Exploring polynomial models in the Search Step of Direct Multisearch

Ana L. Custódio* Marta Pozzi † Everton J. Silva‡

Abstract

Direct Multisearch (DMS) is a class of direct-search algorithms designed for multiobjective derivative-free optimization. Its framework consists of an optional search step and a poll step, the latter ensuring the corresponding theoretical convergence properties. Recently, a search strategy based on the minimization of quadratic polynomial models, constructed from previously evaluated points, was proposed to improve the numerical efficiency of the method. While the construction of these surrogate models has been investigated, considerably less attention has been devoted to how they should be jointly minimized, with min-max scalarization typically being adopted.

This work investigates how different model-minimization strategies influence the performance of DMS. To this end, alternative strategies for exploring the quadratic polynomial models within the search step of DMS are proposed and numerically assessed, including one based on the recently proposed Improved Front Steepest Descent algorithm.

Keywords: Multiobjective optimization; Derivative-free optimization; Direct search methods; Quadratic interpolation; Pareto front approximation.

AMS Classification: 90C29, 90C30, 90C56, 65K05.

1 Introduction

Multiobjective optimization is a challenging scientific domain, relevant both from theoretical and practical perspectives, in which several conflicting objective functions must be optimized

*Center for Mathematics and Applications (NOVA Math), Campus de Caparica, 2829-516 Caparica, Portugal (alcustodio@fct.unl.pt). Support for this author was provided by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) under the scope of the projects UID/00297/2025 and UID/PRR/00297/2025.

†Computer Science Department, Bocconi University, Milan, Italy (marta.pozzi2@phd.unibocconi.it)

‡Center for Mathematics and Applications (NOVA Math), Campus de Caparica, 2829-516 Caparica, Portugal (ejo.silva@alumni.fct.unl.pt). Support for this author was provided by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) under the scope of the projects UID/00297/2025 and UID/PRR/00297/2025.

simultaneously. Problems of this type arise in a wide range of applications [1, 9, 21, 23, 26, 29].

In this work, we consider the following multiobjective derivative-free optimization problem:

$$\min \; F(x) = \big(f_1(x), \ldots, f_q(x)\big)^\top \text{ s.t. } x \in \Omega \tag{1}$$

where $q \geq 2$, $\Omega \subseteq \mathbb{R}^n$ is the feasible region, and $F : \Omega \subseteq \mathbb{R}^n \to (\mathbb{R} \cup \{+\infty\})^q$. It is assumed that the derivatives of each f_i, $i \in I := \{1, \ldots, q\}$ are neither available nor can be reliably approximated. This setting is common in simulation-based optimization, where function evaluations arise from complex and computationally expensive simulations. In such situations, objective functions are often nonsmooth, noisy, or unreliable, motivating the use of derivative-free methods. Comprehensive reviews of single-objective derivative-free optimization methods can be found in [4, 13].

Standard approaches for multiobjective derivative-free optimization include scalarization techniques [6, 5], which transform the original problem into a sequence of single-objective subproblems that can be addressed using classical nonlinear optimization techniques. BiMADS [6], designed for biobjective optimization, generates a sequence of scalarized reformulations solved by MADS [3], exploiting the fact that Pareto points can be ordered in the two-objective setting. To extend this idea to problems with more than two objectives, MULTIMADS was introduced in [5], combining MADS with the Normal Boundary Intersection (NBI) framework [17] to approximate the Pareto front. Although these methods benefit from the robustness of MADS in derivative-free settings, BiMADS is restricted to biobjective problems, and a practical implementation of MULTIMADS is not publicly available.

In [12], an unidirectional method was proposed that generalizes implicit filtering to bound-constrained multiobjective derivative-free optimization. At each iteration, the algorithm constructs a simplex-based Jacobian to approximate a multiobjective steepest-descent direction, which is then explored via a line search. Line search strategies were also considered in [25], for general nonlinear constrained multiobjective problems, when proposing DFMO, a derivative-free algorithm based on an exact-penalty approach combined with sufficient-decrease line searches.

Unlike scalarization-based approaches, the methods proposed in [7, 12, 15, 25] rely directly on the concept of Pareto dominance and avoid aggregation schemes. In [15], the authors introduce Direct Multisearch (DMS), a generalization of directional direct search to multiobjective derivative-free optimization. DMS was the first practical derivative-free method with theoretical convergence guarantees capable of approximating the entire Pareto front, and it has served as the foundation for subsequent developments.

Building on the ideas in [5, 6, 15], DMultiMADS was proposed in [7], a MADS-based extension for bound-constrained multiobjective optimization. Unlike BiMADS and MULTIMADS, DMultiMADS does not reduce the problem to a sequence of scalarized single-objective subproblems. Instead, it maintains and updates a list of nondominated points at each iteration, in the spirit of DMS. Although DMultiMADS shares conceptual similarities with DMS, it differs in the selection of incumbent points at each iteration, which impacts its practical performance.

As a directional direct-search method, each iteration of DMS [15] follows the search-poll paradigm, where the search step is optional and convergence is ensured by the poll step. In [8],

the authors introduced BoostDMS, a variant of DMS that incorporates quadratic polynomial interpolation and regression models into the search step to enhance performance.

BoostDMS reuses previously evaluated points to construct local quadratic surrogate models for each objective function, and its search step relies on the successive minimization of combinations of these surrogates. This approach extends to the multiobjective setting the strategy originally developed in [16] for single-objective directional direct search, while preserving the theoretical convergence properties of DMS. Since the search step determines trial points evaluated by the algorithm, the procedure adopted for jointly minimizing the surrogate models may have a significant impact on the efficiency of the overall method.

Although considerable attention has been devoted to the construction of surrogate models in derivative-free optimization, the question of how these models should be exploited once available has received comparatively little attention In DMS, this choice directly determines trial points to be evaluated by the algorithm and may therefore have a substantial influence on its numerical performance.

In this work, different approaches for jointly minimizing the surrogate models built during the search step of BoostDMS are proposed, thereby defining alternative search strategies. In particular, the ϵ-constraint scalarization, the Normal Boundary Intersection (NBI) framework, and a strategy based on the recently introduced Improved Front Steepest Descent (IFSD) algorithm [24] are detailed and numerically assessed as alternatives for model minimization within the search step of DMS.

The paper is organized as follows. Sections 2 and 3 review Direct Multisearch and model computation, respectively. Alternative approaches for exploiting the quadratic polynomial models are detailed in Section 4. Section 5 reports the numerical experiments supporting the proposed search-step alternatives. Finally, Section 6 concludes the work with final remarks and directions for future research.

2 Direct Multisearch (DMS)

Direct Multisearch (DMS), introduced in [15], extends directional direct search to multiobjective optimization. It directly exploits the concept of Pareto dominance to generate approximations to the complete Pareto front throughout the optimization process.

DMS is an algorithmic class encompassing several variants, depending, for example, on the chosen globalization strategy or the definition of the search step. Since its original introduction, several improved versions have been proposed, namely MultiGLODS [14], a global variant of DMS, BoostDMS [8], where a search step based on quadratic polynomial models was introduced, its parallel version Parallel BoostDMS [32], and DMS-FILTER-IR [31], developed for constrained optimization through a filter approach combined with an inexact restoration step.

The DMS framework has been successfully applied in several contexts, including financial portfolio optimization [9, 10] and the optimization of three-dimensional road alignments [22], and has proven competitive even when compared with derivative-based multiobjective optimization algorithms [2].

This section presents a simplified algorithmic description of BoostDMS and summarizes

its main theoretical properties, inherited from DMS. The following definition formalizes the concept of Pareto dominance through the strict partial order induced by the cone $\mathbb{R}^q_+ = \{z \in \mathbb{R}^q \mid z \geq 0\}$.

