Les Cahiers du GERAD ISSN: 0711–2440

Efficient search strategies for constrained multiobjective blackbox optimization

S. Le Digabel, A. Lesage-Landry, L. Salomon, C. Tribes

G–2025–28

April 2025

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2025
– Bibliothèque et Archives Canada, 2025

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2025
– Library and Archives Canada, 2025

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Efficient search strategies for constrained multiobjective blackbox optimization

Sébastien Le Digabel [a, c]

Antoine Lesage-Landry [b, c]

Ludovic Salomon [a, c]

Christophe Tribes [a, c]

[a] *Département de mathématiques et de génie industriel, Polytechnique Montréal, Montréal (Qc), Canada, H3T H3C 3A7*

[b] *Département de génie électrique, Polytechnique Montréal, Montréal (Qc), Canada, H3T H3C 3A7*

[c] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[d] *MILA, Montréal (Qc), Canada, H2S 3H1*

sebastien.le-digabel@polymtl.ca
antoine.lesage-landry@polymtl.ca
ludovic.salomon@polymtl.ca
christophe.tribes@polymtl.ca

April 2025
Les Cahiers du GERAD
G–2025–28

Abstract : Multiobjective blackbox optimization deals with problems where the objective and constraint functions are the outputs of a numerical simulation. In this context, no derivatives are available, nor can they be approximated by finite differences, which precludes the use of classical gradient-based techniques. The DMulti-MADS algorithm implements a state-of-the-art direct search procedure for multiobjective blackbox optimization based on the mesh adaptive direct search (MADS) algorithm. Since its conception, many search strategies have been proposed to improve the practical efficiency of the single-objective MADS algorithm. Inspired by this previous research, this work proposes the integration of two search heuristics into the DMulti-MADS algorithm. The first uses quadratic models, built from previously evaluated points, which act as surrogates for the true objectives and constraints, to suggest new promising candidates. The second exploits the sampling strategy of the Nelder-Mead algorithm to explore the decision space for new non-dominated points. Computational experiments on analytical problems and three engineering applications show that the use of such search steps considerably improves the performance of the DMulti-MADS algorithm.

Keywords : Multiobjective optimization, derivative-free optimization (DFO), blackbox optimization (BBO), constrained optimization

Acknowledgements: This work is supported by the NSERC/Mitacs Alliance grant (#571311–21) in collaboration with Hydro-Québec and a postdoctoral fellowship of GERAD (FRQNT–309029).

Conflicts of interest: The authors report no conflict of interest.

Data availability statement: The NOMAD solver is available at https://github.com/bbopt/nomad. The solar 8.1 and solar 9.1 problems are available at github.com/bbopt/solar. The STYRENE problem is available at github.com/bbopt/styrene. The implementation of the benchmark functions can be found at https://github.com/bbopt/DMultiMads_search_benchmarks.

1 Introduction

This work considers the following constrained multiobjective optimization problem

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

$\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0,\ j \in \mathcal{J}\} \subset \mathbb{R}^n$ represents the *feasible decision set*, and $\mathcal{X}$ is a subset of $\mathbb{R}^n$. The sets $\mathbb{R}^n$ and $\mathbb{R}^m$ are, respectively, designed as the *decision space* and the *objective space*. Each objective component $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $i = 1, 2, \ldots, m$, and constraints $c_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $j \in \mathcal{J}$ are blackbox functions, for which derivatives are not available nor cannot be approximated numerically. Blackbox optimization (BBO) algorithms aim to solve such problems.

In the absence of analytical expressions, one cannot exploit certain properties of the problem (e.g., continuity, convexity, or differentiability), justifying the use of derivative-free optimization (DFO) [8, 25, 41], a more general case of BBO. Many engineering applications that face several conflicting, expensive, and/or unreliable objectives fit into this framework, e.g., [3, 31, 32, 52]. For such problems, an optimal solution for all objectives does not always exist. The goal is then to provide the practitioners with the best set of trade-off solutions from which they can extract a particular solution according to their preferences [18, 22, 45].

Prior work has already proposed several deterministic algorithms to address this class of problems. A first approach is to reformulate the original multiobjective optimization problem into a series of single-objective subproblems, each of which is solved by a dedicated single-objective blackbox solver. Direct search algorithms such as BiMADS [12] and MultiMADS [13] for biobjective and multiobjective optimization, respectively, rely on the Mesh Adaptive Direct Search (MADS) [6, 7] algorithm for single-objective constrained optimization. The BiObjective Trust-Region algorithm [37] adopts a derivative-free trust-region method. Methods with a posteriori articulation of preferences work directly with the original formulation, in contrast to the scalarization-based approaches mentioned above. The Direct MultiSearch (DMS) framework [27] and its variants [15, 16, 26, 30] generalize single-objective direct search algorithms to multiobjective optimization. The Derivative-Free MultiObjective algorithm [43] is a linesearch-based approach, while the MultiObjective Implicit Filtering method [36] can be viewed as a multiobjective extension of the implicit filtering algorithm [39].

Direct search methods are DFO iterative methods. At each iteration, they examine a finite set of trial candidates to try to improve a current (set of) incumbent solution(s). If an improvement is found, the current (set of) incumbent solution(s) is updated. Otherwise, the associated step size parameter of the current incumbent is decreased, and a new iteration is started. Most efficient direct search algorithms for single-objective optimization such as MADS [6] or Generated Search Set [40] alternate between search and poll steps. Search strategies, although not mandatory for convergence, significantly enhance the practical performance of direct search algorithms. Thus, many search strategies have been proposed for single-objective optimization: speculative search [6], quadratic models [24, 28, 29], Variable Neighbourhood Search [5], general surrogate models [10, 53], or a Nelder-Mead (NM) based search strategy [14].

In multiobjective optimization, the integration of search strategies can take two forms. Scalarization-based approaches such as BiMADS [12] or MultiMADS [13] incorporate the direct search single-objective solver MADS, which in turns implements a certain number of single-objective search heuristics. However, computational experiments [15, 16, 27] have shown that even with these search strategies activated, the algorithms have similar performance with state-of-the-art direct search methods with a posteriori articulation of preferences using only polling. References [19, 26] propose “native” search strategies specifically designed for multiobjective direct search algorithms.

This work proposes two new native search strategies for the multiobjective direct search algorithm DMulti-MADS [15, 16]. The first one is based on quadratic models in the lineage of [19]. The second search method is an adaptation of a NM strategy proposed in [14] for multiobjective optimization.

Note that the use of surrogate models in the context of multiobjective blackbox optimization is not new. The following works [2, 48, 50] combine radial basis functions with evolutionary strategies. Some research has also explored Gaussian processes within a Bayesian multiobjective framework, e.g., [17, 33]. For such methods, convergence analysis is not well established, or at most in probability (see, for example, [51]). On the contrary, the following works [37, 46, 49, 55, 56] describe multiobjective algorithms using quadratic models within a trust-region framework with convergence guarantees. However, these algorithms are limited to unconstrained or bound-constrained multiobjective optimization.

Researchers have also proposed NM-based algorithms for multiobjective optimization. The authors in [57] combine a NM strategy with a Chebyshev scalarization. In [44], the authors adopt an evolutionary strategy based on the NM simplex ordering. However, all these methods do not provide convergence guarantees and are not deterministic, potentially making them less robust that existing convergent methods.

Based on these considerations, this work:

- proposes two search strategies for constrained multiobjective blackbox optimization, motivated by their practical performance already observed in a single-objective context;
- investigates quadratic-based formulations other than the ones proposed by [19], whose main drawback is the number of subproblems to solve, which increases exponentially with the number of objectives. However, this issue remains limited for a low number of objectives, i.e., $m \leq 4$. The authors have also investigated the use of parallelism to speed up this search [54]. The formulations used in this work exploit the nature of the incumbent iterate, resulting in less problems to solve (i.e., at most linear in the number of objectives), and keeping a performance similar to the search strategy proposed in [19].

This document is organized as follows. Sections 2 and 3 summarize the main notations and multiobjective concepts, and optimality conditions. Section 4 provides a general presentation of the DMultiMADS algorithm. New single-objective formulations are given in Section 5. The new search strategies are then proposed in Section 6. This work concludes with numerical experiments in Section 7, which illustrate the potential of such new strategies in analytical benchmarks and three applications. It ends with a general conclusion.

2 Notations and Pareto dominance

This work adopts the following conventions. Given two vectors $\mathbf{y}^1,\ \mathbf{y}^2 \in \mathbb{R}^m$,

- $\mathbf{y}^1 = \mathbf{y}^2 \iff y_i^1 = y_i^2$ for $i = 1, 2, \ldots, m$;
- $\mathbf{y}^1 < \mathbf{y}^2 \iff y_i^1 < y_i^2$ for $i = 1, 2, \ldots, m$;
- $\mathbf{y}^1 \leq \mathbf{y}^2 \iff y_i^1 \leq y_i^2$ for $i = 1, 2, \ldots, m$ and there exists at least an index $i_0 \in \{1, 2, \ldots, m\}$ such that $y_{i_0}^1 < y_{i_0}^2$.

In multiobjective optimization, as the objective function possesses several components, one needs the concept of *Pareto dominance* [45] to compare a pair of points.

Definition 2.1. Let two feasible decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ in Ω.

- $\mathbf{x}^1$ *dominates* $\mathbf{x}^2$ ($\mathbf{x}^1 \prec \mathbf{x}^2$) if and only if $f(\mathbf{x}^1) \leq f(\mathbf{x}^2)$.
- $\mathbf{x}^1$ *strictly dominates* $\mathbf{x}^2$ ($\mathbf{x}^1 \prec\!\prec \mathbf{x}^2$) if and only if $f(\mathbf{x}^1) < f(\mathbf{x}^2)$.
- $\mathbf{x}^1$ and $\mathbf{x}^2$ are *incomparable* ($\mathbf{x}^1 \sim \mathbf{x}^2$) if and only if $\mathbf{x}^1 \not\prec \mathbf{x}^2$ and $\mathbf{x}^2 \not\prec \mathbf{x}^1$, i.e. $f(\mathbf{x}^1) \nleq f(\mathbf{x}^2)$ and $f(\mathbf{x}^2) \nleq f(\mathbf{x}^1)$.

This definition can be extended to objective vectors, i.e., points in the objective space $\mathbb{R}^m$. It is now possible to define what an optimal solution for (MOP) is.

Definition 2.2. A feasible decision vector $\mathbf{x}^\star \in \Omega$ is said to be (globally) *Pareto optimal* if there is no other decision vector $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^\star$.

The set of all Pareto optimal solutions in Ω is called the *Pareto set* denoted by $\mathcal{X}_{\mathrm{P}}$ and its image by the objective function is designated as the *Pareto front* denoted by $\mathcal{Y}_{\mathrm{P}} \subseteq \mathbb{R}^m$.

Definition 2.3. A feasible decision vector $\mathbf{x}^\star \in \Omega$ is said to be *locally Pareto optimal* if it exists a neighbourhood $\mathcal{N}(\mathbf{x}^\star)$ of $\mathbf{x}^\star$ such that there is no other decision vector $\mathbf{x} \in \mathcal{N}(\mathbf{x}^\star) \cap \Omega$ which satisfies $\mathbf{x} \prec \mathbf{x}^\star$.

Any set of locally Pareto optimal solutions is called a *local Pareto set*. The Pareto set may contain an infinite number of solutions [23], making its enumeration impractical. Furthermore, obtaining even locally optimal solutions in a DFO context is a challenging task. Algorithms try to find a representative set of non-dominated points, i.e., a *Pareto set approximation* [58], whose mapping by the objective function f is designed as a *Pareto front approximation*. In the best case, all elements of a Pareto set approximation should be (locally) Pareto optimal, but this condition is not always satisfied.

The two following objective vectors provide information on the range of the Pareto front. The *ideal objective vector* [45] $\mathbf{y}^{\mathrm{I}} \in \mathbb{R}^m$ (if it exists) bounds the Pareto front from below and is defined as

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

The ideal objective vector is related to the *extreme points of the Pareto set*, i.e., the elements of the Pareto set that are the solutions of each single-objective problem $\min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for $i = 1, 2, \ldots, m$. The *nadir objective vector* [45] $\mathbf{y}^{\mathrm{N}} \in \mathbb{R}^m$ (if it exists) provides an upper bound on the Pareto front. It is defined as

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

3 Necessary optimality conditions

This section contains a summary of the necessary optimality conditions for multiobjective optimization used in the rest of this work. Clarke nonsmooth analysis [21] provides a rigorous framework for studying derivative-free multiobjective methods in the constrained case.

The Clarke tangent cone is a generalization of the tangent cone used in classical differentiable nonlinear optimization. Roughly speaking, it defines a set of directions in the decision space that an algorithm can use to move from one feasible solution to another. The definition and notations are taken from [6].

Definition 3.1 (Clarke tangent vector)**.** A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point $\mathbf{x}$ in the closure of Ω if for every sequence $\{\mathbf{y}^k\}$ of elements of Ω that converge to $\mathbf{x}$ and every sequence of positive real numbers $\{t^k\}$ converging to zero, there exists a sequence of vectors $\{\mathbf{w}^k\}$ converging to $\mathbf{d}$ such that $\mathbf{x}^k + t^k\mathbf{w}^k \in \Omega$.

The set of all Clarke tangent vectors to Ω at $\mathbf{x}$ is called the Clarke tangent cone to Ω at $\mathbf{x}$, denoted as $\mathcal{T}_\Omega^{\mathrm{Cl}}(\mathbf{x})$. The hypertangent cone is the interior of the Clarke tangent cone, assuming that the latter exists. It is often used in the convergence analysis of direct search methods.

