An active-set method for box-constrained multiobjective optimization

N. Fazzio[*] L. F. Prudente[†] M. L. Schuverdt[‡]

December 28, 2025

Abstract

We propose an active-set algorithm for smooth multiobjective optimization problems subject to box constraints. The method works on one face of the feasible set at a time, treating it as a lower-dimensional region on which the problem simplifies. At each iteration, the algorithm decides whether to remain on the current face or to move to a different one, characterizing two types of iterations: face-exploring and face-abandoning steps. Backtracking and extrapolation strategies are combined, allowing the working set to be expanded or reduced by multiple constraints in a single iteration. Global convergence to Pareto critical points is established, and under a dual-nondegeneracy assumption we prove finite identification of the active set. Implementation aspects are discussed in detail, and numerical experiments on benchmark problems illustrate the practical performance of the method in comparison with existing approaches.

Key words: Multiobjective optimization, box-constrained problems, active-set method, global convergence.

1 Introduction

Many real-world problems involve the simultaneous optimization of multiple, often conflicting, objectives; see, for example, [34]. In this paper, we consider the box-constrained multiobjective optimization problem

$$\min_x F(x) \quad \text{s. t.} \quad x \in \mathcal{B}, \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is given componentwise by $F(x) \coloneqq (f_1(x), \ldots, f_m(x))$, and each $f_j : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. The feasible region is the box

$$\mathcal{B} \coloneqq \{x \in \mathbb{R}^n : \ell \leq x \leq u\},$$

[*]Department of Mathematics, FCE, University of La Plata, La Plata, Bs. As., Argentina. Email: nfazzio@mate.unlp.edu.ar

[†]Instituto de Matemática e Estatística, Universidade Federal de Goiás, CEP 74001-970 - Goiânia, GO, Brazil. Email: lfprudente@ufg.br

[‡]CONICET, Department of Mathematics, FCE, University of La Plata, La Plata, Bs. As., Argentina. Email: schuverd@mate.unlp.edu.ar

with $\ell, u \in \mathbb{R}^n$, $\ell < u$, and possibly $\ell_i = -\infty$ or $u_i = \infty$ for some components. Besides their practical relevance, box-constrained problems also play an important role as subproblems in augmented Lagrangian and penalty methods for more general constrained optimization; see, e.g., [3]. In the multiobjective setting, such inner subproblems arise naturally in recent extensions of augmented Lagrangian schemes [5]. Therefore, the development of numerical algorithms capable of efficiently solving (1) is of significant importance for both theoretical advances and practical applications.

Several general methodologies have been developed for solving constrained multiobjective optimization problems. Scalarization techniques constitute the classical approach, reducing the vector problem to a scalar one; see [24, 28]. Despite their versatility, scalarization methods may fail to recover nonconvex regions of the Pareto front. Heuristic and evolutionary schemes form another important class of approaches [10], although such methods generally lack theoretical convergence guarantees. A third family comprises descent-type schemes, which mimic classical iterative scalar optimization algorithms by constructing common descent directions; a representative example is the projected-gradient method of [18]. In addition, several algorithms have been developed specifically for box-constrained multiobjective optimization. Hybrid schemes combining evolutionary components with descent-type steps have been proposed in [21, 33]. Interior-point–type strategies were investigated in [25, 32], while a derivative-free approach was considered in [8].

Active-set strategies play a central role in scalar box-constrained optimization. Their development dates back to the seminal work of Polyak [29] in the 1960s and the later contribution of Dembo and Tulowitzki [11], with numerous refinements introduced over subsequent decades. The essential idea is to exploit the geometry of the feasible region by treating each face of the box as a lower-dimensional region on which the problem simplifies. An active-set algorithm operates on one face at a time, taking *face-exploring steps* that seek progress within the current face and *face-abandoning steps* that leave it whenever further progress becomes unproductive. Consequently, robust criteria are required to determine when to continue exploring a face and when abandoning it becomes advantageous. Modern active-set schemes implement this idea by combining projected-gradient-type steps used to abandon faces with efficient first or second-order procedures to explore a face. Two of the most successful methods following this strategy are the algorithm of Birgin and Martínez [2], available in the Gencan software, and the method of Hager and Zhang [19], which forms the basis of the ASA_CG solver. These solvers are widely regarded as state-of-the-art for bound-constrained scalar optimization, owing to their efficiency, robustness, and strong theoretical guarantees.