Definition 2.1 (Pareto dominance)**.** *Let $x, x' \in \Omega \subseteq \mathbb{R}^n$. We say that x dominates x' if*

$$F(x) \prec_F F(x'), \text{ i.e., if } F(x') - F(x) \in \mathbb{R}^q_+ \setminus \{\mathbf{0}\}.$$

Pareto dominance is used to characterize both global and local optimality. The efficient set is the subset of points in Ω that are not dominated by any other point in Ω. The image of the efficient set through the objective function defines the Pareto front, which represents the solution of the multiobjective optimization problem. A point $\overline{x} \in \Omega$ is said to be a local Pareto minimizer of Problem (1) if there exists a neighborhood $\mathcal{N}$ of $\overline{x}$ in which $\overline{x}$ is nondominated, that is, if there is no $x \in \mathcal{N} \cap \Omega$ such that $F(x) \prec_F F(\overline{x})$.

BoostDMS, similarly to DMS, can adopt two distinct globalization strategies: one based on a sufficient decrease condition and another based on the use of integer lattices. For simplicity, and since it corresponds to the available numerical implementation, only the latter will be described here, meaning that all points generated by the algorithm belong to an implicit mesh. For a more general description, we refer the reader to the original work [15].

Throughout the iterations, BoostDMS maintains and updates a list of feasible nondominated points and corresponding stepsize parameters, defining the current approximation to the Pareto front. Constraints are handled through an extreme barrier approach, meaning that only feasible points are evaluated in the objective function, while infeasible points are assigned the value $(+\infty, \ldots, +\infty) \in \mathbb{R}^q$. An iteration is declared successful whenever at least one new feasible nondominated point is found. Such points are added to the list, which is then updated by removing all dominated points.

Each iteration of BoostDMS is divided into four steps. First, a point and corresponding stepsize parameter are selected from the list of feasible nondominated points as the current iterate. Any selection criterion may be adopted, since it does not affect convergence. In the original implementation of DMS, as well as in BoostDMS, the selection relies on a spread metric, designed to reduce gaps between consecutive points in the list.

An optional Search Step follows, constituting the distinguishing feature of BoostDMS. Previously evaluated points are reused to construct local quadratic polynomial models for each objective function component. These points are selected within a ball centered at the current iterate, with radius proportional to the current stepsize parameter. The resulting surrogate models are then minimized, generating new trial points for evaluation in the original objective function.

Depending on the number of available sample points, the quadratic models may be underdetermined (handled through minimum Frobenius-norm interpolation), determined (corresponding to polynomial interpolation), or overdetermined (constructed through regression techniques). During the initial iterations, the search step is typically skipped until a predefined sample threshold is reached.

The models are minimized through a level-based procedure, individually or in combination, inside a trust-region corresponding to the ball used for sample selection. At the first level, each individual quadratic model is minimized separately. If among the resulting points there

isn't a feasible nondominated point of the original problem, the algorithm proceeds to the second level, where pairs of surrogate models are combined through a min–max scalarization and minimized again within the trust-region.The procedure terminates as soon as at least one feasible nondominated point is found or all levels have been exhausted. If the search step produces at least one new feasible nondominated point, the iteration is declared successful and the poll step is skipped. Otherwise, the poll step becomes mandatory.

In the Poll Step, a local search is performed around the current iterate by evaluating the objective function at points generated along a set of directions scaled by the current stepsize parameter. The requirements imposed on the set of directions are directly related to convergence conditions. In general, the directions should locally conform to the geometry of the feasible region and be asymptotically dense in the unit sphere, being typically positive-spanning sets. For a detailed discussion, we refer the reader to the original work [15]. If a new feasible nondominated point is found during the poll step, the iteration is declared successful. Otherwise, the list remains unchanged and the iteration is deemed unsuccessful.

At the end of each iteration, the stepsize parameter is updated for the current iterate (if it remains in the list) and for all newly generated points. If the iteration is successful, the stepsize parameter is maintained or increased. Otherwise, it is decreased.

The overall structure of this simplified description of BoostDMS is outlined in Algorithm 1.

Being an algorithmic variant of DMS, BoostDMS inherits all associated convergence results. The convergence analysis begins by establishing the existence of a subsequence of stepsizes converging to zero. Convergence is then studied along particular sequences of unsuccessful iterates, referred to as refining subsequences.

Definition 2.2. *A subsequence $\{x_k\}_{k\in K}$ of iterates generated by Algorithm 1, corresponding to unsuccessful poll steps, is said to be a refining subsequence if $\{\alpha_k\}_{k\in K}$ converges to zero. A limit point $\bar{x} \in \mathbb{R}^n$ of a refining subsequence is called a refined point.*

The behavior of the algorithm is subsequently analyzed through limit directions associated with refined points, called refining directions.

Definition 2.3. *Let $\bar{x}$ be the refined point of a convergent refining subsequence $\{x_k\}_{k\in K}$. If $\lim_{k\in K'} \frac{d_k}{\|d_k\|}$ exists, where $K' \subseteq K$, $d_k \in D_k$, and $x_k + \alpha_k d_k \in \Omega$, for sufficiently large $k \in K'$, then this limit is said to be a refining direction for $\bar{x}$.*

Assuming that the set of refining directions is asymptotically dense in the Clarke tangent cone to Ω at refined points, convergence to Pareto-Clarke critical points or Pareto-Clarke-KKT critical points can be established.

Theorem 2.4. *[15, Thm. 4.9] Consider a refining subsequence $\{x_k\}_{k\in K}$ converging to $\bar{x} \in \Omega$. Assume that F is Lipschitz continuous near $\bar{x}$ and $\mathrm{int}(T_\Omega^{Cl}(\bar{x})) \neq \emptyset$. If the set of refining directions for $\bar{x}$ is dense in $T_\Omega^{Cl}(\bar{x})$, then $\bar{x}$ is a Pareto-Clarke critical point, i.e,*

$$\forall d \in T_\Omega^{Cl}(\bar{x}), \exists \ell = \ell(d) \in \{1, \ldots, q\} \text{ such that } f_\ell^\circ(\bar{x}; d) \geq 0.$$

If, in addition, F is strictly differentiable at $\bar{x}$, then $\bar{x}$ is a Pareto-Clarke-KKT critical point, i.e.,

$$\forall d \in T_\Omega^{Cl}(\bar{x}), \exists \ell = \ell(d) \in \{1, \ldots, q\} \text{ such that } \nabla f_\ell(\bar{x})^\top d \geq 0.$$

Algorithm 1 A simplified description of BoostDMS

Initialization

Choose $x_0 \in \Omega$ such that $f_i(x_0) < +\infty$, $\forall i \in \{1, \ldots, q\}$, $\alpha_0 > 0$ an initial stepsize parameter, $0 < \beta_1 \leq \beta_2 < 1$ the coefficients for stepsize contraction and $\gamma \geq 1$ the coefficient for stepsize expansion. Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets, with directions d satisfying $0 < d_{\min} \leq \|d\| \leq d_{\max}$. Consider $L_0 = \{(x_0; \alpha_0)\}$ the initial list of nondominated points and corresponding stepsize parameters.

For $k = 0, 1, 2, \ldots$

1. **Selection of an iterate point:** Order the list L_k according to some criteria and select $(x_k; \alpha_k) \in L_k$ as the current iterate and stepsize parameter.

2. **Search step:** By reusing previously evaluated points, build a quadratic polynomial model for each component of the objective function, f_i, $i \in \{1, \ldots, q\}$. Minimize combinations of these models, computing a finite set of points $P_s = \{z_s \mid s \in S\}$. Evaluate F at $L_{\texttt{add}} = P_s \cap \Omega$. Let $L_{\texttt{trial}}$ be the set obtained from $L_k \cup \{(x; \alpha_k) : x \in L_{\texttt{add}}\}$ by removing dominated points. If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the search step) successful, set $L_{k+1} = L_{\texttt{trial}}$, and go to Step 4.

3. **Poll step:** Choose a positive spanning set D_k from the set $\mathcal{D}$ and consider the set $P_k = \{x_k + \alpha_k d \mid d \in D_k\}$. Evaluate F at $L_{\texttt{add}} = P_k \cap \Omega$. Let $L_{\texttt{trial}}$ be the set obtained from $L_k \cup \{(x; \alpha_k) : x \in L_{\texttt{add}}\}$ by removing dominated points. If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the poll step) as successful and set $L_{k+1} = L_{\texttt{trial}}$. Otherwise, declare the iteration as unsuccessful and set $L_{k+1} = L_k$.

4. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k,\texttt{new}} \in [\alpha_k, \gamma\alpha_k]$. Replace all the new points $(x_k + \alpha_k d; \alpha_k)$ in L_{k+1} by $(x_k + \alpha_k d; \alpha_{k,\texttt{new}})$, when success is coming from the poll step, or $(z_s; \alpha_k)$ in L_{k+1} by $(z_s; \alpha_{k,\texttt{new}})$, when success is coming from the search step. Replace also $(x_k; \alpha_k)$, if in L_{k+1}, by $(x_k; \alpha_{k,\texttt{new}})$. Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k,\texttt{new}} \in [\beta_1\alpha_k, \beta_2\alpha_k]$, and replace the poll pair $(x_k; \alpha_k)$ in L_{k+1} by $(x_k; \alpha_{k,\texttt{new}})$.

EndFor

3 Quadratic polynomial models

Since the work of Powell [28, 27], quadratic polynomial interpolation models have become an important tool in derivative-free optimization, acting as local surrogates of the objective function. Whether used within a trust-region framework or in the definition of a search step for single-objective derivative-free optimization, they naturally emerged in BoostDMS as models for the different components of the objective function. In this section, we summarize the main ideas and properties associated with these models, adapting the presentation in [8] to our notation and framework.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth function for which derivatives are unavailable. Then f can be approximated in a neighborhood of a current iterate x_k by a polynomial model $m : \mathbb{R}^n \to \mathbb{R}$ of the form

$$m(y) = \xi^\top \phi(y),$$

where $\phi(y)$ is a polynomial basis of dimension $s+1$ and ξ is a vector of coefficients. Given a set of previously evaluated sample points $Y = \{y_0, y_1, \ldots, y_p\}$ and corresponding function values $f(Y) = (f(y_0), \ldots, f(y_p))^\top$, with $y_0 = x_k$, the model coefficients are obtained by solving the linear system $M(\phi, Y)\,\xi = f(Y)$, where

$$M(\phi, Y) = \begin{bmatrix} \phi_0(y_0) & \phi_1(y_0) & \cdots & \phi_s(y_0) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(y_p) & \phi_1(y_p) & \cdots & \phi_s(y_p) \end{bmatrix} \quad f(Y) = \begin{bmatrix} f(y_0) \\ \\ \vdots \\ f(y_p) \end{bmatrix}$$

This system is square when the number of interpolation points equals the dimension of the chosen polynomial basis. In the remaining cases, namely when the system is underdetermined or overdetermined, the model coefficients can be computed through a minimum-norm solution or least-squares regression [8], respectively.

For simplicity, let us consider the natural basis of monomials. Under standard smoothness assumptions, namely the Lipschitz continuity of the gradient, even underdetermined models can recover the accuracy of first-order Taylor expansions, provided that the sample set satisfies a suitable geometric property, known as Λ-poisedness [13].

Theorem 3.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and assume that its gradient is Lipschitz continuous with constant $C_{\nabla f} > 0$ on the ball*

$$B(x_k; \Delta_k) = \{\, x \in \mathbb{R}^n \mid \|x - x_k\| < \Delta_k \,\}.$$

Let $Y = \{y_0, \ldots, y_p\}$ be a sample set with $n + 1 \le p < \frac{(n+1)(n+2)}{2} - 1$, which is Λ_L-poised for linear interpolation (or linear regression) at x_k, and let m be a quadratic model built from Y. Then there exists a constant $C_p > 0$ (depending only on p) such that, for all $y \in B(x_k; \Delta_k)$,

$$\begin{aligned} \|\nabla f(y) - \nabla m(y)\| &\le C_p\, \Lambda_L \left(C_{\nabla f} + \|H\|\right) \Delta_k, \\ |\, f(y) - m(y)\,| &\le \left(C_p\, \Lambda_L + \tfrac{1}{2}\right) \left(C_{\nabla f} + \|H\|\right) \Delta_k^2, \end{aligned}$$

with H denoting the Hessian of the model.

These relations show that the interpolation error decreases proportionally to the trust-region radius, provided that the norm of the model Hessian remains controlled, motivating the Minimum Frobenius Norm (MFN) approach for computing minimum-norm solutions.

Formally, let g and H denote the gradient and Hessian of the model, respectively. The MFN model is computed as the solution of

$$\begin{aligned} \min \quad & \tfrac{1}{4}\|H\|_F^2 \\ \text{s.t.} \quad & f(y_i) = f(y_0) + g^\top (y_i - y_0) + \tfrac{1}{2}(y_i - y_0)^\top H (y_i - y_0) \quad i = 1, \ldots, p, \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm [8]. This construction balances the number of required sample points and the ability of the model to incorporate curvature information.

When the number of sample points is sufficient to build a complete quadratic model, Taylor-like error bounds can also be established for the Hessian approximation. Again, an appropriate geometric condition must be satisfied by the sample set [13].

Theorem 3.2. *Let f be a twice continuously differentiable function with a Lipschitz continuous Hessian (with Lipschitz constant $C_{\nabla^2 f} > 0$) in the ball $B(x_k; \Delta_k)$. If the sample set $Y = \{y_0, \ldots, y_p\}$, with $p + 1 \geq \frac{(n+1)(n+2)}{2}$, is Λ_Q-poised for quadratic interpolation (or quadratic regression), then for all $y \in B(x_k; \Delta_k)$*

$$\begin{aligned} \|\nabla^2 f(y) - \nabla^2 m(y)\| &\leq C_p^1 \Lambda_Q C_{\nabla^2 f} \Delta_k, \\ \|\nabla f(y) - \nabla m(y)\| &\leq C_p^2 \Lambda_Q C_{\nabla^2 f} \Delta_k^2, \\ |f(y) - m(y)| &\leq \left(C_p^3 \Lambda_Q C_{\nabla^2 f} + \frac{C_{\nabla^2 f}}{6} \right) \Delta_k^3, \end{aligned}$$

where C_p^i, $i \in \{1, 2, 3\}$, are positive constants depending on p.

In [16], the use of quadratic polynomial models, built from previously evaluated points, was proposed for the definition of a search step in single-objective direct search. BoostDMS extends this strategy to direct multisearch, for multiobjective optimization. Previously evaluated points are again selected within a ball centered at the current iterate and used to construct a model for each objective function component. The type of model considered (underdetermined, determined, or overdetermined) depends on the number of available points. In all cases, the radius of the ball used for sample selection is directly related to the stepsize parameter, ensuring the quality of the models associated with refining subsequences, since the corresponding sequence of stepsize parameters converges to zero. For more details, the original references could be consulted [13, 16, 8].

The remaining question is how to exploit these models in the definition of a search step for DMS. Solving the corresponding multiobjective minimization problem, namely

$$\min \; M_k(x) = \left(m_1^k(x), \ldots, m_q^k(x) \right)^\top \text{ s.t. } x \in \Omega, \|x - x_k\| \leq \Delta_k \tag{2}$$

requires a careful approach. Although this is a derivative-based problem, the solution strategy should be selected judiciously. Should an approximation to the Pareto front be sought? Should scalarization approaches be considered? If so, which ones?

In [8], a weighted Chebyshev norm scalarization is applied to the joint minimization of models for a subset of components of the objective function by solving the problems:

$$\begin{aligned} \min \quad & \zeta \\ \text{s.t.} \quad & m_i^k(x) \;\le\; \zeta, \quad i \in I \\ & \|x - x_k\| \;\le\; \Delta_k \\ & x \in \Omega, \end{aligned} \tag{3}$$

where $I \subseteq \{1, 2, \ldots, q\}$.

The remainder of the paper investigates how the choice of the model-minimization strategy influences the performance of BoostDMS, thereby assessing whether this procedure should itself be regarded as an algorithmic design component rather than as a secondary implementation detail.