Definition 3.2 (Hypertangent vector)**.** A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a hypertangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at $\mathbf{x}$ in Ω if there exists a scalar $\epsilon > 0$ such that

$$\mathbf{y} + t\mathbf{w} \in \Omega, \forall \mathbf{y} \in \Omega \cap \mathcal{B}(\mathbf{x};\epsilon), \mathbf{w} \in \mathcal{B}(\mathbf{x};\epsilon) \text{ and } 0 < t < \epsilon.$$

The set of all hypertangent vectors at $\mathbf{x} \in \Omega$ is called the hypertangent cone at $\mathbf{x}$ in Ω, denoted as $\mathcal{T}_\Omega^{\mathrm{H}}(\mathbf{x})$.

Assuming that f is Lipschitz continuous near $\mathbf{x} \in \Omega$, i.e., each component f_i for $i = 1, 2, \ldots, m$ of the objective function is Lipschitz continuous near $\mathbf{x}$, one can define the Clarke-Jahn generalized derivative [38] of each objective component in the direction $\mathbf{d}$ belonging to the hypertangent cone at $\mathbf{x}$ in Ω:

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

The Clarke-Jahn generalized derivative of objective f_i in the direction $\mathbf{v}$ belonging to the Clarke tangent cone at $\mathbf{x}$ in Ω can be found by going to the limit, i.e., $f_i^o(\mathbf{x}; \mathbf{v}) = \lim\limits_{\substack{\mathbf{d} \in \mathcal{T}_\Omega^{\mathrm{H}}(\mathbf{x}) \\ \mathbf{d} \to \mathbf{v}}} f_i^o(\mathbf{x}; \mathbf{d})$ [6].

It is now possible to give the following main necessary condition for nonsmooth multiobjective optimization.

Theorem 3.1. *Let f be Lipschitz continuous near $\hat{\mathbf{x}} \in \Omega$. If $\hat{\mathbf{x}}$ is locally Pareto optimal, then for any direction $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$, there exists at least one index $i \in \{1, 2, \ldots, m\}$ such that:*

$$f_i^o(\hat{\mathbf{x}}; \mathbf{d}) \geq 0.$$

The above theorem means that there does not exist any descent direction that is a descent direction for all objectives at a locally Pareto optimal solution.

4 DMulti-MADS

This work considers the DMulti-MADS algorithm [15] with the progressive barrier technique [16] to handle inequality constraints. DMulti-MADS is an iterative direct search method that extends the MADS algorithm [6, 7] to multiobjective constrained optimization. Inspired by BiMADS [12], it can be seen as a specific instance of the DMS framework [27]. Good numerical results on benchmark functions [15] and real applications justify its use in this research.

Each iteration of DMulti-MADS consists of two main steps: the *search* and the *poll*. The search is an optional step that allows the use of various strategies, such as surrogates [19], to explore the decision space by evaluating a finite number of candidates. The poll performs a local exploration of the decision space in a region delimited by a frame size parameter $\Delta^k > 0$ centered around a current incumbent. The poll follows stricter rules, which, when properly configured, ensures convergence [15].

All candidates generated during an iteration k must belong to a discrete set called the *mesh* defined by

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

where $\delta^k > 0$ is the mesh size parameter, $\mathbf{D} = \mathbf{G}\mathbf{Z} \in \mathbb{R}^{n \times n_{\mathrm{D}}}$ is a positive spanning set matrix, i.e., a matrix whose columns form a positive spanning set for $\mathbb{R}^n$ (see [8, Chapter 6] or [25, Chapter 2]) for some non-singular matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ and integer matrix $\mathbf{Z} \in \mathbb{Z}^{n \times n_{\mathrm{D}}}$. In practice, $\mathbf{G} = I_n$ and $\mathbf{Z} = [-I_n\ I_n] = \mathbf{D}$, where I_n is the identity matrix of size $n \times n$. The *cache* $V^k \subset \mathcal{X}$ is the set of trial points which have been evaluated by the algorithm by the start of iteration k.

The poll set at iteration k, denoted as P^k, is a subset of M^k. Its construction must satisfy specific requirements. It involves a current iterate incumbent $(\mathbf{x}^k, \Delta^k) \in V^k \times \mathbb{R}_+^*$. The point $\mathbf{x}^k$ is denoted as the current frame incumbent center at iteration k. The parameter Δ^k is called the *frame size parameter* at iteration k and must satisfy $0 < \delta^k \leq \Delta^k$ for all k and $\lim_{k \in K} \delta^k = 0$ if and only if $\lim_{k \in K} \Delta^k = 0$ for any subset of indices K. In practice, $\delta^k = \min\{\Delta^k, \left(\Delta^k\right)^2\}$ satisfies these requirements. Formally,

$$P^k = \{\mathbf{x}^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k\} \subset \{\mathbf{x} \in M^k : \|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \Delta^k b\},$$

where $b = \max\{\|\mathbf{d}'\|_\infty : \mathbf{d}' \in \mathbf{D}\}$, $\mathbb{D}^k_\Delta$ is a positive spanning set of directions. Interested readers can refer to [1, 6] for procedures to build such positive spanning sets $\mathbb{D}^k_\Delta$.

To handle inequality constraints, DMulti-MADS with the progressive barrier uses the constraint violation function [34, 35]:

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

The constraint violation function quantifies the violation of the constraints at a given point. It is non-negative and satisfies $h(\mathbf{x}) = 0$ if and only if $\mathbf{x} \in \Omega$.

Using this constraint violation function, DMulti-MADS introduces an extension of the dominance relation for constrained multiobjective optimization [16].

Definition 4.1 (Dominance for constrained optimization)**.** Let two decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ be in $\mathcal{X}$. Then $\mathbf{x}^1 \in \mathcal{X}$ is said to dominate $\mathbf{x}^2 \in \mathcal{X}$ if

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

Note that a pair of feasible and infeasible points are never compared.

Each iteration of DMulti-MADS is organized around at least one of the following set of current incumbent solutions:

- the set of feasible incumbent solutions

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

- the set of infeasible incumbent solutions

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

  with U^k the set of infeasible nondominated points

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

  and $h^k_{\max} > 0$ the barrier threshold at iteration k.

Using one of these two sets of current incumbents, DMulti-MADS constructs a so-called *iterate list* [27] of non-dominated points

$$L^k = \{(\mathbf{x}^l, \Delta^l) : \mathbf{x}^l \in X^k, l = 1, 2, \ldots, l_k\},$$

where $l_k = |X^k|$ and $X^k \subseteq \begin{cases} I^k & \text{if } F^k = \emptyset \\ F^k & \text{otherwise} \end{cases}$, with $X_k \neq \emptyset$. Each element of the list possesses its own frame and mesh size parameters.

At iteration k, DMulti-MADS chooses the current incumbent iterate $(\mathbf{x}^k, \Delta^k)$ among the elements of L^k. This last one must satisfy at least the following condition:

$$(\mathbf{x}^k, \Delta^k) \in \{(\mathbf{x}, \Delta) \in L^k : \tau^{w^+} \Delta^k_{\max} \leq \Delta\},$$

where $\tau \in \mathbb{Q} \cap (0, 1)$ is the *mesh size adjustment parameter* [8], $w^+ \in \mathbb{N}$ a fixed integer, and $\Delta^k_{\max}$ the maximum frame size parameter at iteration k defined as $\Delta^k_{\max} = \displaystyle\max_{l=1,2,\ldots,|L^k|} \Delta^l$.

When $F^k \neq \emptyset$ and $I^k \neq \emptyset$, the poll may be executed around two frame centers belonging to these two sets with the frame size parameter Δ^k of the current iteration k.

All new candidates generated during the poll and the search are affected a frame size parameter larger than the frame size parameter of the current incumbent iterate. The frame size parameter of the current incumbent is reduced at the end of the iteration if no other candidate is found which dominates at least one of the current frame centers. In all cases, the iterate list is filtered at the end of the iteration to remove new dominated points for the next iteration and the barrier threshold is reduced or kept constant, i.e., $h_{\max}^{k+1} \leq h_{\max}^k$.

Algorithm 1 provides a high-level description of DMulti-MADS for constrained optimization. Interested readers are referred to [15, 16] for detailed information.

Algorithm 1 A high-level description of the DMulti-MADS algorithm for constrained optimization, inspired by [15, 16].

```
Initialization: Given a finite set of points $V^0 \subset \mathcal{X}$, choose $\Delta^0 > 0$, $\mathbf{D} = \mathbf{GZ}$ a positive spanning set matrix, $\tau \in (0,1) \cap \mathbb{Q}$ the frame size adjustment parameter, $w^+ \in \mathbb{N}$ a fixed integer parameter, and $h_{\max}^0 = +\infty$ the initial barrier threshold. Initialize the iterate list $L^0 = \{(\mathbf{x}^l, \Delta^0), l = 1, 2, \ldots, |L^k|\}$ for some $\mathbf{x}^l \in V^0 \cap F^0$ or (exclusively) $\mathbf{x}^l \in V^0 \cap I^0$.
for $k = 0, 1, 2, \ldots$ do
    Selection of the current frame centers. Select the current incumbent iterate $(\mathbf{x}^k, \Delta^k)$ of the iterate list $L^k$, satisfying:
        $$(\mathbf{x}^k, \Delta^k) \in \{(\mathbf{x}, \Delta) \in L^k : \tau^{w^+} \Delta_{\max}^k \leq \Delta\},$$
    and its associated second frame center it it exists.
    Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$. Initialize $L^{\text{add}} := \emptyset$.
    Search (optional): Evaluate $f$ and $h$ at a finite set of points $S^k \subset \mathcal{X}$ on the mesh $M^k$. Set $L^{\text{add}} := \{(\mathbf{x}, \Delta^k) : \mathbf{x} \in S^k\}$.
    If a 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 $P^k \subset M^k$ using at least the current frame center. Set $L^{\text{add}} := L^{\text{add}} \cup \left\{(\mathbf{x}, \Delta^k) : \mathbf{x} \in P^k\right\}$. If a success 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. Set $L^{k+1} := L^k$. Update $h_{\max}^{k+1} \leq h_{\max}^k$. Update the iterate list $L^{k+1}$ by adding new non-dominated points from $L^{\text{add}}$ with their updated associated frame center $\Delta \in \{\Delta^k, \tau^{-1}\Delta^k\}$. Remove new dominated points from $L^{k+1}$.
    If the iteration is unsuccessful, replace the iterate element $(\mathbf{x}^k, \Delta^k)$ by $(\mathbf{x}^k, \Delta^{k+1})$ with $\Delta^{k+1} := \tau\Delta^k$.
end for
```

If all trial points generated during the optimization belong to a bounded set and the set of refining directions [6] is "sufficiently rich", the main convergence result of DMulti-MADS [15] states that it will generate at least one subsequence converging to a point $\mathbf{x}^*$ such that the Clarke generalized derivative [21] $f_{i_0}^{\circ}(\mathbf{x}^*; \mathbf{d})$ will be nonnegative for at least one objective index $i_0 \in \{1, 2, \ldots, m\}$ for every hypertangent direction $\mathbf{d}$ [38] to the domain Ω at $\mathbf{x}^*$, assuming the existence of feasible iterates.

When there is no feasible iterate, one can show similarly [16] that DMulti-MADS will generate an accumulation infeasible point $\mathbf{x}^*$ such that its Clarke generalized derivative $h^{\circ}(\mathbf{x}^*; \mathbf{d})$ will be nonnegative for every hypertangent direction $\mathbf{d}$ to the domain $\mathcal{X}$ at $\mathbf{x}^*$.

All these results hold independently of the search used to build S^k, as long as this set is finite and search candidates belong to the mesh M^k. The new search strategies described in the rest of this work meet these two requirements.

5 Single-objective formulations

The search strategies proposed in this work are based on the resolution of single-objective subproblems derived from the original multiobjective optimization problem (MOP), in the continuation of [12, 13, 19].

Definition 5.1 (adapted from [12, 13]). Consider the single-objective optimization problem:

$$R_r : \min_{\mathbf{x}\in\Omega} \psi_Y(\mathbf{x}) \text{ with } \psi_Y(\mathbf{x}) = \phi_Y\left(f(\mathbf{x})\right),$$

where $\phi_Y : \mathbb{R}^m \to \mathbb{R}$ is parameterized with respect to some finite reference set $Y \subset \mathbb{R}^m \neq \emptyset$ satisfying

$$\text{for all } (\mathbf{r}^1, \mathbf{r}^2) \in Y, \mathbf{r}^1 \not\leq \mathbf{r}^2 \text{ and } \mathbf{r}^2 \not\leq \mathbf{r}^1.$$

Then R_r is called a *single-objective formulation at Y of (MOP)* if the following conditions hold:

- if f is Lipschitz continuous near some $\tilde{\mathbf{x}} \in \Omega$, then ψ_Y is also Lipschitz continuous near $\tilde{\mathbf{x}} \in \Omega$.
- if f is Lipschitz continuous near some $\tilde{\mathbf{x}} \in \Omega$ with $f_i(\tilde{\mathbf{x}}) < r_i$ component-wise for at least one element $\mathbf{r} \in Y$ and if $\mathbf{d} \in \mathcal{T}_\Omega^{\mathrm{Cl}}(\tilde{\mathbf{x}})$ such that $f_i^\circ(\tilde{\mathbf{x}}; \mathbf{d}) < 0$ for $i = 1, 2, \ldots, m$, then $\psi_Y^\circ(\tilde{\mathbf{x}}; \mathbf{d}) < 0$.

Such formulation is useful as it preserves local Lipschitz continuity (first condition of Definition 5.1) and ensures that a descent direction for the objective function is also a descent direction for the single-objective formulation function (second condition of Definition 5.1).