Despite the success of active-set techniques in the scalar case, no analogous framework has been developed for the box-constrained multiobjective problem (1). Given the effectiveness of active-set ideas in exploiting the structure of bound constraints, extending these techniques to the multiobjective setting is both natural and desirable. This gap motivates the active-set algorithm proposed in this work.

The method developed here can be viewed as a multiobjective counterpart of the active-set framework of Birgin and Martínez [2]. The decision of whether to continue exploring the current face or to switch to a different one is guided by comparing two optimality measures: the optimality measure of the original problem and that of the problem restricted to the current face. These quantities are computed through their projected-gradient subproblems. The proposed algorithm integrates a projected-gradient-type

mechanism for abandoning faces with a general face-exploring search direction. Backtracking and extrapolation strategies are incorporated into the iteration, allowing the working set to be expanded or reduced by multiple constraints in a single iteration. We establish global convergence to Pareto critical points and prove finite active-set identification under a dual-nondegeneracy assumption. After the correct face is identified, the algorithm behaves like a reduced-space multiobjective descent method.

A detailed discussion of implementation aspects is also provided. In particular, face-abandoning iterations may use a *Barzilai–Borwein spectral* projected-gradient direction. Face-exploring iterations employ a *truncated Newton–Gradient* direction built from a scalarized Hessian based on the Lagrange multipliers of the steepest-descent subproblem. An Armijo backtracking line search based on quadratic interpolation is used to select the step size, ensuring practical efficiency. Numerical experiments on a broad collection of benchmark problems illustrate the practical behavior of the proposed method. Comparisons with classical projected gradient schemes and the widely used evolutionary algorithm NSGA-II [10] assess both computational efficiency and the quality of the approximated Pareto fronts.

The remainder of the paper is organized as follows. Section 2 recalls fundamental concepts of Pareto optimality, projected-gradient subproblems, and their dual characterizations. Section 3 presents the proposed active-set method, hereafter referred to as the *Multiobjective Active-set Method for Box-constrained Optimization (MAMBO)*, and discusses its main algorithmic components. Section 4 establishes the global convergence analysis, while Section 5 introduces the notion of dual-nondegenerate Pareto critical points and proves finite identification of the active set. Section 6 discusses implementation aspects. Section 7 reports numerical experiments on standard multiobjective benchmark problems. Concluding remarks are presented in Section 8.

Notation. We denote by $\mathbb{N}$ the set of nonnegative integers numbers $\{0, 1, 2, \ldots\}$, and by $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ the set of real numbers, the set of nonnegative real numbers, and the set of positive real numbers, respectively. As usual, $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the set of n-dimensional real column vectors and the set of $n \times m$ real matrices, respectively. If $u, v \in \mathbb{R}^n$, then $u \leq v$ (resp. $u < v$) is to be understood in a componentwise sense, i.e., $u_i \leq v_i$ (resp. $u_i < v_i$) for all $i \in \{1, \ldots, n\}$. $\|\cdot\|$ is the Euclidean norm. Given $S \subset \mathbb{R}^n$ a set of points in $\mathbb{R}^n$, $\operatorname{conv}(S)$ denotes the convex hull of S. We denote by Δ_m the m-dimensional simplex, i.e., $\Delta_m \coloneqq \{\lambda \in \mathbb{R}^m \colon \lambda_j \geq 0, \sum_j \lambda_j = 1\}$. If $\mathbb{K} = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$, with $k_j < k_{j+1}$ for all $j \in \mathbb{N}$, then we denote $\mathbb{K} \underset{\infty}{\subset} \mathbb{N}$.

2 Preliminaries

In this section, we recall the main concepts and tools used throughout the paper. We consider the general multiobjective optimization problem

$$\min_x F(x) \quad \text{s.t.} \quad x \in \Omega, \tag{2}$$

where $\Omega \subset \mathbb{R}^n$ is a nonempty closed convex set and $F = (f_1, \ldots, f_m) \colon \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable.