4 Alternative approaches for using quadratic models

In this section, three different strategies for exploiting the quadratic polynomial models within the search step of BoostDMS are proposed. Specifically, we consider the Normal Boundary Intersection (NBI) method [17], an adapted ϵ-constraint approach, and the Improved Front Steepest Descent (IFSD) algorithm [24]. These three approaches were selected because they represent fundamentally different philosophies for solving multiobjective optimization problems. The adapted ϵ-constraint method follows a scalarization approach, NBI relies on a geometric construction designed to improve the distribution of solutions, whereas IFSD directly exploits multiobjective descent directions. This diversity allows us to investigate whether the manner in which the surrogate models are minimized has a systematic influence on the performance of BoostDMS.

4.1 The Normal Boundary Intersection (NBI) method

The Normal Boundary Intersection (NBI) method, originally introduced in [17], is a scalarization approach designed to generate a well-distributed approximation of the Pareto front in multiobjective optimization. NBI constructs a family of subproblems that project candidate solutions along directions normal to the so-called *convex hull of individual minima* (CHIM) in the objective space. This geometric construction mitigates the uneven spacing of points typically associated with classical scalarization approaches and is also capable of revealing nonconvex regions of the Pareto front.

Let $M(x) = (m_1(x), m_2(x), \ldots, m_q(x))^\top$ denote the vector of objective functions, and let $x_i^* = \arg\min_{x \in \Omega} m_i(x)$, for $i = 1, \ldots, q$, be the individual global minimizers of each objective. The corresponding vector $M^* = (m_1(x_1^*), \ldots, m_q(x_q^*))^\top$, often referred to as the *ideal point*, serves as basis from which the payoff matrix Φ is defined as

$$\Phi = \left[M(x_1^*) - M^*,\; M(x_2^*) - M^*,\; \ldots,\; M(x_q^*) - M^* \right].$$

The CHIM is then described by the set $\{ M^* + \Phi\beta \mid \beta \in \Delta \}$, where $\Delta = \{\beta \in \mathbb{R}_+^q \mid \sum_i \beta_i = 1\}$ is the unit simplex. For each chosen weight vector β, a unit normal vector n to the CHIM is

computed, and the classical NBI subproblem is given by

$$\begin{aligned} \max_{x\in\Omega,\, t\in\mathbb{R}} \quad & t \\ \text{s.t.} \quad & M(x) - M^* = \Phi\beta + tn. \end{aligned} \tag{4}$$

The optimal solution of (4) corresponds to the intersection between the feasible image of $M(x)$ and the ray departing from the CHIM in direction n, yielding a Pareto-optimal solution for the chosen β.

The formulation above may generate spurious Pareto-front approximations, particularly when the search region is nonconvex. Indeed, the method aims at generating boundary points, rather than necessarily Pareto-optimal ones. To address this issue, a modified version of the problem was proposed in [30]:

$$\begin{aligned} \max_{x\in\Omega,\, t\in\mathbb{R}} \quad & t \\ \text{s.t.} \quad & M(x) - M^* \leq \Phi\beta + t\, n. \end{aligned} \tag{5}$$

Although relaxed, this formulation preserves the geometric essence of NBI by driving the search along the normal direction, while improving robustness under model uncertainty.

In BoostDMS, the NBI mechanism is embedded within the model-based search step that precedes the poll step. Each quadratic model is minimized individually, over the feasible region, subject to the trust-region constraint, to estimate the local minima x_i^*. The ideal point M^* and model-based payoff matrix Φ are then computed from these approximate minimizers. The CHIM and its associated normal direction n are subsequently constructed from these model-based quantities.

To generate additional trial points, $2^q - 1$ equidistant weight vectors on the simplex are considered, matching the maximum number of trial points that can be generated during the search step in the current implementation of BoostDMS (corresponding to the different models combinations inside each level). The $2^q - 1$ modified NBI subproblems (5) are solved for the selected weight vectors respecting the original constraints defining Ω and within the trust-region. This ensures that the resulting trial points remain in a region where the surrogate models provide reliable approximations.

4.2 Adapted $\epsilon-$constraint method

The ϵ-constraint method is a classical scalarization technique for multiobjective optimization. It transforms a multiobjective problem into a sequence of constrained single-objective problems, where one objective is minimized while the remaining objectives are restricted by upper bounds, denoted by ϵ_j. By systematically varying these bounds, the method can generate representative Pareto-optimal solutions, even when the Pareto front is nonconvex.

Given $M(x) = (m_1(x), \ldots, m_q(x))^\top$ and the feasible region Ω, the standard ϵ-constraint problem associated with the i-th objective is formulated as

$$\begin{aligned} \min_{x\in\Omega} \quad & m_i(x) \\ \text{s.t.} \quad & m_j(x) \leq \epsilon_j, \quad j = 1, \ldots, q,\ j \neq i. \end{aligned} \tag{6}$$

Each choice of the bounds ϵ_j defines a distinct subproblem, whose minimizer corresponds to a different trade-off point on the Pareto front. This approach is particularly effective when one objective is prioritized while the remaining ones act as constraints controlling the allowed degradation.

In BoostDMS the classical formulation (6) is adapted to operate on local quadratic surrogate models, within the trust-region framework.

Let x_k denote the current iterate, and let $m_i^k(x)$ represent the quadratic polynomial model associated with the objective function $f_i(x)$, constructed from sample points selected in the ball centered at x_k with radius Δ_k. The ϵ-bounds are dynamically fixed at the current objective values, namely $\epsilon_j = f_j(x_k)$ for $j \neq i$, so that the search focuses on improving one objective without significantly deteriorating the remaining ones. The resulting model-based subproblem is given by

$$\begin{aligned} \min_{x\in\Omega} \quad & m_i^k(x) \\ \text{s.t.} \quad & m_j^k(x) \leq \epsilon_j = f_j(x_k), \quad j \neq i, \\ & \|x_k - x\| \leq \Delta_k. \end{aligned} \tag{7}$$

This formulation ensures that the generated trial point remains within a region where the surrogate models are expected to provide accurate approximations of the objective function components. By solving (7) for each objective $i \in \{1, \ldots, q\}$, a set of q candidate points is obtained, which are then evaluated using the true objective function F in an attempt to identify new feasible nondominated points.

4.3 Improved front steepest descent (IFSD)

The Improved Front Steepest Descent (IFSD) method [24] builds upon the classical steepest descent algorithm for multiobjective optimization [20]. However, in the spirit of DMS, IFSD explicitly maintains a list of nondominated points that is iteratively refined to approximate the Pareto front. Each iteration combines *refinement steps*, which drive the points toward Pareto stationarity, with *exploration steps*, which enrich the front by identifying new trade-offs among the objectives.

Given the smooth multiobjective problem

$$\min_{x\in\mathbb{R}^n} M(x) = (m_1(x), \ldots, m_q(x))^\top, \tag{8}$$

the steepest-descent direction is defined by

$$v(x) = \arg\min_{d\in\mathbb{R}^n} \max_{j=1,\ldots,q} \left\{ \nabla m_j(x)^\top d + \tfrac{1}{2}\|d\|^2 \right\}.$$

For any subset of objectives $I \subset \{1, \ldots, q\}$, the corresponding *partial descent direction* is defined as

$$v_I(x) = \arg\min_{d\in\mathbb{R}^n} \max_{j\in I} \left\{ \nabla m_j(x)^\top d + \tfrac{1}{2}\|d\|^2 \right\}.$$

The optimal values of these subproblems, denoted by $\theta(x)$ and $\theta_I(x)$, respectively, measure the local potential for common or partial descent. Pareto-stationary points satisfy $\theta(x) = 0$.

At each iteration of the IFSD algorithm [24], every nondominated point x_c in the current list undergoes a two-stage procedure consisting of a *refinement step* followed by an *exploration step*.

In the refinement step, a common steepest descent direction $v(x_c)$ is computed, and an Armijo-type line search is performed to obtain

$$z_c = x_c + \alpha_c v(x_c),$$

ensuring sufficient decrease in all objectives, whenever $\theta(x_c) < 0$. This step guarantees convergence toward Pareto stationarity.