This definition differs from the one proposed in [12, 13], as it is no more restricted to the use of only one reference vector. Note that the *single-objective distance formulation*, defined in [13] by

$$\begin{aligned} \bar{R}_{\mathbf{r}} : \min_{\mathbf{x}\in\Omega} \bar{\psi}_{\mathbf{r}}(\mathbf{x}) &= \bar{\phi}_{\mathbf{r}}(f(\mathbf{x})) \\ &= \begin{cases} -\mathrm{dist}^2(\partial\mathcal{D}(\mathbf{r}), f(\mathbf{x})) & \text{if } f(\mathbf{x}) \in \mathcal{D}(\mathbf{r}), \\ \mathrm{dist}^2(\partial\mathcal{D}(\mathbf{r}), f(\mathbf{x})) & \text{otherwise,} \end{cases} \end{aligned}$$

where $\mathrm{dist}^2(\partial\mathcal{D}(\mathbf{r}), f(\mathbf{x}))$ is the distance from $f(\mathbf{x})$ to the boundary $\partial\mathcal{D}(\mathbf{r})$ of the dominance zone $\mathcal{D}(\mathbf{r}) = \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y} \leq \mathbf{r}\}$ relative to some reference objective vector $\mathbf{r} \in \mathbb{R}^m$, satisfies the two conditions of Definition 5.1 by considering the singleton $Y = \{\mathbf{r}\}$.

By allowing the single-objective formulation to be parameterized by a reference set, an algorithm can potentially explore a larger zone of the objective space where only the zone dominated by a reference objective vector would have been prioritized. This work also uses the *single-objective dominance move formulation*, inspired by [42] and defined as

$$\begin{aligned} R_Y^{\mathrm{Do}} : \min_{\mathbf{x}\in\Omega} \psi_Y^{\mathrm{Do}}(\mathbf{x}) &= \phi_Y^{\mathrm{Do}}\left(f(\mathbf{x})\right) \\ &= \begin{cases} -\min\limits_{\mathbf{r}\in Y} \sum\limits_{i=1}^m \max\left\{0, r_i - f_i(\mathbf{x})\right\} & \text{if } f(\mathbf{x}) \text{ is not dominated by any element of } Y, \\ \min\limits_{\mathbf{r}\in Y} \sum\limits_{i=1}^m \max\left\{0, f_i(\mathbf{x}) - r_i\right\} & \text{otherwise.} \end{cases} \end{aligned}$$

Figure 1 shows the level sets of the R_Y^{Do} formulation. Note that the "zone of interest" is larger than the union of the zones relative to each element of the reference set Y. However, this formulation does not preserve the differentiability of the original problem (MOP). Also, the evaluation of ϕ_Y^{Do} at a given solution has a complexity cost of $\mathcal{O}\left(m|Y|\right)$, which could be expensive if Y is large.

The next theorem shows that R_Y^{Do} is a single-objective formulation of (MOP). Before proving the theorem, the following lemma is provided.

Lemma 5.1. *Let $(g_i)_{i\in\mathcal{I}}$ be a finite collection of functions from Ω to $\mathbb{R}$. If g_i for $i \in \mathcal{I}$ is Lipschitz continuous near some $\bar{\mathbf{x}} \in \Omega$, then $\max\limits_{i\in\mathcal{I}} g_i$ and $\min\limits_{i\in\mathcal{I}} g_i$ are Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$.*

Proof. Let $(g_i)_{i\in I}$ be Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$ with $\lambda_i \geq 0$ the Lipschitz constant associated to g_i for $i \in \mathcal{I}$. Let $\mathbf{x} \in \Omega$ be another point in the neighbourhood of $\bar{\mathbf{x}} \in \Omega$, $i_0 \in \arg\max\limits_{i\in\mathcal{I}} g_i(\bar{\mathbf{x}})$ and

$i_1 \in \arg\max_{i\in\mathcal{I}} g_i(\mathbf{x})$. Then for all $i \in \mathcal{I}$:

$$\left|\max_{i\in\mathcal{I}} g_i(\bar{\mathbf{x}}) - \max_{i\in\mathcal{I}} g_i(\mathbf{x})\right| = \begin{cases} g_{i_0}(\bar{\mathbf{x}}) - g_{i_1}(\mathbf{x}) & \leq g_{i_0}(\bar{\mathbf{x}}) - g_{i_0}(\mathbf{x}) \text{ or} \\ g_{i_1}(\mathbf{x}) - g_{i_0}(\bar{\mathbf{x}}) & \leq g_{i_1}(\mathbf{x}) - g_{i_1}(\bar{\mathbf{x}}). \end{cases}$$

Hence

$$\left|\max_{i\in\mathcal{I}} g_i(\bar{\mathbf{x}}) - \max_{i\in\mathcal{I}} g_i(\mathbf{x})\right| \leq \max_{i\in\mathcal{I}} |g_i(\bar{\mathbf{x}}) - g_i(\mathbf{x})| \leq \max_{i\in\mathcal{I}}\{\lambda_i\}\|\bar{\mathbf{x}} - \mathbf{x}\|.$$

Consequently, $\max_{i\in\mathcal{I}} g_i$ is Lipschitz continuous near $\bar{\mathbf{x}}$.

To prove the second part of the lemma, observe that

$$\left|\min_{i\in\mathcal{I}} g_i(\bar{\mathbf{x}}) - \min_{i\in\mathcal{I}} g_i(\mathbf{x})\right| = \left|\max_{i\in\mathcal{I}} -g_i(\bar{x}) - \max_{i\in\mathcal{I}} -g_i(\mathbf{x})\right| \leq \max_{i\in\mathcal{I}} |g_i(\bar{\mathbf{x}}) - g_i(\mathbf{x})| \leq \max_{i\in\mathcal{I}}\{\lambda_i\}\|\bar{\mathbf{x}} - \mathbf{x}\|,$$

which concludes the proof. □

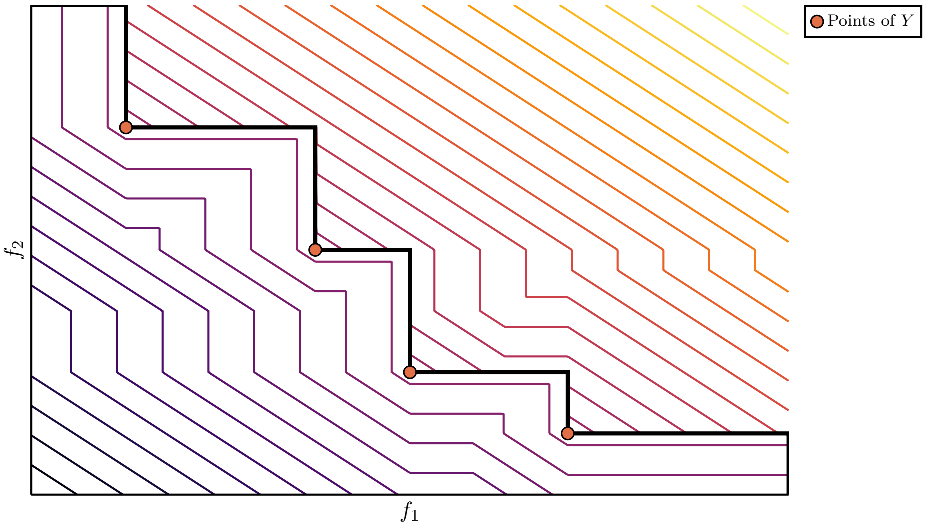


Figure 1: Levels sets in the objective space of the R_Y^{Do} formulation for a biobjective minimization problem.

Next, R_Y^{Do}'s single-objective formulation is established.

Theorem 5.1. *R_Y^{Do} is a single-objective formulation in the sense of Definition 5.1.*

Proof. Let f be Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$.

Then for all $\mathbf{r} \in Y$, $\mathbf{x} \mapsto r_i - f_i(\mathbf{x})$ is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$ for $i = 1, 2, \ldots, m$. Consequently, using Lemma 5.1, $\mathbf{x} \mapsto \max\{0, r_i - f_i(\mathbf{x})\}$ is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$, for all $\mathbf{r} \in Y$ and $i = 1, 2, \ldots, m$.

As a finite sum of Lipschitz continuous functions near $\bar{\mathbf{x}} \in \Omega$, one can deduce that $\mathbf{x} \mapsto \sum_{i=1}^{m} \max\{0, r_i - f_i(\mathbf{x})\}$ is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$ for all $\mathbf{r} \in Y$.

Reusing Lemma 5.1 yields that $\psi_Y^{\text{Do},1} : \mathbf{x} \mapsto \min_{\mathbf{r}\in Y} \sum_{i=1}^{m} \max\{0, r_i - f_i(\mathbf{x})\}$ is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$. Using the same reasoning, $\psi_Y^{\text{Do},2} : \mathbf{x} \mapsto \min_{\mathbf{r}\in Y} \sum_{i=1}^{m} \max\{0, f_i(\mathbf{x}) - r_i\}$ is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$.

To show that ψ_Y^{Do} is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$, let $\mathbf{x} \in \Omega$ be in the neighbourhood of $\bar{\mathbf{x}} \in \Omega$. Then:

$$\left|\psi_Y^{\text{Do}}(\bar{\mathbf{x}}) - \psi_Y^{\text{Do}}(\mathbf{x})\right| \leq \left|\max\left\{\psi_Y^{\text{Do},1}(\bar{\mathbf{x}}), \psi_Y^{\text{Do},2}(\bar{\mathbf{x}})\right\} - \max\left\{\psi_Y^{\text{Do},1}(\mathbf{x}), \psi_Y^{\text{Do},2}(\mathbf{x})\right\}\right|.$$

Using Lemma 5.1, $\mathbf{x} \mapsto \max\left\{\psi_Y^{\text{Do},1}(\mathbf{x}), \psi_Y^{\text{Do},2}(\mathbf{x})\right\}$ is Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$ and hence ψ_Y^{Do} satisfies the first condition of Definition 5.1.

To prove the second condition of Definition 5.1, let f be Lipschitz continuous near $\bar{\mathbf{x}} \in \Omega$ such that there exists at least one $\mathbf{r} \in Y$ with $f(\mathbf{x}) < \mathbf{r}$. Then $\psi_Y^{\text{Do}}(\bar{\mathbf{x}}) = \sum_{i=1}^m f_i(\bar{\mathbf{x}}) - \tilde{r}_i$ for one particular $\tilde{\mathbf{r}} \in Y$. Let $\mathbf{d} \in \mathcal{T}_\Omega^{\text{Cl}}(\bar{\mathbf{x}})$ satisfying $f_i^o(\bar{\mathbf{x}}; \mathbf{d}) < 0$ for $i = 1, 2, \ldots, m$. Then, by Proposition [21, 2.3.3],

$$\psi_Y^{\text{Do}}(\bar{\mathbf{x}}; \mathbf{d}) \leq \sum_{i=1}^m f_i^{\text{o}}(\bar{\mathbf{x}}; \mathbf{d}) < 0,$$

which concludes the proof. □

All elements of the reference set $Y \neq \emptyset$ can be selected anywhere in the objective space $\mathbb{R}^m$, as long as their number is finite and that they are incomparable to each other. When these conditions are satisfied, one can show that the single-objective dominance move formulation R_Y^{Do} "partially captures" the Pareto dominance relation. This is discussed in the following lemma.

Lemma 5.2. *Let $(\mathbf{x}^1, \mathbf{x}^2) \in \Omega^2$ and $Y \subset \mathbb{R}^m$ satisfying Definition 5.1.*

(i) If $\mathbf{x}^1 \prec\prec \mathbf{x}^2$, then $\psi_Y^{Do}(\mathbf{x}^1) < \psi_Y^{Do}(\mathbf{x}^2)$.

(ii) If $\mathbf{x}^1 \prec \mathbf{x}^2$, then $\psi_Y^{Do}(\mathbf{x}^1) \leq \psi_Y^{Do}(\mathbf{x}^2)$.

Proof. Let start with the first case. Suppose that $\mathbf{x}^1 \prec\prec \mathbf{x}^2$ and consider the following situations:

1. $f(\mathbf{x}^1)$ is not dominated by any element of Y and there exists at least one $\mathbf{r} \in Y$ such that $\mathbf{r} \leq f(\mathbf{x}^2)$. By definition, $\psi_Y^{\text{Do}}(\mathbf{x}^1) < 0 \leq \psi_Y^{\text{Do}}(\mathbf{x}^2)$.
2. $f(\mathbf{x}^1)$ and $f(\mathbf{x}^2)$ are dominated by at least one element of Y. Let $\mathbf{r} \in Y$. Then for all $i = 1, 2, \ldots, m$,

$$f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2) \iff f_i(\mathbf{x}^1) - r_i < f_i(\mathbf{x}^2) - r_i \Rightarrow \max\left\{0, f_i(\mathbf{x}^1) - r_i\right\} \leq \max\left\{0, f_i(\mathbf{x}^2) - r_i\right\}.$$

All elements of Y being non-dominated between each other, there exists at least one objective index $i_0 \in \{1, 2, \ldots, m\}$ such that $f_{i_0}(\mathbf{x}^1) - r_{i_0} > 0 \Rightarrow f_{i_0}(\mathbf{x}^2) - r_{i_0} > f_{i_0}(\mathbf{x}^1) - r_{i_0} > 0$ (by assumption). Hence,

$$\sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^1) - r_i\right\} < \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^2) - r_i\right\}.$$

This inequality is satisfied for all $\mathbf{r} \in Y$. Let $(j_1, j_2) \in \{1, 2, \ldots, |Y|\}^2$ such that

$$\psi_Y^{\text{Do}}(\mathbf{x}^1) = \min_{\mathbf{r} \in Y} \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^1) - r_i\right\} = \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^1) - r_i^{j_1}\right\},$$

and,

$$\psi_Y^{\text{Do}}(\mathbf{x}^2) = \min_{\mathbf{r} \in Y} \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^2) - r_i\right\} = \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^2) - r_i^{j_2}\right\}.$$

Then:

$$\begin{aligned}\psi_Y^{\text{Do}}(\mathbf{x}^1) = \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^1) - r_i^{j_1}\right\} \leq \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^1) - r_i^{j_2}\right\} < \sum_{i=1}^m \max\left\{0, f_i(\mathbf{x}^2) - r_i^{j_2}\right\} \\ = \psi_Y^{\text{Do}}(\mathbf{x}^2).\end{aligned}$$

