algorithm [17] to handle blackbox constraints, generalizing the works conducted for the single-objective MADS algorithm [8, 9]. It is proved that these two extensions possess the same convergence properties than DMulti-MADS [17] when studying feasible sequences generated by these two extensions. Convergence analysis for the infeasible case is also derived, as in [8].

Experiments show that these two variants are competitive comparing to other state-of-the-art methods, and more robust on real engineering applications than a penalty-based approach, as proposed in [45]. These experiments also reveal that a two-phase approach performs surprisingly well on blackbox multiobjective optimization problems, contrary to single-objective ones [9]. Future work involves the integration of surrogate methods into a search strategy [20, 23], and the use of parallelism. An integration in the `NOMAD` solver is also planned.

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] S.N. Alexandropoulos, C.K. Aridas, S.B. Kotsiantis, and M.N. Vrahatis. Multi-Objective Evolutionary Optimization Algorithms for Machine Learning: A Recent Survey. In I.C. Demetriou and P.M. Pardalos, editors, Approximation and Optimization: Algorithms, Complexity and Applications, pages 35–55, Cham, 2019. Springer.

[3] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. Journal of Global Optimization, 41(2):299–318, 2008.

[4] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon. Performance indicators in multiobjective optimization. European Journal of Operational Research, 292(2):397–422, 2021. Invited Review.

[5] C. Audet, A.R. Conn, S. Le Digabel, and M. Peyrega. A progressive barrier derivative-free trust-region algorithm for constrained optimization. Computational Optimization and Applications, 71(2):307–329, 2018.

[6] C. Audet and J.E. Dennis, Jr. A pattern search filter method for nonlinear programming without derivatives. SIAM Journal on Optimization, 14(4):980–1010, 2004.

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

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

[9] C. Audet, J.E. Dennis, Jr., and S. Le Digabel. Globalization strategies for Mesh Adaptive Direct Search. Computational Optimization and Applications, 46(2):193–215, 2010.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] F. Augustin and Y.M. Marzouk. NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints. Technical report, arXiv, 2014.

[17] J. Bigeon, S. Le Digabel, and L. Salomon. DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization. Computational Optimization and Applications, 79(2):301–338, 2021.

[18] J. Blank and K. Deb. Pymoo: Multi-Objective Optimization in Python. IEEE Access, 8:89497–89509, 2020.

[19] J. Branke, K. Deb, K. Miettinen, and R. Slowiński. Multiobjective optimization: Interactive and evolutionary approaches, volume 5252. Springer, 2008.

[20] C.P. Brás and A.L. Custódio. On the use of polynomial models in multiobjective directional direct search. Computational Optimization and Applications, 77(3):897–918, 2020.

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

[22] Y. Collette and P. Siarry. Multiobjective optimization: principles and case studies. Springer, 2011.

[23] 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.

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

[25] 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, 2018.

[26] 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.

[27] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, and Hans-Paul H.P. Schwefel, editors, Parallel Problem Solving from Nature PPSN VI, pages 849–858, Berlin, Heidelberg, 2000. Springer.

[28] S. Dedoncker, W. Desmet, and F. Naets. An adaptive direct multisearch method for black-box multi-objective optimization. Optimization and Engineering, 2021.

[29] F. Di Pierro, S.T. Khu, D. Savić, and L. Berardi. Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. Environmental Modelling & Software, 24(2):202–213, 2009.

[30] M.A. Diniz-Ehrhardt, J.M. Martinez, and L.G. Pedroso. Derivative-free methods for nonlinear programming with general lower-level constraints. Journal of Computational and Applied Mathematics, 30(1), 2011.

[31] H. Fang, M. Rais-Rohani, Z. Liu, and M.F. Horstemeyer. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. Computers & Structures, 83(25-26):2121–2136, 2005.

[32] P. Feliot, J. Bect, and E. Vazquez. A Bayesian approach to constrained single-and multi-objective optimization. Journal of Global Optimization, 67(1–2):97–133, 2017.

[33] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Mathematical Programming, Series A, 91:239–269, 2002.

[34] R. Fletcher, S. Leyffer, and Ph.L. Toint. On the global convergence of a filter–SQP algorithm. SIAM Journal on Optimization, 13(1):44–59, 2002.

[35] J. Fliege and A.I.F. Vaz. A method for constrained multiobjective optimization based on SQP techniques. SIAM Journal on Optimization, 26(4):2091–2119, 2016.

[36] M. Lemyre Garneau. Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers. Master's thesis, Polytechnique Montréal, 2015.

[37] S. Gratton and L.N. Vicente. A Merit Function Approach for Direct Search. SIAM Journal on Optimization, 24(4):1980–1998, 2014.

[38] T.G. Kolda, R.M. Lewis, and V. Torczon. A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints. Technical Report SAND2006-5315, Sandia National Laboratories, USA, 2006.

[39] J. Larson, M. Menickelly, and S.M. Wild. Derivative-free optimization methods. Acta Numerica, 28:287–404, 2019.

[40] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm. ACM Transactions on Mathematical Software, 37(4):44:1–44:15, 2011.

[41] S. Le Digabel and S.M. Wild. A Taxonomy of Constraints in Simulation-Based Optimization. Technical Report G-2015-57, Les cahiers du GERAD, 2015.

[42] M. Li and X. Yao. Dominance Move: A Measure of Comparing Solution Sets in Multiobjective Optimization. Technical Report 1702.00477, arXiv, 2017.

[43] 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.

[44] G. Liuzzi and S. Lucidi. A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function. SIAM Journal on Optimization, 20(1):1–29, 2009.

[45] 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.

[46] K. Miettinen. Nonlinear Multiobjective Optimization. Springer, 1999.

[47] R.G. Regis. On the properties of positive spanning sets and positive bases. Optimization and Engineering, 17(1):229–262, 2016.

[48] R.T. Rockafellar. Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming. SIAM Journal on Control, 12(2):268–285, 1974.

[49] S. Sharma and G.P. Rangaiah. Multi-Objective Optimization Applications in Chemical Engineering, chapter 3, pages 35–102. John Wiley & Sons, 2013.

[50] J. Yuan, H-L. Liu, Y-S. Ong, and Z. He. Indicator-based Evolutionary Algorithm for Solving Constrained Multi-objective Optimization Problems. IEEE Transactions on Evolutionary Computation, pages 1–1, 2021.

[51] 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.

[52] E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of Pareto set approximations. In Multiobjective Optimization, pages 373–404. Springer, Berlin, Heidelberg, 2008.

[53] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms – A comparative case study. In A.E Eiben, T. Bäck, M. Schoenauer, and H.P. Schwefel, editors, Parallel Problem Solving from Nature – PPSN V, pages 292–301, Berlin, Heidelberg, 1998. Springer.