The exploration step aims at densifying the approximation to the Pareto front. Starting from z_c, for every subset $I \subset \{1, \ldots, q\}$ such that $\theta_I(z_c) < 0$, a partial descent direction $v_I(z_c)$ is computed. Line searches along these directions generate candidates $z_c + \alpha_I v_I(z_c)$, improving subsets of objectives. Newly generated points are added to the list, while dominated solutions are removed. This mechanism ensures that all points may contribute to the exploration process.

Under standard smoothness and compactness assumptions, the resulting sequences of sets converge to sets of Pareto-stationary points, and every accumulation point of the iterates is Pareto stationary for Problem (8).

IFSD was originally designed for unconstrained multiobjective optimization. However, the surrogate models computed during the search step of BoostDMS are only considered reliable within the trust-region $\|x - x_k\| \leq \Delta_k$. Therefore, the computation of the descent direction and the partial descent directions took into consideration the trust-region and the original problem constraints. A different possibility, also tested but with worst numerical performance, was to solve the initial unconstrained problems and from the final approximation to the Pareto front of Problem (8), only keep the points feasible with respect to both $\boldsymbol{\Omega}$ and the trust-region constraint.

Even so, this set is typically large. Considering the expensive nature of the true objective function, only a small subset of these points can be selected for evaluation. The model values are assumed to provide accurate approximations of the objective function components, and only points that remain nondominated with respect to the current list L_k (using model values as surrogates for the true objective function values) are retained. Let $L_k^{\texttt{IFSD}}$ denote this reduced list of points.

If the list $L_k^{\texttt{IFSD}}$ contains at most $2^q - 1$ points, then all of them are selected for evaluation, that is, $P_s = L_k^{\texttt{IFSD}}$. Otherwise, a maximum of $2^q - 1$ points must be selected from $L_k^{\texttt{IFSD}}$, prioritizing the most isolated ones in an attempt to densify the current approximation to the Pareto front.

To identify these isolated candidates, the crowding distance [18] is employed. For each candidate point, the crowding distance is computed with respect to an auxiliary list $L_{\texttt{aux}}$, initially defined as L_k, still using the model values as surrogates for the objective function values at the new points. Points associated with the least crowded regions of the objective space are therefore preferred. In the case of ties, the selected point is the one closest to the centroid of the corresponding crowding box (see Figure 1). Once selected, the point is added to P_s and included in $L_{\texttt{aux}}$, again using the model values as surrogates for the objective function

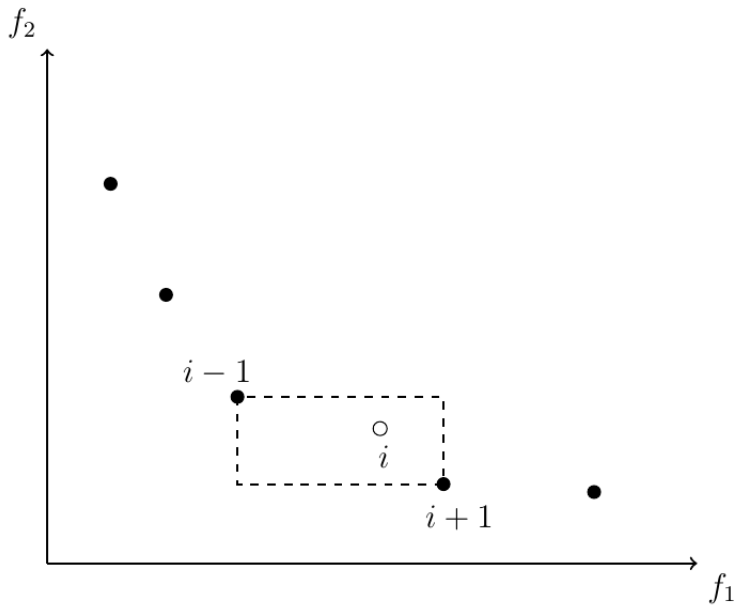


Figure 1: Example of the computation of crowding distance and centroid.

values. The selection strategy is repeatedly applied until a maximum of $2^q - 1$ points have been selected for evaluation in the true objective function.

Figure 1 illustrates the concept of crowding distance in the biobjective case. The black dots represent the points in $L_{\texttt{aux}}$ in the objective space (f_1, f_2), while the white dot corresponds to the model solution indexed by i in the final set computed by IFSD for model minimization. The neighboring points, denoted by $i-1$ and $i+1$, define the bounds of a rectangular region associated with point i, represented by the dashed box. This region is constructed using the objective values of adjacent points in each objective direction. The crowding distance quantifies the extent of this region through normalized differences between the objective values of neighboring points. It therefore provides an estimate of the local density of solutions around point i, where larger values indicate that the solution lies in a less crowded region of the objective space, thereby promoting diversity among the selected candidates. The centroid of this box is defined as the midpoint of the rectangle determined by the neighboring points $i-1$ and $i+1$.

5 Numerical experiments

In this section, our goal is to evaluate the numerical performance of the proposed search strategies. Comparisons were made between the original BoostDMS and the three proposed schemes. All tests were conducted on a server equipped with 512 GB of RAM and two Intel® Xeon® Gold 6534 processors, each featuring 8 cores and 16 threads, with a base clock speed of 3.9 GHz, and a maximum turbo frequency of 4.2 GHz. The algorithms were implemented and executed using MATLAB R2024b.

5.1 Performance assessment

To compare the proposed strategies, we employ performance profiles, introduced by Dolan and Moré [19], which provide a comprehensive framework for benchmarking optimization algorithms across a collection of test problems. A performance profile represents the cumulative distribution of a given performance metric, allowing to simultaneously assess efficiency and robustness. For a solver $s \in S$ and a set of problems P, the profile is defined by

$$\rho_s(\tau) = \frac{1}{|P|} \left| \left\{ p \in P \mid t_{p,s} \leq \tau \min_{s \in S} t_{p,s} \right\} \right|,$$

where $\tau \geq 1$ and $t_{p,s}$ denotes the value of a selected performance metric obtained by solver s on problem p. Thus, $\rho_s(\tau)$ corresponds to the fraction of problems for which solver s performs within a factor τ of the best solver. In particular, $\rho_s(1)$ measures the proportion of problems for which solver s is the most efficient, whereas larger values of $\rho_s(\tau)$ for increasing τ indicate greater robustness.

The numerical comparison was carried out using four widely adopted quality indicators: purity, hypervolume, and the spread measures Γ and Δ. The purity metric quantifies the proportion of nondominated points generated by a solver that belong to the reference Pareto front. Let $F_{p,s}$ denote the approximation to the Pareto front produced by solver s for problem p, and let F_p be a reference Pareto front obtained by merging the approximations generated by all solvers for the same problem and removing dominated points [15]. The purity value is then defined as

$$\bar{t}_{p,s} = \mathrm{Pur}_{p,s} = \frac{|F_{p,s} \cap F_p|}{|F_{p,s}|}.$$

To assess both convergence and diversity, we also considered the hypervolume metric [33], which measures the volume of the objective space dominated by the approximation $F_{p,s}$ with respect to a reference point $U_p \in \mathbb{R}^q$, dominated by all the points generated by the different solvers for the same problem:

$$\bar{t}_{p,s} = \mathrm{HV}_{p,s} = \mathrm{Vol}\left(\bigcup_{x \in F_{p,s}} [x, U_p] \right),$$

where $\mathrm{Vol}(\cdot)$ denotes the q-dimensional Lebesgue measure and $[x, U_p]$ is the hyper-rectangle defined by the lower corner x and the upper corner U_p.

Since larger values of purity and hypervolume indicate better performance, the corresponding performance profiles were constructed using their reciprocals, that is, $t_{p,s} = \frac{1}{\bar{t}_{p,s}}$, so that lower values consistently represent superior performance across all metrics.

To complement the assessment of the Pareto front distribution, we further considered the spread metrics Γ and Δ [18]. The Γ metric measures the largest gap between consecutive nondominated points, whereas the Δ metric evaluates the uniformity of their distribution.