3. $f_i(\mathbf{x}^1)$ and $f(\mathbf{x}^2)$ are not dominated by any element of Y. Let $\mathbf{r} \in Y$. Then for all $i = 1, 2, \dots, m$,

$$f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2) \iff r_i - f_i(\mathbf{x}^1) > r_i - f_i(\mathbf{x}^2) \Rightarrow \max\left\{0, r_i - f_i(\mathbf{x}^1)\right\} \geq \max(0, r_i - f_i(\mathbf{x}^2)).$$

As $f(\mathbf{x}^2)$ is not dominated by $\mathbf{r}$ and $\mathbf{x}^1$ strictly dominates $\mathbf{x}^2$, there exists at least one component objective index $i_0 \in \{1, 2, \dots, m\}$ such that $r_{i_0} - f_{i_0}(\mathbf{x}^1) > r_{i_0} - f_{i_0}(\mathbf{x}^2) \geq 0$. Hence

$$\sum_{i=1}^{m} \max\left\{0, r_i - f_i(\mathbf{x}^1)\right\} > \sum_{i=1}^{m} \max\left\{0, r_i - f_i(\mathbf{x}^2)\right\}.$$

Let $(j_1, j_2) \in \{1, 2, \dots, |Y|\}^2$ such that

$$\min_{\mathbf{r}\in Y} \sum_{i=1}^{m} \max\left\{0, r_i - f_i(\mathbf{x}^1)\right\} = \sum_{i=1}^{m} \max\left\{0, r_i^{j_1} - f_i(\mathbf{x}^1)\right\},$$

and,

$$\min_{\mathbf{r}\in Y} \sum_{i=1}^{m} \max\left\{0, r_i - f_i(\mathbf{x}^2)\right\} = \sum_{i=1}^{m} \max\left\{0, r_i^{j_2} - f_i(\mathbf{x}^2))\right\}.$$

Then:

$$\begin{aligned}\psi_Y^{\text{Do}}(\mathbf{x}^1) = -\sum_{i=1}^{m} \max\left\{0, r_i^{j_1} - f_i(\mathbf{x}^1)\right\} < -\sum_{i=1}^{m} \max\left\{0, r_i^{j_1} - f_i(\mathbf{x}^2)\right\} &\leq -\sum_{i=1}^{m} \max\left\{0, r_i^{j_2} - f_i(\mathbf{x}^2)\right\} \\ &= \psi_Y^{\text{Do}}(\mathbf{x}^2).\end{aligned}$$

The second part of the lemma follows the same reasoning. Note that in this case, the inequality is not strict in Cases 2 and 3, as illustrated in Figure 2. □

Finally, the following theorem shows that solving the R_Y^{Do} single-objective formulation could result in a locally Pareto optimal solution. Note that, in practice, one does not solve R_Y^{Do} to optimality.

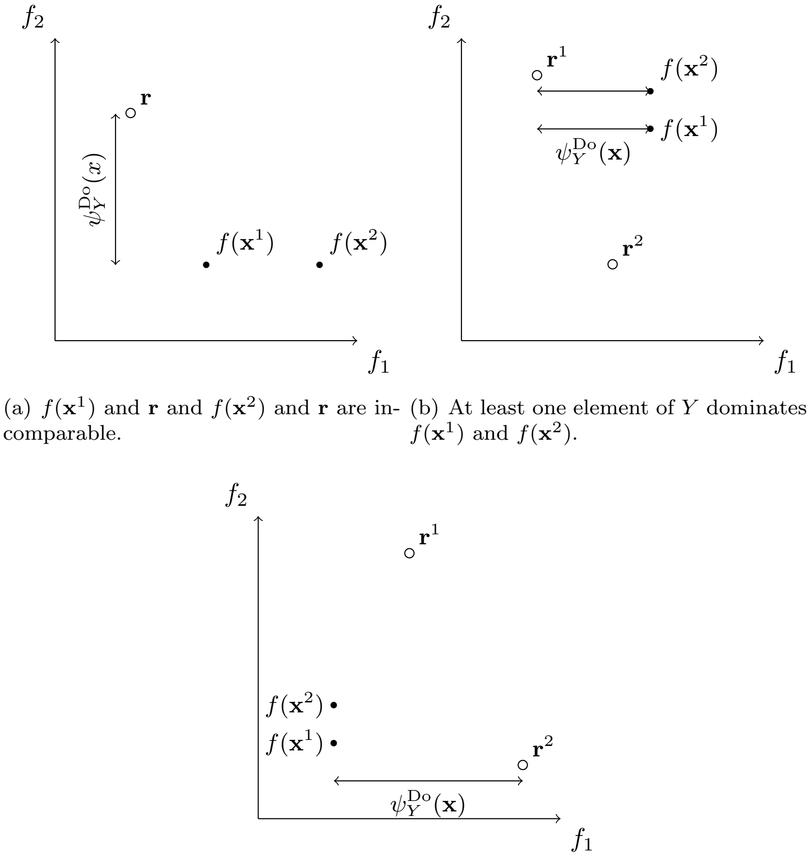
Theorem 5.2 (adapted from [13]). *If $\hat{\mathbf{x}} \in \Omega$ is the unique solution of R_Y^{Do} for some reference set $Y \subset \mathbb{R}^m$ satisfying Definition 5.1, then $\bar{\mathbf{x}}$ is Pareto optimal for (MOP).*

Proof. Let $\hat{\mathbf{x}} \in \Omega$ be the unique solution of R_Y^{Do}. Let $\mathbf{x} \in \Omega$ with $\mathbf{x} \neq \hat{\mathbf{x}}$. Then $\psi_Y^{\text{Do}}(\hat{\mathbf{x}}) < \psi_Y^{\text{Do}}(\mathbf{x})$. By Lemma 5.2 and the fact that $\mathbf{x} \neq \hat{\mathbf{x}}$, there exists at least one index $i_0 \in \{1, 2, \dots, m\}$ such that $f_{i_0}(\hat{\mathbf{x}}) < f_{i_0}(\mathbf{x})$ and $\mathbf{x}$ does not dominate $\hat{\mathbf{x}}$. The point $\hat{\mathbf{x}}$ is therefore Pareto optimal. □

6 Search strategies for the DMulti-MADS algorithm

When designing an algorithm that solves a multiobjective problem, several goals are considered to obtain a good Pareto set approximation.

- Ideally, all elements of the Pareto set approximation should be (locally) Pareto optimal, i.e., the distance between the Pareto front and its representation in the objective space should be minimized.
- It should contain the extreme solutions of the Pareto set, i.e., solutions that minimize each objective component separately. Such solutions provide information about the “limit” trade-offs the decision maker can face when a choice has to be made.
- Its representation in the objective space should be dense, i.e., the gaps between the different solutions should be minimized.



(a) $f(\mathbf{x}^1)$ and $\mathbf{r}$ and $f(\mathbf{x}^2)$ and $\mathbf{r}$ are incomparable.

(b) At least one element of Y dominates $f(\mathbf{x}^1)$ and $f(\mathbf{x}^2)$.

(c) $f(\mathbf{x}^1)$ and $f(\mathbf{x}^2)$ dominate at least one element of Y.

Figure 2: Illustration for a biobjective minimization problem: $\mathbf{x}^1 \prec \mathbf{x}^2$ but $\psi_Y^{\mathrm{Do}}(\mathbf{x}^1) = \psi_Y^{\mathrm{Do}}(\mathbf{x}^2)$.

Originally, DMulti-MADS tries to satisfy these different goals by using the information provided by the frame size parameters of the iterate list. Their values act as indicators to quantify the progression of each solution towards a locally Pareto optimal point. Furthermore, DMulti-MADS takes into account the objective values of each element of the iterate list to fulfill largest gaps into the current Pareto front approximation [15]. However, restricting to poll steps only as it was proposed in the implementation of DMulti-MADS may be too rigid in exploring the decision space and not precise enough to efficiently achieve the three goals mentioned above.

This section describes two search strategies for DMulti-MADS, based on the single-objective formulations described in Section 5.

6.1 The MultiMADS search strategy

Inspired by the BiMADS and MultiMADS algorithms, the MultiMADS search strategy introduces an additional search step in DMulti-MADS that aims to accelerate the algorithm progress towards a good solution set. This search adaptively switches between a single-objective distance formulation subproblem optimization presented in Section 5 and an extreme point exploration, depending on the nature of the current frame incumbent $\mathbf{x}^k$ at iteration k. Algorithm 2 details the procedure for a fixed k.

Algorithm 2 The MultiMADS search strategy.

```
**Initialization**: Let $(\mathbf{x}^k, \Delta^k) \in L^k$ be the current frame incumbent chosen by DMulti-MADS at iteration $k$ and $V^k \subset \mathcal{X}$ the set of points already evaluated by DMulti-MADS from the beginning of iteration $k$.
**Determination of the nature of $\mathbf{x}^k$**. If $\mathbf{x}^k$ is a current extreme solution, i.e., there exists at least one objective index $i_0 \in \{1, 2, \dots, m\}$ for which $f_{i_0}(\mathbf{x}^k) = \min_{(\mathbf{x},\Delta) \in L^k} f_{i_0}(\mathbf{x})$, go to *extreme point exploration*; else go to *single-objective distance formulation exploration*.
**Extreme point exploration**.
for objective index $i \in \{1, 2, \dots, m\}$ for which $\mathbf{x}^k$ is a current extreme solution do
    Solve
        $$\min_{\mathbf{x} \in \mathcal{X}^k} f_i(\mathbf{x}),$$
    using a subsolver procedure, starting from $\mathbf{x}^k$, where $\mathcal{X}^k \subseteq \mathcal{X}$ is a subset of $X$, depending on $\Delta^k$, $V^k$ and the choice of the subsolver.
    Collect all solution points $S^{\text{tmp}} \subset \mathcal{X} \cap M^k$ proposed by the subsolver. Set $S^k := S^k \cup S^{\text{tmp}}$.
end for
If a success criterion is met, the search procedure may terminate.
Exit the procedure.
**Single-objective distance formulation exploration.** Compute the reference objective vector
    $$\mathbf{r}^k := \texttt{computeSingleObjectiveDistanceFormReferencePoint}(\mathbf{x}^k, L^k) \text{ (see Algorithm 3).}$$
Solve
    $$\min_{\mathbf{x} \in \mathcal{X}^k} \bar{\psi}_{\mathbf{r}^k}(\mathbf{x}),$$
using a subsolver procedure, starting from $\mathbf{x}^k$, where $\mathcal{X}^k \subseteq \mathcal{X}$ is a subset of $\mathcal{X}$, depending on $\Delta^k$, $V^k$ and the choice of the subsolver.
Collect all solution points $S^{\text{tmp}} \subset \mathcal{X} \cap M^k$ proposed by the subsolver. Set $S^k := S^k \cup S^{\text{tmp}}$. If a success criterion is met, the search procedure may terminate.
```

The procedure starts by determining the nature of the current incumbent $\mathbf{x}^k$. If it is a current extreme solution, an attempt is made to extend the current Pareto front approximation by trying to minimize each objective for which $\mathbf{x}^k$ is marked as extreme. Otherwise, the MultiMADS search strategy computes a reference objective vector $\mathbf{r}^k \in \mathbb{R}^m$ using the iterate list L^k and the current incumbent $\mathbf{x}^k$.

Algorithm 3 details this process. This procedure is an adaptation of the reference point determination procedure used for BiMADS [12] to more than two objectives. It returns a reference vector $\mathbf{r}^k$ satisfying $f(\mathbf{x}^k) \leq \mathbf{r}^k$ or $\mathbf{r}^k = f(\mathbf{x}^k)$. Note that $\mathbf{r}^k$ is never computed if $\mathbf{x}^k$ is an extreme point of the current Pareto front approximation. By approximately solving the associated single-objective distance formulation subproblem, one can hope to converge towards the Pareto front, or to fill the gaps in the current Pareto front approximation around $f(\mathbf{x}^k)$.