A point $x^* \in \Omega$ is said to be *Pareto optimal* (or *efficient*) for (2) if there exists no $x \in \Omega$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$. It is called *weakly Pareto optimal* (or *weakly efficient*) if there is no $x \in \Omega$ such that $F(x) < F(x^*)$. Local versions

of both concepts are defined in the usual way by restricting the feasible region to a neighborhood of x^*. A necessary condition for weak Pareto optimality of x^* is

$$JF(x^*)\,(\Omega - x^*) \cap (-\mathbb{R}^m_{++}) = \emptyset,$$

where $JF(x^*) \in \mathbb{R}^{m\times n}$ is the Jacobian of F at x^* and $\Omega - x^* := \{u - x^* \colon u \in \Omega\}$. A point satisfying this condition is called a *Pareto critical* or *stationary* point of (2). If x is nonstationary, then there exists $d \in \Omega - x$ such that $\nabla f_j(x)^\top d < 0$ for all $j \in \{1,\dots,m\}$. Every such d is a *descent direction* for F at x, i.e., there exists $\varepsilon > 0$ such that $F(x + \alpha d) < F(x)$ for all $\alpha \in (0,\varepsilon)$; see [18, Proposition 1].

Let us define $\mathcal{D}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$\mathcal{D}(x,d) := \max_{j\in\{1,\dots,m\}} \nabla f_j(x)^\top d. \tag{3}$$

Note that $x^* \in \Omega$ is Pareto critical of (2) if and only if $\mathcal{D}(x^*,d) \geq 0$ for all $d \in \Omega - x^*$. We recall the following basic properties.

Lemma 2.1. *([17, Lemma 2.2]) For all $x \in \Omega$ and $d \in \mathbb{R}^n$:*

(a) $\mathcal{D}(x,\alpha d) = \alpha\mathcal{D}(x,d)$ for all $\alpha > 0$;

(b) $(x,d) \mapsto \mathcal{D}(x,d)$ is continuous.

We now introduce the *projected gradient direction*, following [18]. Given $x \in \Omega$, consider the constrained scalar-valued minimization problem:

$$\min_d \mathcal{D}(x,d) + \tfrac{1}{2}\|d\|^2 \quad \text{s.t.} \quad d \in \Omega - x. \tag{4}$$

Since the objective function is strongly convex and Ω is a closed and convex set, problem (4) is well-defined and admits a unique optimal solution. Let us denote by $(v(x),\theta(x)) \in \mathbb{R}^n \times \mathbb{R}$ the solution and optimal value of (4), i.e.,

$$v(x) := \operatorname*{argmin}_{d\in\Omega - x} \ \mathcal{D}(x,d) + \tfrac{1}{2}\|d\|^2. \tag{5}$$

and

$$\theta(x) := \mathcal{D}(x,v(x)) + \tfrac{1}{2}\|v(x)\|^2. \tag{6}$$

The following proposition establishes continuity properties of $v(\cdot)$ and $\theta(\cdot)$ and characterizes stationary points of (2) in terms of these functions.

Proposition 2.2. *Let $v(\cdot)$ and $\theta(\cdot)$ be as in (5)–(6). Then:*

(a) $\theta(x) \leq 0$ for all $x \in \Omega$;

(b) $v(\cdot)$ and $\theta(\cdot)$ are continuous on Ω;

(c) The following statements are equivalent: (i) $x \in \Omega$ is nonstationary; (ii) $\theta(x) < 0$; (iii) $v(x) \neq 0$.

In particular, x is stationary for (2) if and only if $\theta(x) = 0$ (equivalently, $v(x) = 0$).

Proof. See [18, Propositions 3 and 4], [16, Proposition 3.4] and [13, Theorem 1]. □

Note that, whenever x is nonstationary, $\theta(x) < 0$ and hence $\mathcal{D}(x, v(x)) < 0$, so $v(x)$ is a descent direction for F at x. Introducing an auxiliary variable $\tau \in \mathbb{R}$, problem (4) can be written in the differentiable form

$$\min_{\tau,d} \ \tau + \tfrac{1}{2}\|d\|^2 \quad \text{s.t.} \quad \nabla f_j(x)^\top d \le \tau \ (\forall j \in \{1,\dots,m\}), \quad d \in \Omega - x. \tag{7}$$

The next result provides a dual representation for the projected gradient direction. For this, define

$$J(x,d) \coloneqq \operatorname*{argmax}_{j\in\{1,\dots,m\}} \ \nabla f_j(x)^\top d. \tag{8}$$

the set of indices attaining the maximum in the definition of $\mathcal{D}$.