Suppose that solver s computes for problem p an approximation consisting of the points $x^1, \ldots, x^N$. Let x^0 and x^{N+1} denote the extreme points, corresponding to the points with the

best and worst values for each component of the objective function [15]. Then, the Γ metric is given by

$$\Gamma_{p,s} = \max_{j\in\{1,\ldots,q\}} \left(\max_{i\in 0,\ldots,N} \delta_{j,i} \right), \tag{9}$$

where $\delta_{j,i} = f_j(x^{i+1}) - f_j(x^i)$, assuming that the objective values are sorted in ascending order for each objective component.

The Δ metric is defined as

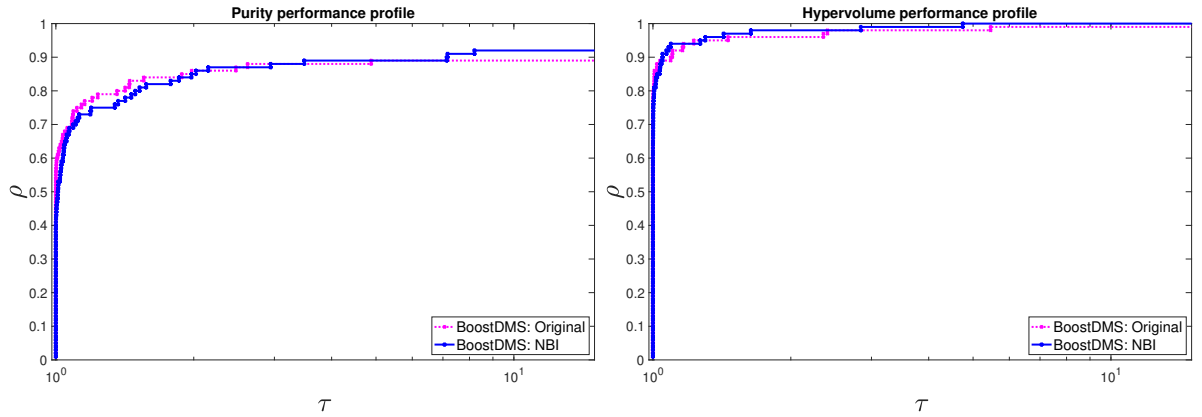
$$\Delta_{p,s} = \max_{j\in\{1,\ldots,q\}} \left(\frac{\delta_{j,0} + \delta_{j,N} + \sum_{i=1}^{N-1} |\delta_{j,i} - \bar{\delta}_j|}{\delta_{j,0} + \delta_{j,N} + (N-1)\bar{\delta}_j} \right), \tag{10}$$

where $\bar{\delta}_j$ denotes the average of the distances $\delta_{j,i}$, $i = 1, \ldots, N-1$. Lower values of both Γ and Δ indicate better performance.

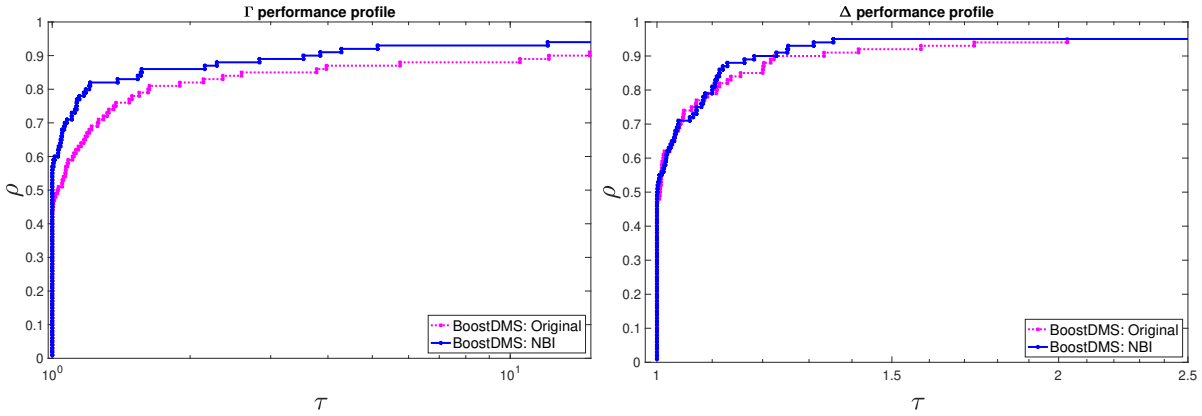
The assessment was conducted on a comprehensive benchmark consisting of 100 bound-constrained test problems proposed in [15], covering a broad range of objective landscapes and difficulty levels. Regarding the parameters of BoostDMS, all default values were considered. For all variants, we set $\alpha_0 = 1$, $\beta_1 = \beta_2 = 0.5$, $\gamma = 1$, and allowed a maximum of 5000 function evaluations, jointly with a minimum step size of 10^{-3}, for all the points in the list.

5.2 The NBI method

The NBI method is first compared with the original strategy implemented in BoostDMS, based on the min-max scalarization. Figure 2 presents the performance profiles obtained for the four metrics considered.



(a) Purity (b) Hypervolume



(c) Spread Gamma (d) Spread Delta

Figure 2: Comparing BoostDMS(Original) with BoostDMS(NBI) based on performance profiles, for a maximum of 5000 function evaluations.

The profiles show that the NBI method behaves similarly to the original strategy in terms of purity and hypervolume. In both cases, the corresponding curves remain close, and no clear dominance of one strategy over the other can be observed. This indicates that incorporating NBI into the search step does not lead to a systematic improvement in the indicators more directly related to the quality of the final approximation, namely, the proportion of points belonging to the reference nondominated front and the dominated volume covered in the objective space.

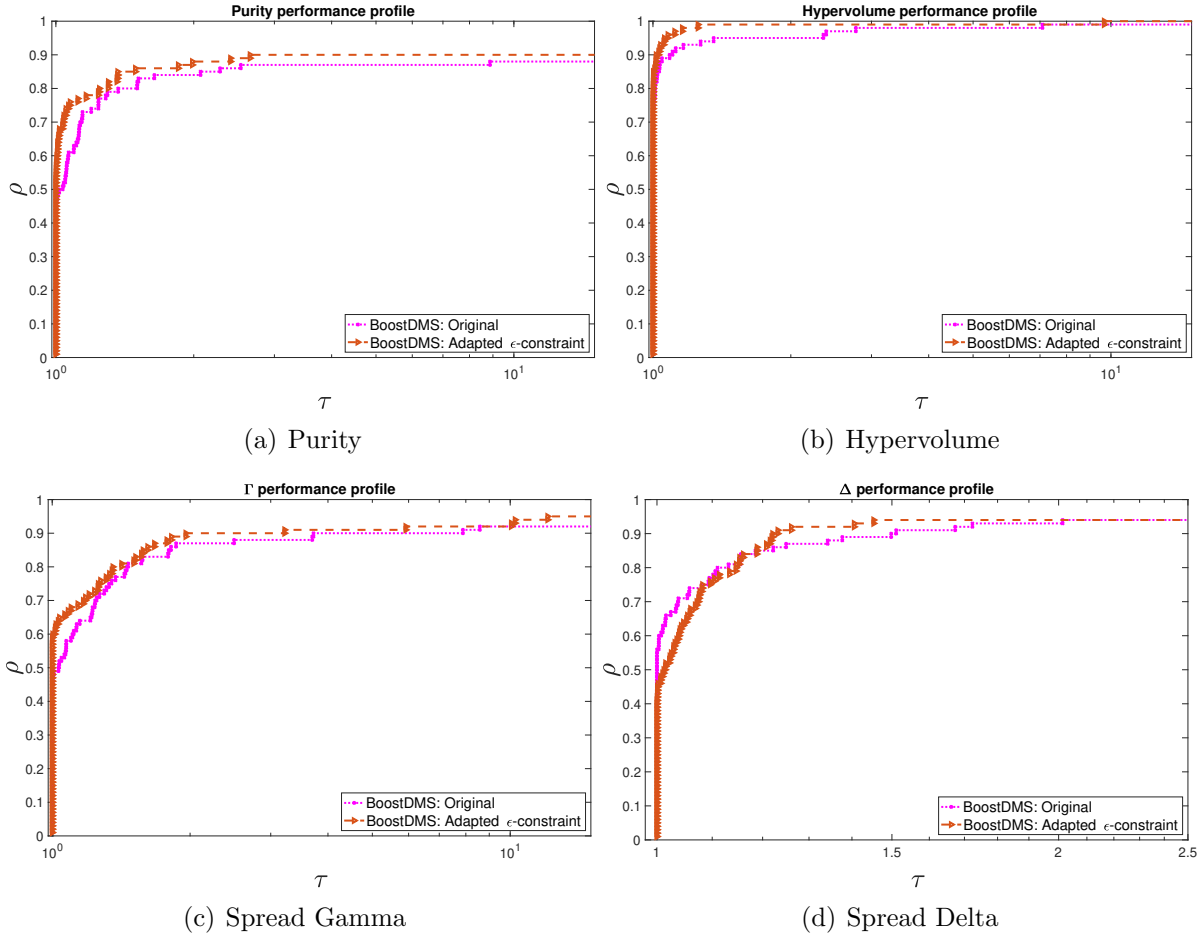
A different behavior is observed for the spread metrics. For both Γ and Δ spreads, the curve associated with the NBI method lies above the one corresponding to the original strategy. This indicates that the use of NBI tends to generate better-distributed approximations of the Pareto front. In particular, the improvement observed in the Γ-spread profile suggests a reduction in the largest gaps of the computed front, while the Δ-spread profile indicates a more uniform distribution of nondominated points. This is consisting with the original aims of NBI.