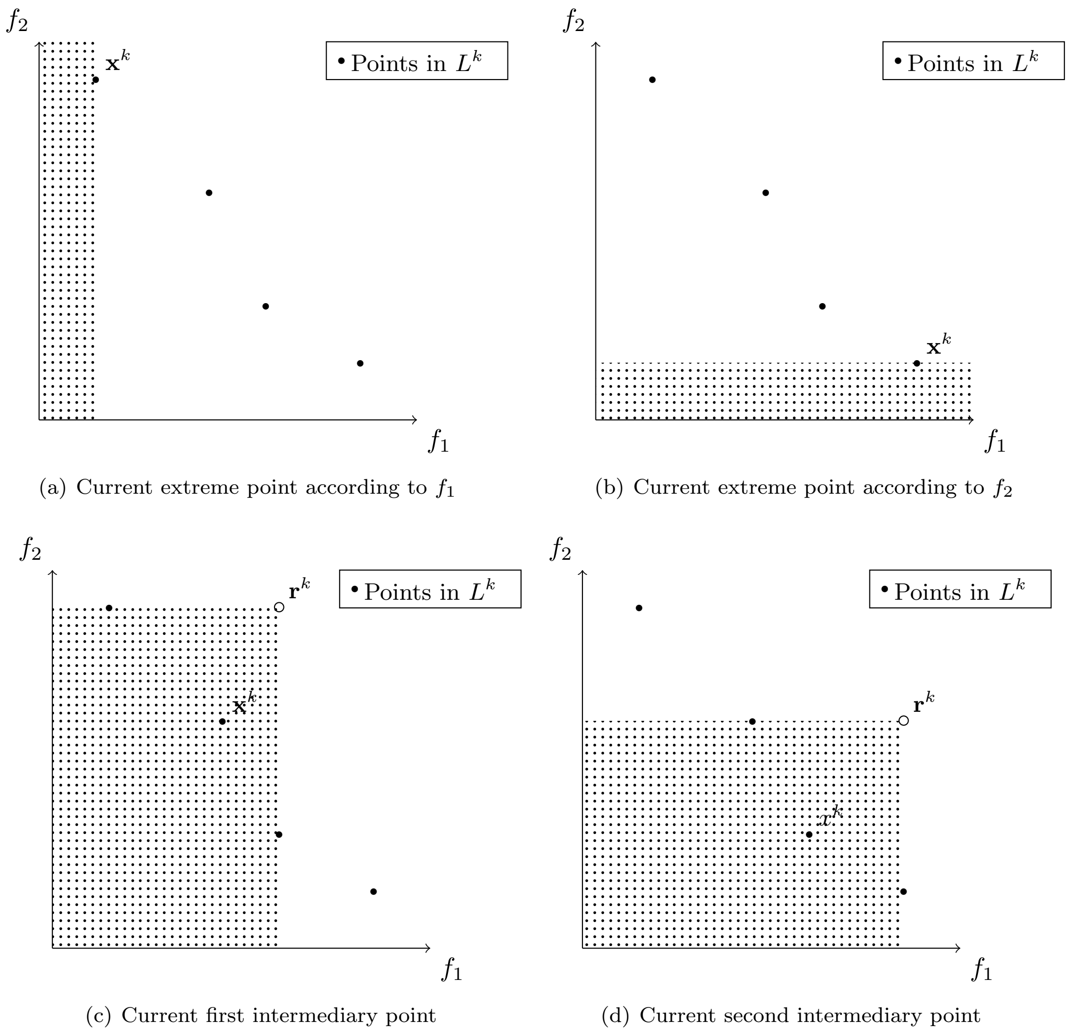
Algorithm 3 `computeSingleObjectiveDistanceFormReferencePoint`$(\mathbf{x}^k, L^k)$**.**

```
Let $\mathbf{r}^k := \mathbf{0}$.
for $i = 1, 2, \dots, m$ do
    Order $L^k = \{(\mathbf{x}^1, \Delta^1), (\mathbf{x}^2, \Delta^2), \dots, (\mathbf{x}^{|L^k|}, \Delta^{|L^k|}\}$ by increasing objective component value $f_i$, i.e., $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \leq \dots \leq f_i(\mathbf{x}^{|L^k|})$.
    Let $j_k \in \{1, \dots, |L^k|\}$ be the list index of the sorted iterate list corresponding to $(\mathbf{x}^{j_k}, \Delta^{j_k}) = (\mathbf{x}^k, \Delta^k)$.
    Define the following list index $j_{\text{select}} := \min\left\{j_k + 1, |L^k|\right\}$.
    Set $r_i := f_i(\mathbf{x}^{j_{\text{select}}})$.
end for
Return $\mathbf{r}^k$.
```

All the points proposed by the subsolver are stored into S^k and evaluated by the blackbox. When a success criterion is met, the search may terminate, i.e., the other search strategies that follow this search procedure are not executed; the poll is skipped and DMulti-MADS goes to the update step as described in Algorithm 1.

The efficiency of this procedure depends on the choice of the subsolver, which will be detailed in Section 6.3.

Figure 3 illustrates the zones of interest prioritized by the subsolver for different choices of $\mathbf{x}^k$.



(a) Current extreme point according to f_1

(b) Current extreme point according to f_2

(c) Current first intermediary point

(d) Current second intermediary point

Figure 3: Zones of interest (dotted) in the objective space for the MultiMADS search strategy for different configuration of incumbent points $\mathbf{x}^k$ in an iterate list L^k for a biobjective minimization problem.

6.2 The dominance move search strategy

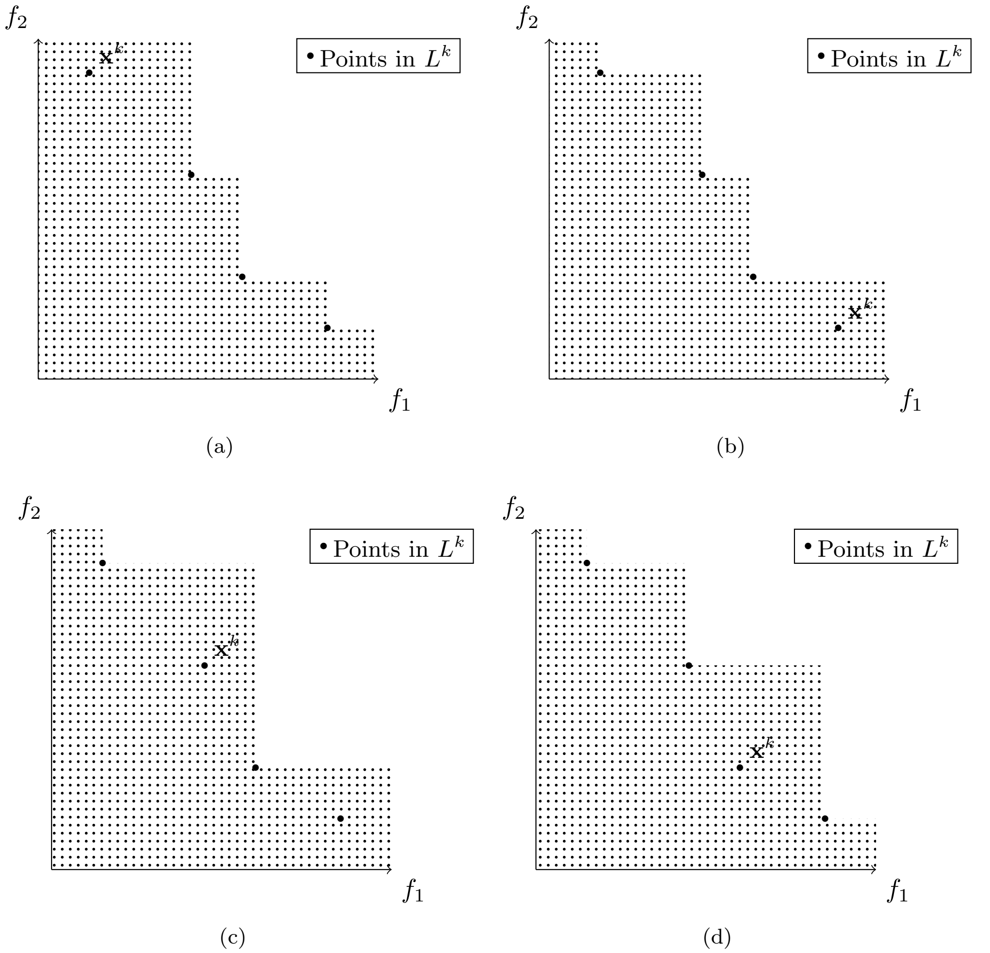
Algorithm 4 describes the dominance move search strategy. The procedure starts by determining a reference objective set, using the iterate list L^k and the current incumbent solution $\mathbf{x}^k$. If the iterate list contains a single element, $f(\mathbf{x}^k)$ is used as the reference set. Otherwise, the reference set contains all objective vectors $f(\mathbf{x})$ corresponding to each element $(\mathbf{x}, \Delta)$ of the iterate list L^k, at the exception of $f(\mathbf{x}^k)$. Then, the procedure solves the corresponding single-objective dominance formulation subproblem using a single-objective subsolver (see Section 6.3).

Algorithm 4 The dominance move search strategy.

```
Initialization: Let $(\mathbf{x}^k, \Delta^k) \in L^k$ be the current frame incumbent chosen by DMulti-MADS (see Algorithm 1) at iteration $k$ and $V^k \subset \mathcal{X}$ the set of points already evaluated by DMulti-MADS since the beginning of iteration $k$.
Determination of the reference objective list $R^k$. If $|L^k| = 1$, set $R^k := \{f(\mathbf{x}^k)\}$. Otherwise, set $R^k := \{f(\mathbf{x}) : (\mathbf{x}, \Delta) \in L^k\} \setminus \{f(\mathbf{x}^k)\}$.
Single-objective dominance move exploration.
Solve
    $$\min_{\mathbf{x} \in \mathcal{X}^k} \psi^{\mathrm{Do}}_{R^k}(\mathbf{x})$$
  using a subsolver procedure, starting from $\mathbf{x}^k$, where $\mathcal{X}^k \subseteq \mathcal{X}$ is a subset of $\mathcal{X}$, depending on $\Delta^k$, $V^k$ and the choice of the subsolver. Collect all solution points $S^{\mathrm{tmp}} \subset \mathcal{X} \cap M^k$ proposed by the subsolver. Set $S^k := S^k \cup S^{\mathrm{tmp}}$.
```

All the points proposed by the subsolver are stored in the search set S^k. As for the MultiMADS search strategy, the search may terminate if a success criterion is met. In this case, DMulti-MADS skips all the successive search strategies, and the poll step, and moves to the parameter update (see Algorithm 1).

If the dominance move search strategy considers all elements of the iterate list L^k for the construction of the reference objective set R^k, then for all $(\mathbf{x}, \Delta) \in L^k$, $\psi^{\mathrm{Do}}_{R^k}(\mathbf{x}) = 0$. By excluding $f(\mathbf{x}^k)$ from R^k, the search strategy prioritizes the exploration of a zone near $f(\mathbf{x}^k)$, while retaining the possibility for the subsolver to search for other points in non-dominated parts by R^k of the objective space. Figure 4 illustrates the zones of interest in the objective space for a biobjective minimization problem.



(a)

(b)

(c)

(d)

Figure 4: Zones of interest (dotted) in the objective space for the dominance move search strategy for different configuration of incumbent points $\mathbf{x}^k$ in an iterate list L^k for a biobjective minimization problem.

6.3 Subsolvers algorithms

This section details the single-objective subsolvers used by the new search strategies presented in Sections 6.1 and 6.2.

6.3.1 The quadratic model search step

Quadratic models are commonly used in single-objective DFO to locally approximate blackbox functions (see for example [24, 29]). This work uses quadratic regression models inspired by [24], but other models could also have been considered: e.g., minimum Frobenius norm models [29], second directional derivative-based update Hessian models [20].

Consider the following quadratic model

$$Q(\mathbf{x}) = \alpha_0 + \mathbf{g}^\top \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x},$$

of a true function f_{bb}, where $\alpha_0 \in \mathbb{R}$, $\mathbf{g} \in \mathbb{R}^n$ and $\mathbf{H} \in \mathbb{R}^{n \times n}$ symmetric are the unknown parameters to identify. These $p = \frac{(n+1)(n+2)}{2}$ parameters are determined by solving the following least-square problem with respect to the $\|.\|_2$ norm:

$$\min_{(\alpha_0, \mathbf{g}, \mathbf{H})} \|Q(X_{\text{sample}}) - f_{\text{bb}}(X_{\text{sample}})\|_2^2,$$

where $Q(X_{\text{sample}})$ and $f_{\text{bb}}(X_{\text{sample}})$ are defined as:

$$Q(X_{\text{sample}}) = \begin{bmatrix} Q(\mathbf{x}^1) \\ Q(\mathbf{x}^2) \\ \vdots \\ Q(\mathbf{x}^q) \end{bmatrix} \text{ and } f_{\text{bb}}(X_{\text{sample}}) = \begin{bmatrix} f_{\text{bb}}(\mathbf{x}^1) \\ f_{\text{bb}}(\mathbf{x}^2) \\ \vdots \\ f_{\text{bb}}(\mathbf{x}^q) \end{bmatrix}$$

with $q = |X_{\text{sample}}| \geq p$ and $X_{\text{sample}} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^q\}$ is a set of points that have previously been evaluated by the true function f_{bb}.

If the sample set has good properties, i.e., the so-called Λ_R-poisedness condition for quadratic regression ([25, Definition 4.7]), one can show that the error between the true function value and its quadratic regression model, as well as their gradients and Hessians is bounded by some relative constant depending on p, Λ_R and the radius of a ball containing the sample set X_{sample} ([25, Theorem 4.13]).

At iteration k, the quadratic model search step tries to construct $m+1$ quadratic regression models: one for a given scalarization single-objective formulation ψ function and one for each $c_j, j \in \mathcal{J}$. These models, respectively denoted as Q_ψ and $Q_{c_j}, j \in \mathcal{J}$, are supposed to be local approximations of their respective true functions. In a certain region $\mathcal{B}$, one could hope that these models satisfy:

$$Q_\psi(x) \approx \psi(\mathbf{x}) \text{ and } Q_{c_j}(\mathbf{x}) \approx c_j(\mathbf{x}), \text{ for all } j \in \mathcal{J}, \text{ for all } \mathbf{x} \in \mathcal{B},$$

and may be used to guide the search towards new interesting solutions.

At iteration k, consider one of the two incumbent iterate solutions and denote it by $\mathbf{x}^{k,c}$. The quadratic model search step starts by collecting a sample set of already evaluated points having finite objective values and constraints in a ball $\mathcal{B}_\infty(\mathbf{x}^{k,c}, \rho\Delta^k)$, i.e.,

$$X^k_{sample} \subset \mathcal{X} \cap V^k \cap \mathcal{B}_\infty(\mathbf{x}^{k,c}, \rho\Delta^k),$$

where $\rho > 1$ is a radius factor. If $q = |X^k_{\text{sample}}| < p = \frac{(n+1)(n+2)}{2}$, the procedure stops, because there are not enough points to build the model.

Otherwise, the quadratic model search step tries to construct the $m+1$ quadratic regression models using X^k_{sample}. If the least-square resolution fails for at least one model, the procedure stops. Note that the procedure does not assess the quality of the sample set.

Finally, using these models, the quadratic model search step solves the single-objective subproblem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} Q_\psi(\mathbf{x}) \text{ s.t. } \begin{cases} Q_{c_j}(\mathbf{x}) \leq 0, \ j \in \mathcal{J}, \\ \mathbf{x} \in \mathcal{X}, \\ \mathbf{x} \in \mathcal{B}_\infty(\mathbf{x}^{k,\mathrm{c}}, \rho\Delta^k). \end{cases}$$

The solution of these subproblems is then evaluated by the blackbox function after projection onto mesh M^k. Depending on the search strategy, ψ may correspond to:

- f_i for $i = 1, 2, \ldots, m$ and $\bar{\psi}_{r^k}$ for the MultiMADS search strategy;
- $\psi^{\mathrm{Do}}_{R^k}$ for the dominance move search strategy.