Proposition 2.3. *Let $x \in \Omega$. Then v is the unique solution of (4), i.e., $v = v(x)$, if and only if there exists $w \in \operatorname{conv}\{\nabla f_j(x)\}_{j\in J(x,v)}$ such that*

$$v = P_\Omega(x - w) - x.$$

Proof. See, for example, [13]. □

We finish this section by noting that, the vector w in Proposition 2.3 admits a natural interpretation in terms of the differentiable formulation (7). Indeed, for such w there exists $\lambda \in \Delta_m$ with $\lambda_j = 0$ whenever $j \notin J(x,v)$ such that $w = \sum_{j=1}^m \lambda_j \nabla f_j(x)$. Moreover, the coefficients λ_j coincide with the KKT multipliers associated with the constraints $\nabla f_j(x)^\top d \le \tau$ in the differentiable problem (7); see [13, Remark 1].

3 The active-set algorithm

The core idea of an active-set strategy for solving (1) is to exploit the *faces* of the feasible region, treating them as a lower-dimensional regions on which the problem can be simplified. The feasible set $\mathcal{B}$ is partitioned into disjoint faces, each corresponding to a specific subset of active bound constraints. Formally, for $A \subset \{1,\dots,2n\}$, the corresponding face is

$$\mathcal{F}_A \coloneqq \left\{ \begin{array}{lll} z \in \mathcal{B}\colon & z_i = \ell_i, & \text{if } \ i \in A \\ & z_i = u_i, & \text{if } \ n+i \in A \\ & \ell_i < z_i < u_i, & \text{if } \ i \notin A \ \text{ and } \ n+i \notin A \end{array} \right\}.$$

Given $x \in \mathcal{B}$, the set of *active constraints* is

$$A(x) \coloneqq \{i\colon x_i = \ell_i\} \cup \{n+i\colon x_i = u_i\} \tag{9}$$

so that $x \in \mathcal{F}_{A(x)}$. We also define

$$I(x) \coloneqq \{i \in \{1,\dots,n\}\colon i \notin A(x) \text{ and } (n+i) \notin A(x)\}$$

to be the *free coordinates* of x. Let $\overline{\mathcal{F}}_{A(x)}$ denote the closure of $\mathcal{F}_{A(x)}$ and $\operatorname{aff}(\mathcal{F}_{A(x)})$ its affine hull. The linear subspace parallel to that hull is

$$\mathcal{S}_{A(x)} \coloneqq \operatorname{aff}(\mathcal{F}_{A(x)}) - x,$$

which consists precisely of directions that keep all active coordinates fixed.

The active-set algorithm works on one face at a time, exploring it until the progress within that face becomes insufficient. If x^k is already a stationary point of (1), the algorithm terminates successfully. Otherwise, it must decide whether it is worthwhile to remain on the current face $\mathcal{F}_{A(x^k)}$ or to abandon it. To make this decision, we consider the original problem (1) defined over the entire box $\mathcal{B}$, and its restriction to the current face, given by

$$\min_x F(x) \quad \text{s. t.} \quad x \in \overline{\mathcal{F}}_{A(x^k)}. \tag{10}$$

The variables in (10) are precisely the coordinates x_i for $i \in I(x^k)$, corresponding to the *free* components of x within the current face. Both problems admit projected-gradient models of the form (4), which yield stationarity measures and descent directions. Specifically, let

$$(v_{\mathcal{B}}(x^k), \theta_{\mathcal{B}}(x^k)) \quad \text{and} \quad (v_{\overline{\mathcal{F}}}(x^k), \theta_{\overline{\mathcal{F}}}(x^k))$$

denote the solutions and optimal values of (4) when $\Omega = \mathcal{B}$ and $\Omega = \overline{\mathcal{F}}_{A(x^k)}$, respectively. By Proposition 2.2 (c), the quantities $\theta_{\mathcal{B}}(x^k)$ and $\theta_{\overline{\mathcal{F}}}(x^k)$ provide consistent measures of stationarity for problems (1) and (10).