Overall, the main advantage of the NBI method appears to be associated with the distribution of the final approximation rather than with a clear improvement in purity or hypervolume.

Within the local model-based setting considered here, this advantage is primarily reflected in the spread indicators, whereas the quality-related metrics remain essentially comparable to those obtained with the original strategy.

5.3 Adapted $\epsilon-$constraint method

The ϵ-constraint method is now compared with the original strategy implemented in BoostDMS. Figure 3 presents the performance profiles obtained for the four metrics under analysis: purity, hypervolume, Γ spread, and Δ spread.



(a) Purity

(b) Hypervolume

(c) Spread Gamma

(d) Spread Delta

Figure 3: Comparing BoostDMS(Original) with BoostDMS(Adapted $\epsilon-$constraint) based on performance profiles, for a maximum of 5000 function evaluations.

The most relevant differences are observed in the purity and hypervolume profiles. Regarding purity, the adapted ϵ-constraint method outperforms the original strategy on a significant portion of the test set. This suggests that the final approximations generated by this method contain a larger proportion of points belonging to the reference nondominated front.

The hypervolume profile exhibits a similar trend. The curve associated with the adapted ϵ-constraint method is generally above the one corresponding to the original strategy, indicating

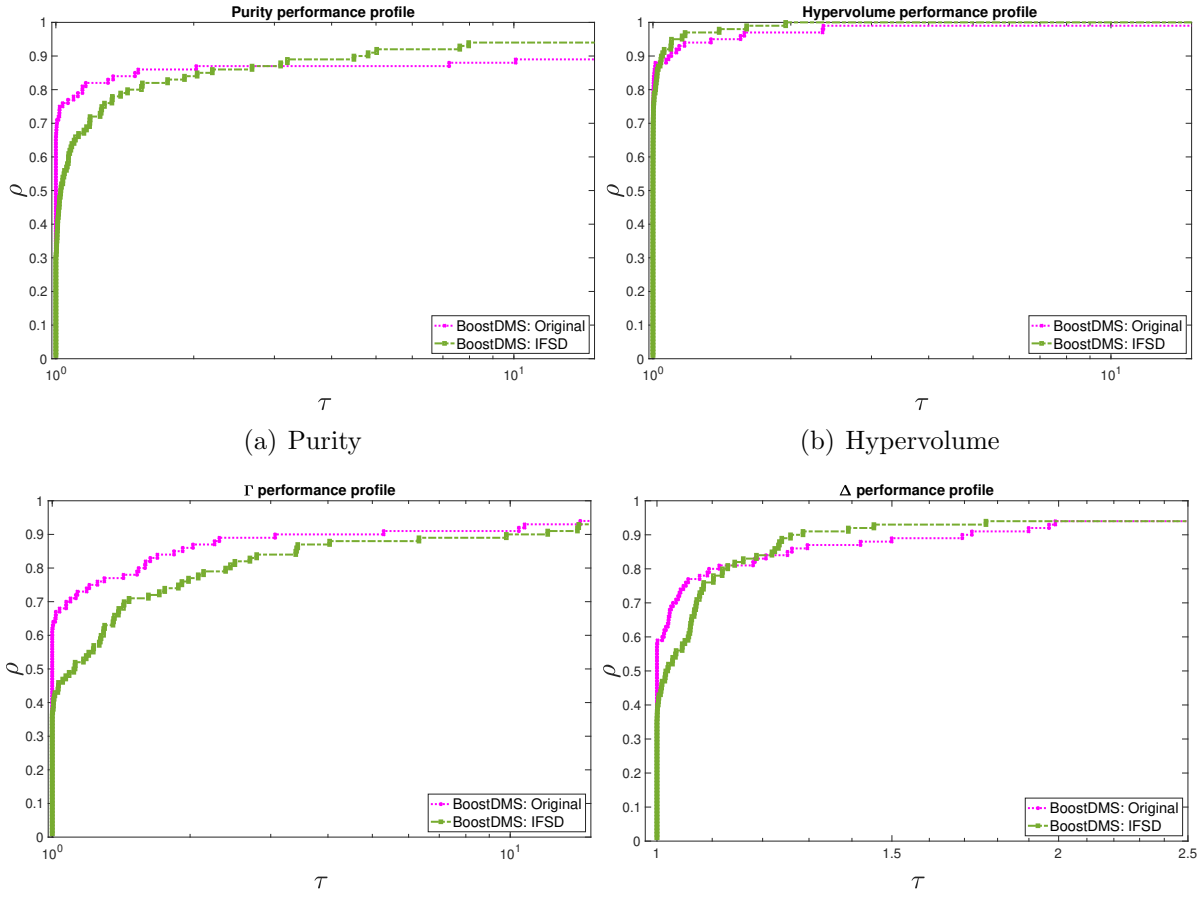
that the computed approximations tend to dominate a larger region of the objective space. The improvement is noticeable and relatively consistent throughout the profile.

The comparison is more balanced for the spread indicators. For Γ spread, the adapted ϵ-constraint method presents a slight advantage, suggesting a reduction in the largest gaps within the computed approximations. For Δ spread, the two strategies exhibit similar behavior, with exception of $\tau = 1$, where there is an advantage of the original strategy. Even so, the use of the adapted ϵ-constraint method does not substantially affect the uniformity of the final approximations, remaining competitive with the original strategy in this regard.

Overall, the adapted ϵ-constraint method primarily improves the quality of the final approximations, as indicated by the superior purity and hypervolume profiles.

5.4 The IFSD method

The final comparison is presented in Figure 4, where the results obtained by BoostDMS(Original) and BoostDMS(IFSD) are reported. The hypervolume performance profile shows that the two



(a) Purity

(b) Hypervolume

(c) Spread Gamma

(d) Spread Delta

Figure 4: Comparing BoostDMS(Original) with BoostDMS(IFSD) based on performance profiles, for a maximum of 5000 function evaluations.

variants behave very similarly, with a slight advantage of the IFSD strategy. This indicates that, in terms of the overall quality of the approximations (combining convergence and diversity), the IFSD variant performs comparably to the original strategy. Moreover, both variants exhibit high efficiency and robustness according to this metric across the considered test set.

For purity, a different behavior is observed. BoostDMS(Original) is the best performer for small τ values, indicating greater efficiency. However, as τ increases, the curve associated with BoostDMS(IFSD) eventually matches or surpasses that of BoostDMS(Original), reaching an equal or higher ρ value for large τ values. This behavior suggests that BoostDMS(IFSD) provides greater robustness in terms of the purity metric.