This subsection concludes with two remarks about the quadratic model search.

Remark 6.1. Contrary to the single-objective case [24], the quadratic model search only builds quadratic models around a current incumbent solution, corresponding to the primary frame center of iteration k [16] (which may be different from the current iterate incumbent). This is done for computational time considerations because the resolution of the quadratic subproblem may be costly.

Remark 6.2. When $\psi \in \{\bar{\psi}_{r^k}, \psi^{\mathrm{Do}}_{Y^k}\}$, the resulting model is not built by combining the quadratic models Q_{f_i}, but directly by computing a regression on ψ. Although this is less precise than the first option because structure is lost, it simplifies the resolution of the subproblem, as it is always a quadratically constrained quadratic problem with bounds on the variables. Furthermore, it improves computation time.

6.3.2 The Nelder-Mead search step

The NM search step for multiobjective optimization follows the procedure described in the single-objective counterpart of this work [14]. The definitions are first stated before adapting the strategy to multiobjective optimization problems. The NM strategy relies on a strict ordering of decision vectors evaluated by the algorithm.

Definition 6.1 (adapted from [7])**.** Let $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ a scalar-valued function and two decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ in $\mathcal{X}$. The decision vector $\mathbf{x}^1 \in \mathcal{X}$ is said to ψ-dominate the decision vector $\mathbf{x}^2 \in \mathcal{X}$ (denoted as $\mathbf{x}^1 \prec_\psi \mathbf{x}^2$) if

- both points are feasible and $\psi(\mathbf{x}^1) < \psi(\mathbf{x}^2)$;
- both points are infeasible and $\psi(\mathbf{x}^1) \leq \psi(\mathbf{x}^2)$ and $h(\mathbf{x}^1) \leq h(\mathbf{x}^2)$ with at least one inequality strictly satisfied.

The following definition is required to be able to distinguish two points that have the same ψ and h value.

Definition 6.2 (from [14])**.** A point $\mathbf{x}^1 \in \mathbb{R}^n$ is *older* than a point $\mathbf{x}^2 \in \mathbb{R}^n$ if it was generated before $\mathbf{x}^2$ by an algorithm. The function `older` : $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ defined as

$$\texttt{older}(\mathbf{x}^1, \mathbf{x}^2) = \begin{cases} \mathbf{x}^1 & \text{if } \mathbf{x}^1 \text{ is older than } \mathbf{x}^2, \\ \mathbf{x}^2 & \text{otherwise}, \end{cases}$$

returns the oldest of two points.

The NM search step depends on a comparison function defined as follows.

Definition 6.3 (adapted from [14]). Let $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a scalar-valued function. The function $\texttt{best}_\psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$

$$\texttt{best}_\psi(\mathbf{x}^1, \mathbf{x}^2) = \begin{cases} \mathbf{x}^1 & \text{if } \mathbf{x}^1 \prec_\psi \mathbf{x}^2 \text{ or } h(\mathbf{x}^1) < h(\mathbf{x}^2), \\ \mathbf{x}^2 & \text{if } \mathbf{x}^2 \prec_\psi \mathbf{x}^1 \text{ or } h(\mathbf{x}^2) < h(\mathbf{x}^1), \\ \texttt{older}(\mathbf{x}^1, \mathbf{x}^2) & \text{if } \psi(\mathbf{x}^1) = \psi(\mathbf{x}^2) \text{ or } h(\mathbf{x}^1) = h(\mathbf{x}^2), \end{cases}$$

returns the best of two points $\mathbf{x}^1$ and $\mathbf{x}^2$ in $\mathbb{R}^n$. The point $\mathbf{x}^1$ is said to be better than $\mathbf{x}^2$ relatively to ψ if $\mathbf{x}^1 = \texttt{best}_\psi(\mathbf{x}^1, \mathbf{x}^2)$.

Using [14, Proposition 4.3], one can show that $\texttt{best}_\psi$ is transitive over the set of trial points.

Algorithm 5 offers a high-level description of the NM subproblem procedure for a particular function ψ. It is described next in more details.

Algorithm 5 A high-level description of the NM subproblem procedure using ψ (adapted from [14]).

Construct an initial ordered simplex $\mathbb{X}_s$. Using the set $\mathbb{T}_{\pi_{\text{radius}}}$ and the function $\texttt{best}_\psi$, try to construct an ordered simplex $\mathbb{X}_s \subseteq \mathbb{T}_{\pi_{\text{radius}}}$. If $|\mathbb{X}_s| \neq n+1$, stop the search.
Update the simplex. If $\mathbb{X}_s$ is not a simplex, go to *Termination*. Otherwise, reorder $\mathbb{X}_s = \{\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n\}$ using $\texttt{best}_\psi$.
Determine a new candidate vertex t. Generate Nelder-Mead candidates $\{\mathbf{x}^{\text{r}}_\oplus, \mathbf{x}^{\text{e}}_\oplus, \mathbf{x}^{\text{ic}}_\oplus, \mathbf{x}^{\text{oc}}_\oplus\} \subset M^k$. Following [14, Algorithm 4], if $\mathbb{X}_s$ is to be shrunk, go to *Termination*. Otherwise, update $\mathbf{t} \in \{\mathbf{x}^{\text{r}}_\oplus, \mathbf{x}^{\text{e}}_\oplus, \mathbf{x}^{\text{ic}}_\oplus, \mathbf{x}^{\text{oc}}_\oplus\}$.
Replace the worst point. If $\mathbf{t} \in V^k$, go to *Termination*. Otherwise, set $\mathbf{x}^n := \mathbf{t}$ and go to *Update the simplex*.
Termination: Collect all solution points $S^{\text{tmp}} \subset \mathcal{X} \cap M^k$ evaluated by the NM subproblem procedure. Set $S^k := S^k \cup S^{\text{tmp}}$. Return.

At iteration k, given the incumbent iterate $\mathbf{x}^k, \Delta^k \in L^k$, and the cache V^k, $\mathbb{T}_{\pi_{\text{radius}}}$ is defined as:

$$\mathbb{T}_{\pi_{\text{radius}}} = \{\mathbf{x} \in V^k : \|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \pi_{\text{radius}} \Delta^k\},$$

where $\pi_{\text{radius}} \geq 1$ is a parameter controlling the size of the zone in which points can be collected. Using $\texttt{best}_\psi$, the NM procedure tries to construct an ordered simplex $\mathbb{X}_s = \{\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n\} \subseteq \mathbb{T}_{\pi_{\text{radius}}}$ such that $\mathbf{x}^{j-1} = \texttt{best}_\psi(\mathbf{x}^{j-1}, \mathbf{x}^j)$ for $j = 1, 2, \ldots, n$.

If $\mathbb{X}_s$ is valid, it contains $n+1$ elements. The NM procedure constructs the following points from $\mathbb{X}_s$ using the standard NM algorithm:

$$\begin{cases} \mathbf{x}^{\text{c}} = \frac{1}{n}\sum_{j=0}^{n-1} \mathbf{x}^j & \text{the centroid of the } n \text{ best points;} \\ \mathbf{x}^{\text{r}} = \mathbf{x}^{\text{c}} + (\mathbf{x}^{\text{c}} - \mathbf{x}^n) & \text{the reflection candidate;} \\ \mathbf{x}^{\text{e}} = \mathbf{x}^{\text{c}} + \delta^{\text{e}}(\mathbf{x}^{\text{c}} - \mathbf{x}^n) & \text{the expansion candidate with } \delta^{\text{e}} \in]1, +\infty[; \\ \mathbf{x}^{\text{oc}} = \mathbf{x}^c + \delta^{\text{oc}}(\mathbf{x}^c - \mathbf{x}^n) & \text{the outside-contraction candidate with } \delta^{\text{oc}} \in]0, 1[\text{ and} \\ \mathbf{x}^{\text{ic}} = \mathbf{x}^c + \delta^{\text{ic}}(\mathbf{x}^{\text{c}} - \mathbf{x}^n) & \text{the inside-contraction candidate with } \delta^{ic} \in]-1, 0[. \end{cases}$$

The NM procedure uses the projections of $\mathbf{x}^{\text{r}}$, $\mathbf{x}^{\text{e}}$, $\mathbf{x}^{\text{oc}}$, and $\mathbf{x}^{\text{ic}}$ on the mesh M^k, respectively $\mathbf{x}^{\text{r}}_\oplus$, $\mathbf{x}^{\text{e}}_\oplus$, $\mathbf{x}^{\text{oc}}_\oplus$, and $\mathbf{x}^{\text{ic}}_\oplus$ as potential candidates to evaluate.

The NM procedure performs a succession of iterative steps, where it tries to replace the worst point $\mathbf{x}^n$ of the simplex $\mathbb{X}_s$ by a new candidate $\mathbf{t} \in \{\mathbf{x}^{\text{r}}_\oplus, \mathbf{x}^{\text{e}}_\oplus, \mathbf{x}^{\text{ic}}_\oplus, \mathbf{x}^{\text{oc}}_\oplus\}$. This substitution procedure is detailed in [14].

The NM procedure stops when:

- it tries to evaluate a point already generated during the previous iterations, i.e., $\mathbf{t} \in V^k$;
- $\mathbb{X}_s$ is not a simplex anymore, due to the projection on the mesh;
- when the NM procedure tries to enter in the shrinking phase of the standard NM algorithm [14] or;
- after the NM procedure reaches a certain number of blackbox evaluations.

The NM search collects all evaluated points as potential candidates to update the iterate list. For more details, the reader is referred to [14].

7 Computational experiments

This section is dedicated to the numerical experiments of new search strategies for DMulti-MADS. The first part presents the considered variants of DMulti-MADS. The second part evaluates the performance of these variants on (bound-)constrained analytical benchmarks using data profiles for multiobjective optimization. The last part shows the benefit of such search strategies on two biobjective [4] and one triobjective [13] engineering applications using convergence profiles.

This work assesses the performance of multiobjective DFO algorithms by adapting data [47] and convergence profiles to the multiobjective context [15, 16]. Both tools require a convergence test based on the hypervolume indicator [59].

Given a Pareto front approximation $Y_N \subset \mathbb{R}^m$, the hypervolume indicator of Y_N is the measure of the space in the objective space dominated by Y_N and bounded above by a objective vector $\mathbf{u} \in \mathbb{R}^m$ such that for all $\mathbf{y} \in Y_N$, $\mathbf{y} \leq \mathbf{u}$. Formally,

$$HV(Y_N, \mathbf{r}) = \Lambda\left(\{\mathbf{v} \in \mathbb{R}^m \mid \exists \mathbf{y} \in Y_N : \mathbf{y} \leq \mathbf{v} \text{ and } \mathbf{v} \leq \mathbf{u}\}\right) = \Lambda\left(\bigcup_{\mathbf{y} \in Y_N} [\mathbf{y}, \mathbf{u}]\right),$$

where $\Lambda(.)$ denotes the Lebesgue measure of a m-dimensional set of points. Pareto-compliant with the dominance order [60] and intuitive to understand, the hypervolume indicator captures several properties of a Pareto front approximation, such as cardinality, convergence, spread, and extent. All these points make the hypervolume indicator a reasonable choice for assessing the quality of a Pareto front approximation.

Given a set of problems $\mathcal{P}$, the convergence test for a problem $p \in \mathcal{P}$ requires a Pareto front approximation reference Y^p. In a DFO context, the analytical Pareto front is not available. Y^p is then constructed by taking the union of the best feasible non-dominated points found by all considered solvers on problem $p \in \mathcal{P}$ for a maximum budget of function evaluations, from which new dominated points are removed. If no solver manages to find a feasible solution, $p \in \mathcal{P}$ is discarded from $\mathcal{P}$. Once Y^p is computed, the approximated ideal objective vector

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

and the approximated nadir objective vector

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

are extracted, with m_p the number of objectives associated to problem $p \in \mathcal{P}$.

Let Y^e be the Pareto front approximation obtained by a deterministic algorithm on problem $p \in \mathcal{P}$ after e blackbox evaluations. To avoid privileging one objective against another, one applies the transformation $T : \mathbb{R}^m \to \mathbb{R}^m$ to each element of Y^e and Y^p, and $\tilde{\mathbf{y}}^{\mathrm{N},p}$, defined as

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

where $\oslash$ is the Hadamard division operator.

Given a Pareto front approximation Y, let denote

$$T(Y) = \{\mathbf{v} \in \mathbb{R}^m : \mathbf{v} = T(\mathbf{y}) \text{ for } \mathbf{y} \in Y\}.$$

A computational problem $p \in \mathcal{P}$ is said to be solved by an algorithm with tolerance $\varepsilon_\tau > 0$ if

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

Note that all elements of $T(Y^e)$ dominated by $T(\tilde{\mathbf{y}}^{\mathrm{N},p})$ do not contribute to the hypervolume and need to be removed before the computation. If the solver does not find at least one transformed objective vector that dominates $T(\tilde{\mathbf{y}}^{\mathrm{N},p})$, $HV\left(T(Y^e), T(\tilde{\mathbf{y}}^{\mathrm{N},p})\right)$ is set to 0.

Data profiles show the proportion of computational problems solved by an algorithm according to the number of groups of $n+1$ function evaluations, i.e., the number of points required to build a gradient simplex in $\mathbb{R}^n$.

7.1 Tested DMulti-MADS variants

The considered variants are:

- DMulti-MADS `basic`: DMulti-MADS with an only speculative search, as described in [15, 16].
- DMulti-MADS `NM-DoM`: DMulti-MADS `basic` with the NM dominance move search strategy.
- DMulti-MADS `NM-Multi`: DMulti-MADS `basic` with the NM MultiMADS search strategy.
- DMulti-MADS `Quad-DoM`: DMulti-MADS `basic` with the quadratic dominance move search strategy.
- DMulti-MADS `Quad-Multi`: DMulti-MADS `basic` with the quadratic MultiMADS search strategy.