If $\theta_{\overline{\mathcal{F}}}(x^k) = 0$, then x^k is Pareto critical for the problem restricted to the current face, and no further improvement can be achieved by exploring that face. More generally, following the scalar criteria of [2, 19], the face should also be abandoned whenever

$$|\theta_{\overline{\mathcal{F}}}(x^k)| \leq \nu \, |\theta_{\mathcal{B}}(x^k)|,$$

where $\nu \in (0, 1)$ is a fixed algorithmic parameter. In this case, a *face-abandoning iteration* is performed, taking the search direction as $v_{\mathcal{B}}(x^k)$ and determining the step size by an *Armijo-type backtracking* line search.

Otherwise, the method performs a *face-exploring iteration*. As a first step, the algorithm computes an auxiliary direction by solving the projected-gradient subproblem restricted to the affine subspace associated with the current face, namely

$$\min_x F(x) \quad \text{s. t.} \quad x \in \operatorname{aff}(\mathcal{F}_{A(x^k)}),$$

whose projected-gradient model takes the form

$$\min_d \mathcal{D}(x^k, d) + \tfrac{1}{2}\|d\|^2 \quad \text{s. t.} \quad d \in \mathcal{S}_{A(x^k)}.$$

Its solution provides a steepest-descent-type direction lying in the reduced space. A face-exploring search direction is then selected by enforcing an *angle-type condition* with the auxiliary vector, guaranteeing sufficient descent within the current face. Along this direction, the algorithm applies either a backtracking or an extrapolation strategy, depending on the observed descent behavior. The Armijo-type backtracking line search enforces feasibility with respect to the box $\mathcal{B}$, ensuring that the new iterate remains within the current face. The extrapolation step, on the other hand, aims at improving practical performance: whenever a satisfactory descent is detected, the step size is enlarged, while each trial point is projected onto the box $\mathcal{B}$ to maintain feasibility. This may result in faster progress along the current face and, in some cases, lead the iterate to the boundary of that face, where additional bound constraints become active and the algorithm moves to a lower-dimensional face. Through this mechanism, multiple

constraints may become active in a single iteration, thus accelerating progress while preserving feasibility.

In summary, there are two possible iteration types:

- *Face-abandoning iteration*: The next iterate is computed using the projected gradient direction $v_{\mathcal{B}}(x^k)$ and an Armijo-type backtracking line search, which moves the point away from the current face.
- *Face-exploring iteration*: The search direction is computed within the affine subspace of the current face and followed by either a backtracking or an extrapolation step, depending on the observed descent behavior. In this case, the backtracking step generates a new iterate that remains on the same face, preserving the active set, whereas the extrapolation step may produce a point in the boundary of the current face, activating additional constraints and moving to a lower-dimensional one.

In the following, we formally present the algorithm.

Algorithm 1
MAMBO: Multiobjective Active-set Method for Box-constrained Optimization

---

Let $x^0 \in \mathcal{B}$ and parameters $\nu \in (0,1)$, $\Gamma \in (0,1)$, $c_1 \in (0, \frac{1}{2})$, $c_2 \in (c_1, 1)$, $N > 1$, and $0 < \underline{\omega} < \bar{\omega}$ be given. Initialize $k \leftarrow 0$.

Step 1. *Projected gradient computation*
Compute $(v_{\mathcal{B}}(x^k), \theta_{\mathcal{B}}(x^k)) \in \mathbb{R}^n \times \mathbb{R}$ as

$$v_{\mathcal{B}}(x^k) \coloneqq \underset{d \in \mathcal{B} - x^k}{\operatorname{argmin}} \ \mathcal{D}(x^k, d) + \tfrac{1}{2}\|d\|^2 \tag{11}$$

and

$$\theta_{\mathcal{B}}(x^k) \coloneqq \mathcal{D}(x^k, v_{\mathcal{B}}(x^k)) + \tfrac{1}{2}\left\|v_{\mathcal{B}}(x^k)\right\|^2.$$

If $\theta_{\mathcal{B}}(x^k) = 0$, then STOP.