Regarding the spread indicators Δ and Γ, BoostDMS(Original) is the most efficient variant, attaining the minimum value for both metrics in 60% of the problems. BoostDMS(IFSD) is able to achieve a similar performance to the one of BoostDMS(Original), but requiring larger values of τ.

6 Conclusions and future work

In this work, alternative strategies for exploiting quadratic polynomial models within the search step of Direct Multisearch were proposed and numerically assessed. The considered approaches were based on the Normal Boundary Intersection method, an adapted ϵ-constraint scalarization, and the Improved Front Steepest Descent algorithm.

The numerical results indicate that the different strategies influence the quality of the final approximations in distinct ways. The NBI-based search step produced results comparable to those of the original BoostDMS strategy in terms of purity and hypervolume, while providing some advantages in the spread indicators. In contrast, the adapted ϵ-constraint strategy exhibited the most consistent improvement among the tested alternatives. In particular, it achieved better performance profiles for purity and hypervolume, while remaining competitive with respect to the spread indicators.

The IFSD-based strategy exhibited a different behavior. Although it was designed to enrich the Pareto front approximation through refinement and exploration mechanisms, the numerical results did not reveal a clear advantage over the original strategy in terms of the quality-oriented indicators.

Overall, the results demonstrate that the choice of the model-minimization strategy within the search step of DMS can significantly influence the numerical performance of the algorithm. Among the approaches evaluated, the adapted ϵ-constraint method appears to be the most promising, particularly when the objective is to improve the quality of the final approximation under a fixed budget of function evaluations. These findings suggest that the model-minimization strategy should be regarded as an algorithmic design choice rather than a secondary implementation detail.

Several research directions may be explored to further enhance the numerical performance of BoostDMS. Promising extensions include the development of a search step based on the construction of quadratic models in low-dimensional subspaces at each iteration [11], as well as the incorporation of parallel computing techniques to improve the efficiency of the algorithm [32].

References

[1] H. Afshari, W. Hare, and S. Tesfamariam. "Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures". In: *Appl. Soft Comp.* 83 (2019), p. 105631.

[2] R. Andreani, A. L. Custódio, and M. Raydan. "Using first-order information in direct multisearch for multiobjective optimization". In: *Optim. Methods Softw.* 37 (2022), pp. 2121–2142.

[3] C. Audet and J. E. Dennis Jr. "Mesh adaptive direct search algorithms for constrained optimization". In: *SIAM J. Optim.* 17 (2006), pp. 188–217.

[4] C. Audet and W. Hare. *Derivative-free and Blackbox Optimization*. Cham, Switzerland: Springer, 2017.

[5] C. Audet, G. Savard, and W. Zghal. "A mesh adaptive direct search algorithm for multiobjective optimization". In: *European J. Oper. Res.* 204 (2010), pp. 545–556.

[6] C. Audet, G. Savard, and W. Zghal. "Multiobjective optimization through a series of single-objective formulations". In: *SIAM J. Optim.* 19 (2008), pp. 188–210.

[7] J. Bigeon, S. Le Digabel, and L. Salomon. "DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization". In: *Comput. Optim. Appl.* 79 (2021), pp. 301–338.

[8] C. P. Brás and A. L. Custódio. "On the use of polynomial models in multiobjective directional direct search". In: *Comput. Optim. Appl.* 77 (2020), pp. 897–918.

[9] R. P. Brito and P. Júdice. "Asset classification under the IFRS 9 framework for the construction of a banking investment portfolio". In: *Intl. Trans. in Op. Res.* 29 (2022), pp. 2613–2648.

[10] R. P. Brito, H. Sebastião, and P. Godinho. "Efficient skewness/semivariance portfolios". In: *J. Asset Manag.* 17 (2016), pp. 331–346.

[11] C. Cartis and L. Roberts. "Scalable subspace methods for derivative-free nonlinear least-squares optimization". In: *Math. Program.* 199 (2023), pp. 461–524.

[12] G. Cocchi, G. Liuzzi, A. Papini, and M. Sciandrone. "An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints". In: *Comput. Optim. Appl.* 69 (2018), pp. 267–296.

[13] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. Philadelphia, USA: SIAM, 2009.

[14] A. L. Custódio and J. F. A. Madeira. "MultiGLODS: Global and local multiobjective optimization using direct search". In: *J. Global Optim.* 72 (2018), pp. 323–345.

[15] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. "Direct multisearch for multiobjective optimization". In: *SIAM J. Optim.* 21 (2011), pp. 1109–1140.

[16] A. L. Custódio, H. Rocha, and L. N. Vicente. "Incorporating minimum Frobenius norm models in direct search". In: *Comput. Optim. Appl.* 46 (2010), pp. 265–278.

[17] I. Das and J. E. Dennis Jr. "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems". In: *SIAM J. Optim.* 8 (1998), pp. 631–657.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Trans. on Evolut. Comput.* 6 (2002), pp. 182–197.

[19] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: *Math. Program.* 91 (2002), pp. 201–213.

[20] J. Fliege and B. F. Svaiter. "Steepest descent methods for multicriteria optimization". In: *Math. Meth. Oper. Res.* 51 (2000), pp. 479–494.

[21] J. Handl, D. B. Kell, and J. Knowles. "Multiobjective optimization in bioinformatics and computational biology". In: *IEEE/ACM Trans. Comput. Biol. and Bioinform.* 4 (2007), pp. 279–292.

[22] D. Hirpa, W. Hare, Y. Lucet, Y. Pushak, and S. Tesfamariam. "A bi-objective optimization framework for three-dimensional road alignment design". In: *Transp. Res. Part C Emerg. Technol.* 65 (2016), pp. 61–78.

[23] Y. Jin and B. Sendhoff. "Pareto-based multiobjective machine learning: An overview and case studies". In: *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.* 38 (2008), pp. 397–415.

[24] M. Lapucci and P. Mansueto. "Improved front steepest descent for multi-objective optimization". In: *Oper. Res. Lett.* 51 (2023), pp. 242–247.

[25] G. Liuzzi, S. Lucidi, and F. Rinaldi. "A derivative-free approach to constrained multiobjective nonsmooth optimization". In: *SIAM J. Optim.* 26 (2016), pp. 2744–2774.

[26] P. S. Potrebko, J. Fiege, M. Biagioli, and J. Poleszczuk. "Investigating multi-objective fluence and beam orientation IMRT optimization". In: *Phys. Med. Biol.* 62 (2017), pp. 5228–5244.

[27] M. J. D. Powell. "A new algorithm for unconstrained optimization". In: *Nonlinear Programming*. Ed. by J. B. Rosen, O. L. Mangasarian, and K. Ritter. New York: Academic Press, 1970, pp. 31–65.

[28] M. J. D. Powell. "An efficient method for finding the minimum of a function of several variables without calculating derivatives". In: 7 (1964), pp. 155–162.

[29] F. Rosso, V. Ciancio, J. Dell'Olmo, and F. Salata. "Multi-objective optimization of building retrofit in the Mediterranean climate by means of genetic algorithm application". In: *Energy Build.* 216 (2020), p. 109945.

[30] P. K. Shukla. "On the Normal Boundary Intersection Method for generation of efficient front". In: *Lect. Notes Comp. Sci.* 4487 (2007), pp. 310–317.

[31] E. J. Silva and A. L. Custódio. "An inexact restoration direct multisearch filter approach to multiobjective constrained derivative-free optimization". In: *Optim. Methods Softw.* 40 (2025), pp. 406–432.

[32] S. Tavares, C. P. Brás, A. L. Custódio, V. Duarte, and P. Medeiros. "Parallel strategies for Direct Multisearch". In: *Numer. Algorithms* 92 (2023), pp. 1757–1788.

[33] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca. "Performance assessment of multiobjective optimizers: An analysis and review". In: *IEEE Trans. Evol. Comput.* 7 (2003), pp. 117–132.