This work implements an additional quadratic search strategy adapted from [19] called DMulti-MADS `Quad-DMS`. The procedure is given in Algorithm 6.

Algorithm 6 A high-level description of the quadratic DMS search strategy (adapted from [19]).

```
Initialization. Given a current frame incumbent center $\mathbf{x}^k$, its associated frame size parameter $\Delta^k > 0$, $\rho > 1$) the radius factor, select a set of points $V^{Q,k} \subseteq V^k \cap B_\infty(\mathbf{x}^k, \rho\Delta^k)$. Initialize the counter $l = 0$.
while $l < m$ do
    Set $l := l + 1$. Let $\mathcal{I}$ the set of all combinations of $l$ quadratic models taken from the total set of $m$ models. Set $S := \emptyset$.
    for $I \in \mathcal{I}$ do
        Approximately solve
            $\mathbf{s}_I \approx \arg\min_{\mathbf{x} \in \mathbb{R}^n} Q_{\psi_I}(\mathbf{x})$ s.t. $\begin{cases} Q_{c_j}(\mathbf{x}) \leq 0, \ j \in \mathcal{J}, \\ \mathbf{x} \in \mathcal{X}, \\ \mathbf{x} \in \mathcal{B}_\infty(\mathbf{x}^{k,\mathrm{c}}, \rho\Delta^k). \end{cases}$
        where $Q_{\psi_I}$ is the quadratic model associated to $\psi_I : \mathbf{x} \mapsto \max_{i \in I} f_i(\mathbf{x})$, built from $V^{Q,k}$. Project $\mathbf{s}_I$ onto $M^k$ and update $S := S \cup \{\mathbf{s}_I\}$.
    end for
    Check for success. Evaluate $f$ and $h$ at each point of $S$. Update $V^{Q,k} := V^{Q,k} \cup S$ and $S^k := S^k \cup S$. Compute $L^{\mathrm{trial}}$ by removing all dominated points from $L^k \cup \left\{(\mathbf{s}, \Delta) : \mathbf{s} \in S \text{ and } \Delta \in \left\{\Delta^k, \tau^{-1}\Delta^k\right\}\right\}$. If $L^{\mathrm{trial}} \neq L^k$, return.
end while
```

At iteration k, the quadratic DMS search procedure incrementally solves successive quadratically constrained quadratic subproblems, starting from the minimization of each individual objective quadratic model (i.e., the level $l = 1$) until the minimization of all objective quadratic models simultaneously (i.e., the level $l = m$). When all objective combinations for a given level l are exhausted, the quadratic DMS search procedure checks whether a new non-dominated solution has been generated. If so, the procedure stops. Otherwise, the procedure moves to the next level. In the worst case, this search iterates over all levels, resulting in the optimization of $2^m - 1$ subproblems. This issue remains limited for a small number of objectives, i.e., $m \leq 4$.

Note that contrary to [19], a success for the quadratic DMS search is not equivalent to a success for the complete DMulti-MADS iteration. In particular, DMulti-MADS may continue with another search step or poll step even if the quadratic DMS search is successful.

All of these search strategies are implemented in the NOMAD software [11], version `4.5.0`, which can be found at `https://github.com/bbopt/nomad`. For the rest of this work, all variants use an opportunistic strategy and an OrthoMads poll step with $n+1$ directions without quadratic models [9]. The search strategies use the same default parameters as their single-objective counterparts implemented in NOMAD. For example, all quadratically constrained quadratic subproblems are solved with MADS.

An implementation of the benchmark functions used in the next subsection can be found at https://github.com/bbopt/DMultiMads_search_benchmarks.

7.2 Computational experiments on synthetic benchmarks

In this subsection, this work considers two analytical benchmark sets:

- a set of 100 bound-constrained multiobjective optimization problems taken from [27], with $n \in [1, 30]$ and $m \in \{2, 3, 4\}$;
- a set of 214 constrained multiobjective optimization problems proposed in [43], based on the previous benchmark suite, with $n \in [3, 30]$, $m \in \{2, 3, 4\}$ and $|\mathcal{J}| \in [3, 30]$.

For each problem $p \in \mathcal{P}$, each variant is run once with a maximal budget of $30,000$ blackbox evaluations, starting from the same n_p points by dividing the line connecting the lower bound and upper bound into n_p equal spaces [27], with n_p the number of variables of problem $p \in \mathcal{P}$.

The data profiles in Figure 5 show that the addition of such search strategies has a positive impact on the performance of DMulti-MADS for all tolerances $\varepsilon_\tau \in \{10^{-2}, 5\times 10^{-2}, 10^{-1}\}$. The NM dominance move search strategy is the most efficient on the bound-constrained benchmark suite for all tolerances for a medium to large budget of blackbox evaluations. It solves up to $8 - 10\%$ more problems for the lowest tolerance $\varepsilon_\tau = 10^{-2}$ and about 5% more problems for $\varepsilon_\tau \in \{5 \times 10^{-2}, 10^{-1}\}$ than DMulti-MADS `basic`. It is followed by DMulti-MADS `Quad-DoM` and `Quad-DMS`, which perform better than DMulti-MADS `basic` for medium to high tolerance ε_τ considered.

This performance gain is more noticeable when solving constrained analytical problems. As shown in Figure 6, all variants with quadratic or NM search strategies enabled perform better than DMulti-MADS `basic` for all evaluations budgets for medium ($\varepsilon_\tau = 5 \times 10^{-2}$) to high tolerance ($\varepsilon_\tau = 10^{-1}$), up to 5% more problems solved for the less efficient search strategy. Note, however, that DMulti-MADS has some difficulty with this benchmark set, solving less than 60% of the problems for the lowest tolerance $\varepsilon_\tau = 10^{-1}$. DMulti-MADS `Quad-DMS` is slightly more efficient than the other variants for a medium to large budget of evaluations in this case for all considered tolerances.

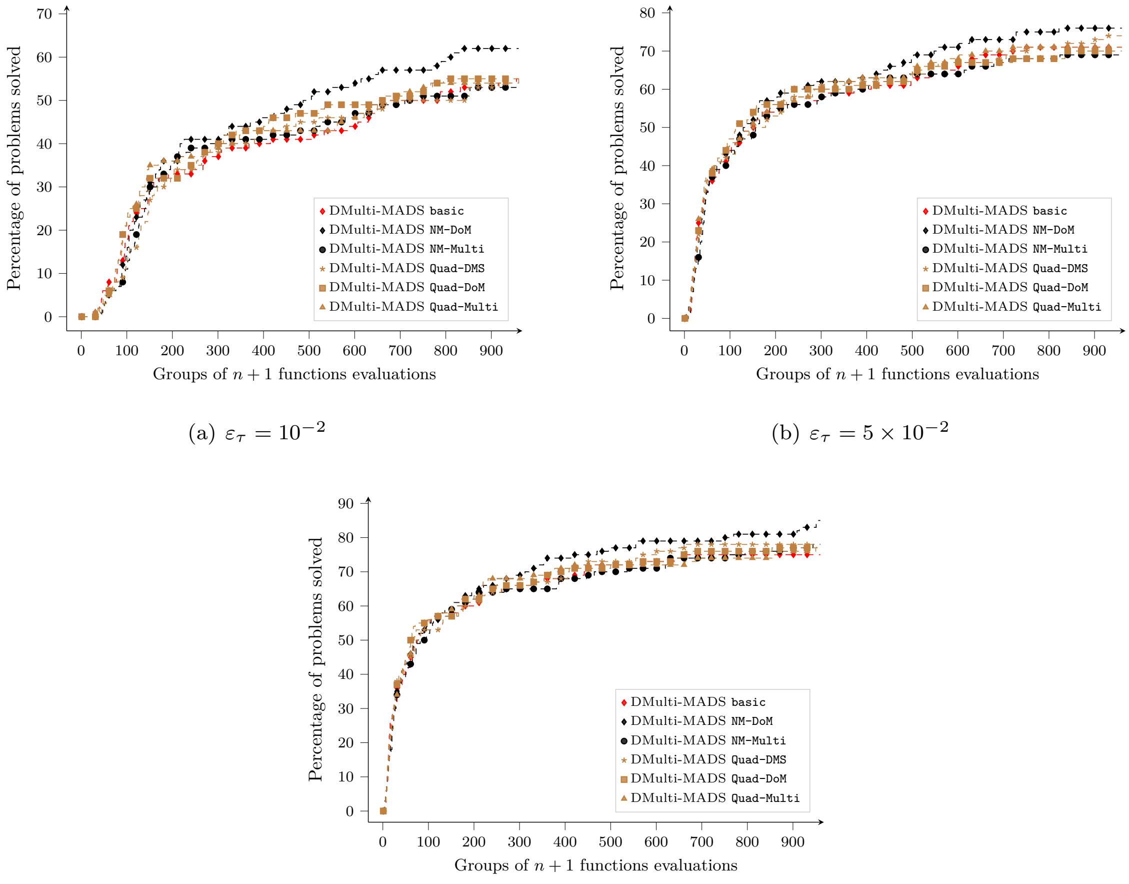
7.3 Computational experiments on practical engineering problems

To validate the performance of the new search strategies, this work considers three “real-world” engineering applications: two biobjective problems, solar 8.1 and solar 9.1 [4], and one triobjective application STYRENE [13]. To follow the progression of the different DMulti-MADS variants, this work uses convergence profiles for multiobjective optimization. Because the evaluation of such blackboxes is more expensive than for previous benchmark sets, computing data profiles cannot be performed in a reasonable time for this research.

Convergence profiles track the progression of a multiobjective optimization algorithm along the iterations by computing the following normalized hypervolume value

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

as a function of the number of blackbox evaluations e, where Y^p is the Pareto front reference for problem p, and Y^e is the Pareto front approximation obtained at evaluation e by the algorithm. The normalized hypervolume indicator is monotonically increasing and bounded above by 1. As mentioned previously at the beginning of Section 7, a normalized hypervolume value equal to 0 indicates that a method has not generated a feasible objective vector that is dominated by the approximated nadir objective vector $\tilde{\mathbf{y}}^{\mathrm{N},p}$ of problem p.



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

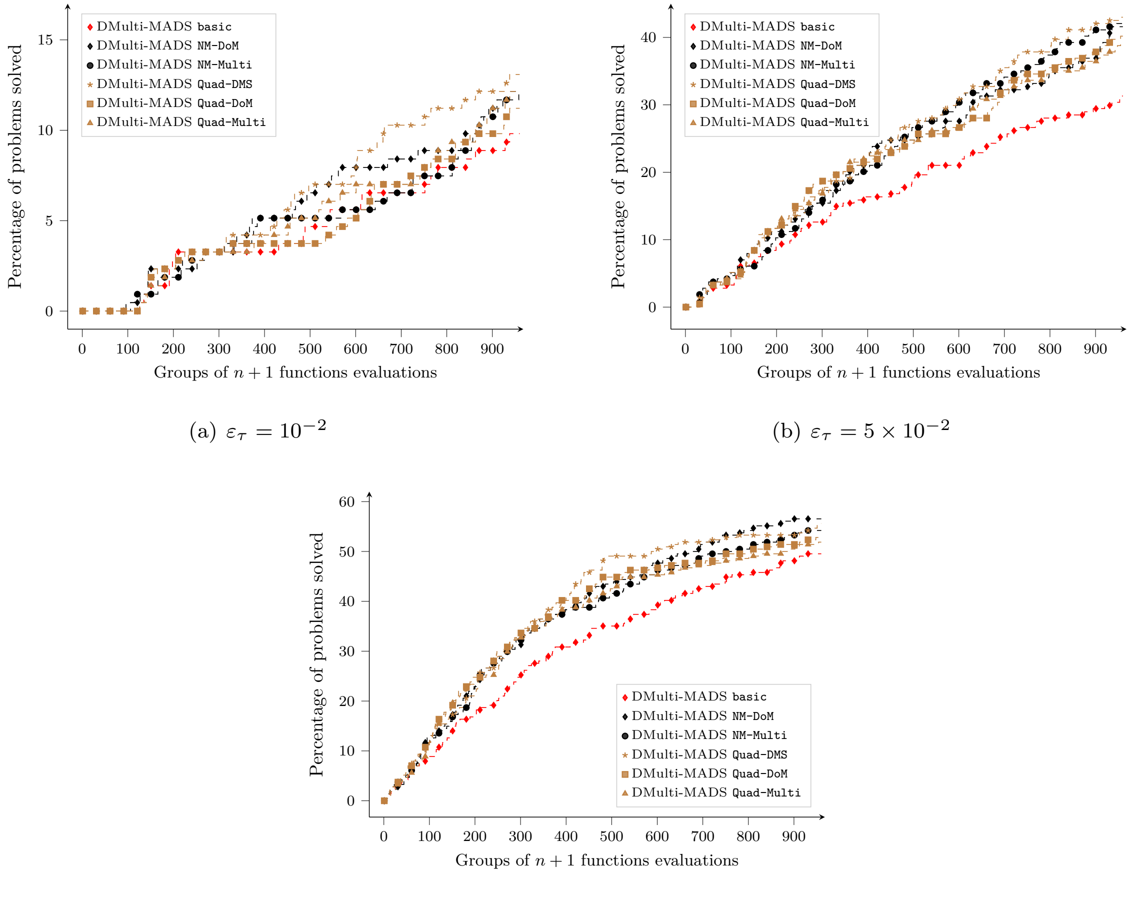
(b) $\varepsilon_\tau = 5 \times 10^{-2}$

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

Figure 5: Data profiles obtained from 100 multiobjective bound-constrained analytical problems taken from [27] for DMulti-MADS `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, and `Quad-Multi` variants with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

solar 8.1 and solar 9.1 are two deterministic biobjective problems derived from the design of a solar plant system modeled by a numerical simulation [4]. The simulator consists of three subsystems. The heliostats field collects solar rays that are sent to a receiver. The second subsystem converts the heat generated by the receiver into thermal energy. The power-block uses this energy to produce water steam, which rotates turbines to generate electricity. Interested readers are referred to [4] for more details. solar 8.1 aims to maximize heliostats field performance and minimize cost. It possesses 13 variables, 2 of which are integer. solar 9.1 aims to maximize power production and minimize cost. It possesses 29 variables, 7 of which are integer.

DMulti-MADS does not handle integer variables. Then, for both problems, integer values are kept constant to their initial value (they can be found at [16]), resulting in blackboxes with 11 and 22 continuous input variables. All variants are run with a budget of 5,000 evaluations and start from the same infeasible initial point.



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

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

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

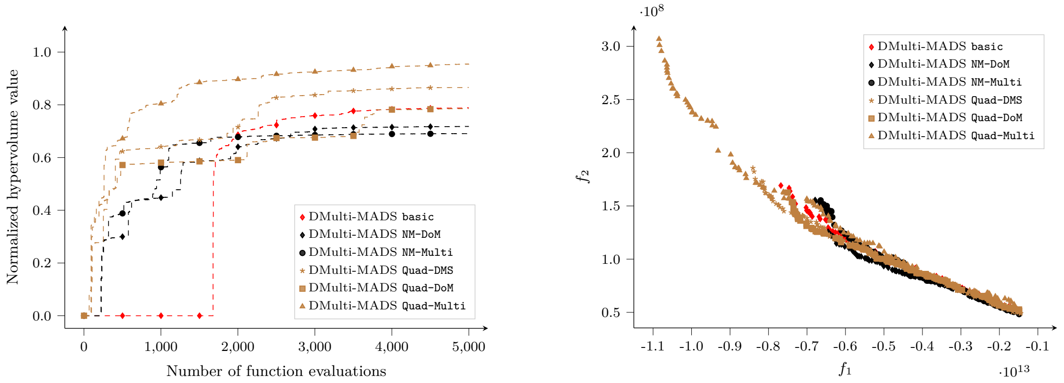
Figure 6: Data profiles obtained from 214 multiobjective analytical problems taken from [43] for DMulti-MADS `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, and `Quad-Multi` variants with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

From Figure 7(a), DMulti-MADS with the quadratic MultiMADS search strategy performs better than the other variants, followed by the quadratic DMS search strategy. Turning to Figure 7(b), the quadratic MultiMADS search manages to extend the Pareto front approximation towards f_1, by exploiting the nature of the current frame incumbent. The quadratic DMS search strategy is slower to extend the Pareto front approximation in this case, because it is myopic in its exploration of the objective space.

Results for solar 9.1 are presented in Figure 8. The two dominance move search strategies are the most efficient, as illustrated in Figure 8(a). The Pareto front approximation in Figure 8(b) is “flat”, which could favor such strategies.

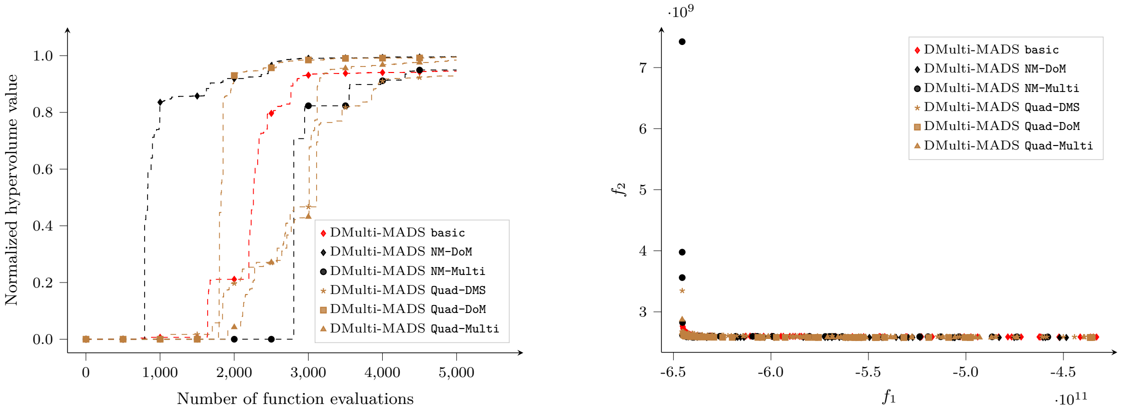
The last problem considered in this subsection is STYRENE [13]. It is based on a numerical simulator that models styrene production, implemented in C++, and available at `github.com/bbopt/styrene`.

The chemical production of styrene involves four steps: reactants preparation, catalytic reactions, and two distillation phases. The first distillation allows the recovery of styrene, while the second is aimed at the recovery of benzene. During this second distillation, one can recycle unreacted ethylbenzaline, which can be reinjected into the styrene production as an initial reactant. The goal is to maximize three objectives simultaneously: the net present value associated with the process (f_1), the purity of the styrene produced (f_2) and the overall conversion of ethylbenzene to styrene (f_3). This problem has eight continuous variables and nine inequality constraints, related to industrial regulations and economic context. Interested readers are referred to [13] for more details. The simulation may crash, i.e., it may not return finite numerical output values, due to the presence of hidden constraints in the blackbox.



(a) (b)

Figure 7: (a) On the left, convergence profiles for the solar 8.1 **problem using DMulti-MADS** `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, **and** `Quad-Multi` **variants for a maximal budget of** $5,000$ **evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of** solar 8.1 **for DMulti-MADS** `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, **and** `Quad-Multi` **in the objective space.**

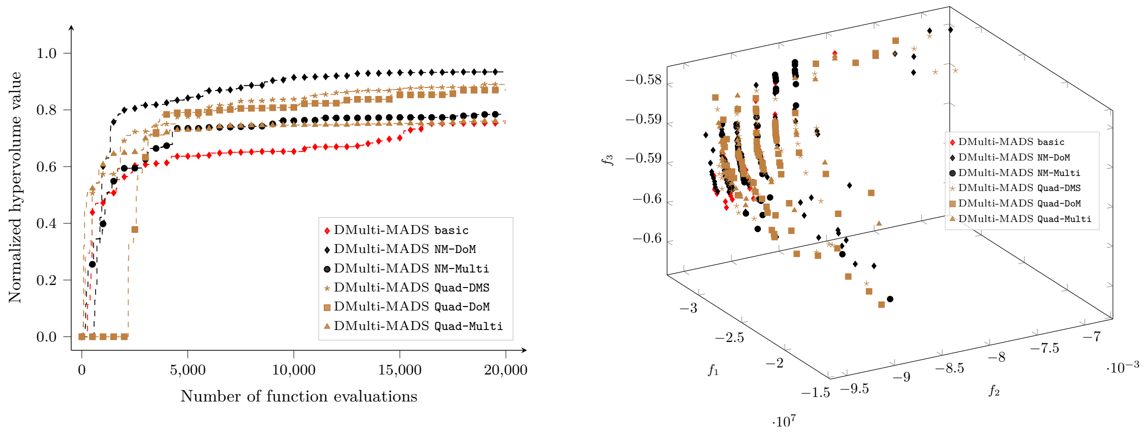


(a) (b)

Figure 8: (a) On the left, convergence profiles for the solar 9.1 **problem using DMulti-MADS** `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, **and** `Quad-Multi` **variants for a maximal budget of** $5,000$ **evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of** solar 9.1 **for DMulti-MADS** `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, **and** `Quad-Multi` **in the objective space.**

All DMulti-MADS variants are affected a maximal budget of $20,000$ blackbox evaluations and start from the same initial feasible point.

From Figure 9(a), one can see that all variants of DMulti-MADS with search strategies outperform DMulti-MADS `basic`. The dominance move search strategies are the most efficient. The Pareto front approximations obtained on Figure 9(b) show that all search strategies introduced in this work are able to capture a larger part of the Pareto front reference than DMulti-MADS standard algorithm.



(a) (b)

Figure 9: (a) On the left, convergence profiles for the STYRENE problem using DMulti-MADS `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, and `Quad-Multi` variants for a maximal budget of $5,000$ evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of STYRENE for DMulti-MADS `basic`, `NM-DoM`, `NM-Multi`, `Quad-DMS`, `Quad-DoM`, and `Quad-Multi` in the objective space.

8 Closing remarks

This work proposes two search strategies for the DMulti-MADS constrained algorithm for multiobjective optimization, that generalizes previous heuristics proposed for the MADS algorithm [14, 24, 28]. The first uses quadratic models, built from previous evaluated points, that act as surrogates of the objective and constraint functions, providing new promising candidates. The other relies on a Nelder-Mead-based sampling strategy. Both strategies are built around the resolution of single-objective subproblems derived from the original multiobjective problem, in the lineage of [12, 13, 19]. This work also introduces a single-objective formulation framework that generalizes [12, 13].

Experiments on (bound-)constrained analytical benchmarks and three engineering problems show that the search strategies bring significant improvement over the baseline version of the DMulti-MADS algorithm. They also highlight the fact that none of these strategies is superior for all problems.

NOMAD chooses as default search strategies `NM-DoM` and `Quad-Multi` because they allow a gain without sacrificing computational time (e.g., in solving quadratic subproblems). A user may modify these default options to obtain better results according to his problem.

We emphasize that these search strategies are not limited to DMulti-MADS and can also be implemented for other direct search algorithms for multiobjective optimization, such as DMS [27]. Future work includes the exploitation of parallelism, in the continuation of [54], and the integration of the single-objective formulations in the construction of smarter poll strategies based on quadratic models [9].

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] T. Akhtar and C.A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. Journal of Global Optimization, 64(1):17–32, 2016.

[3] 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.

[4] N. Andrés-Thió, C. Audet, M. Diago, A.E. Gheribi, S. Le Digabel, X. Lebeuf, M. Lemyre Garneau, and C. Tribes. `solar`: A solar thermal power plant simulator for blackbox optimization benchmarking. Technical Report G-2024-37, Les cahiers du GERAD, 2025. To appear in Optimization and Engineering.

[5] 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.

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

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

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

[9] 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.

[10] C. Audet, M. Kokkolaras, S. Le Digabel, and B. Talgorn. Order-based error for managing ensembles of surrogates in mesh adaptive direct search. Journal of Global Optimization, 70(3):645–675, 2018.

[11] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm. ACM Transactions on Mathematical Software, 48(3):35:1–35:22, 2022.

[12] 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.

[13] 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.

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

[15] 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.

[16] J. Bigeon, S. Le Digabel, and L. Salomon. Handling of constraints in multiobjective blackbox optimization. Computational Optimization and Applications, 89(1):69–113, 2024.

[17] E. Bradford, A.M. Schweidtmann, and A. Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. Journal of Global Optimization, 71(2):407–438, 2018.

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

[19] 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.

[20] A. Bűrmen, J. Olenšek, and T. Tadej. Mesh adaptive direct search with second directional derivative-based Hessian update. Computational Optimization and Applications, 62(3):693–715, 2015.

[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. Three new metrics to measure the convergence of metaheuristics towards the Pareto frontier and the aesthetic of a set of solutions in biobjective optimization. Computers and Operations Research, 32(4):773–792, 2005.

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

[24] 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.

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

[26] 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.

[27] 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.

[28] A.L. Custódio, H. Rocha, and L.N. Vicente. Incorporating minimum Frobenius norm models in direct search. Computational Optimization and Applications, 46(2):265–278, 2010.

[29] A.L. Custódio and L.N. Vicente. Using Sampling and Simplex Derivatives in Pattern Search Methods. SIAM Journal on Optimization, 18(2):537–555, 2007.

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

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

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

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

[38] J. Jahn. Introduction to the theory of nonlinear optimization. Springer, third edition, 2007.

[39] C.T. Kelley. Implicit Filtering. Society for Industrial and Applied Mathematics, Philadephia, PA, 2011.

[40] T.G. Kolda, R.M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review, 45(3):385–482, 2003.

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

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

[43] 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.

[44] V.K. Mehta and B. Dasgupta. A Multi-objective Optimization Method Based on Nelder–Mead Simplex Search Method. In H.K. Voruganti, K.K. Kumar, P.V. Krishna, and X. Jin, editors, Advances in Applied Mechanical Engineering, pages 655–668, Singapore, 2020. Springer Singapore.

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

[46] A. Mohammadi, D. Hajinezhad, and A. Garcia. A trust-region approach for computing pareto fronts in multiobjective derivative-free optimization. Optimization Letters, 19:233–266, 2025.

[47] J.J. Moré and S.M. Wild. Benchmarking Derivative-Free Optimization Algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

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

[49] S. Prinz, J. Thomann, G. Eichfelder, T. Boeck, and J. Schumacher. Expensive multi-objective optimization of electromagnetic mixing in a liquid metal. Optimization and Engineering, 22(2):1065–1089, 2020.

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

[51] R.G. Regis. On the Convergence of Adaptive Stochastic Search Methods for Constrained and Multi-objective Black-Box Optimization. Journal of Optimization Theory and Applications, 170(3):932–959, 2016.

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

[53] B. Talgorn, C. Audet, M. Kokkolaras, and S. Le Digabel. Locally weighted regression models for surrogate-assisted design optimization. Optimization and Engineering, 19(1):213–238, 2018.

[54] S. Tavares, C.P. Brás, A.L. Custódio, V. Duarte, and P. Medeiros. Parallel strategies for Direct Multisearch. Numerical Algorithms, 92(3):1757–1788, 2023.

[55] J. Thomann. A trust region approach for multi-objective heterogeneous optimization. PhD thesis, Technische Universität Ilmenau, 2019.

[56] J. Thomann and G. Eichfelder. A Trust-Region Algorithm for Heterogeneous Multiobjective Optimization. SIAM Journal on Optimization, 29(2):1017–1047, 2019.

[57] S. Zapotecas-Martínez and C.A. Coello Coello. MONSS: A multi-objective nonlinear simplex search approach. Engineering Optimization, 48(1):16–38, 2015.

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

[59] 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.

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