Step 2. *Iteration type selection*
Let $A(x^k)$ be the set of active constraints at x^k as in (9). Compute $(v_{\overline{\mathcal{F}}}(x^k), \theta_{\overline{\mathcal{F}}}(x^k)) \in \mathbb{R}^n \times \mathbb{R}$ as

$$v_{\overline{\mathcal{F}}}(x^k) \coloneqq \underset{d \in \overline{\mathcal{F}}_{A(x^k)} - x^k}{\operatorname{argmin}} \ \mathcal{D}(x^k, d) + \tfrac{1}{2}\|d\|^2 \tag{12}$$

and

$$\theta_{\overline{\mathcal{F}}}(x^k) \coloneqq \mathcal{D}(x^k, v_{\overline{\mathcal{F}}}(x^k)) + \tfrac{1}{2}\left\|v_{\overline{\mathcal{F}}}(x^k)\right\|^2.$$

If

$$|\theta_{\overline{\mathcal{F}}}(x^k)| \leq \nu |\theta_{\mathcal{B}}(x^k)|, \tag{13}$$

then proceed to Step 3. Otherwise, go to Step 4.

Step 3. *Face-abandoning iteration: leaving the current face*
Define

$$d^k := v_{\mathcal{B}}(x^k). \tag{14}$$

Set $\alpha_{\text{trial}} \leftarrow 1$, and go to Step 5.

Step 4. *Face-exploring iteration: producing a point in the closure of the current face*
Compute $v_{\mathcal{S}}(x^k) \in \mathbb{R}^n$ as

$$v_{\mathcal{S}}(x^k) \coloneqq \underset{d \in \mathcal{S}_{A(x^k)}}{\operatorname{argmin}} \ \mathcal{D}(x^k, d) + \tfrac{1}{2}\|d\|^2, \tag{15}$$

and a non-null direction $d^k \in \mathbb{R}^n$ such that

$$d_i^k = 0 \quad \text{if} \quad x_i^k = \ell_i \text{ or } x_i^k = u_i, \ \forall i \in \{1, \dots, n\}, \tag{16}$$

and

$$\mathcal{D}(x^k, d^k) \leq -\Gamma \|v_{\mathcal{S}}(x^k)\| \|d^k\|. \tag{17}$$

Define $\alpha_{\max} \coloneqq \max\{\alpha \geq 0 \colon x^k + \alpha d^k \in \mathcal{B}\}$ and set $\alpha_{\text{trial}} \leftarrow \min\{1, \alpha_{\max}\}$. If $\alpha_{\max} > 1$, then go to Step 4.1. Otherwise, go to Step 4.2.

Step 4.1. *(At this point $\alpha_{\text{trial}} = 1$ and $x^k + d^k \in \operatorname{Int}(\mathcal{B})$)*
If

$$f_j(x^k + d^k) \leq f_j(x^k) + c_1 \mathcal{D}(x^k, d^k), \quad \forall j \in \{1, \dots, m\}, \tag{18}$$

and

$$\mathcal{D}(x^k + d^k, d^k) \geq c_2 \mathcal{D}(x^k, d^k), \tag{19}$$

then set $\alpha_k := 1$, $x^{k+1} := x^k + d^k$, and go to Step 6. If (18) does not hold, go to Step 5. If (19) does not hold, go to Step 4.3.

Step 4.2. *(At this point $\alpha_{\text{trial}} = \alpha_{\max}$ and $x^k + d^k \notin \operatorname{Int}(\mathcal{B})$)*
If

$$f_j(x^k + \alpha_{\max} d^k) < f_j(x^k), \quad \forall j \in \{1, \dots, m\}, \tag{20}$$

then go to Step 4.3. Otherwise, go to Step 5.

Step 4.3. *Extrapolation procedure*
If

$$\alpha_{\text{trial}} < \alpha_{\max} \quad \text{and} \quad N\alpha_{\text{trial}} > \alpha_{\max},$$

then set $\alpha_{\text{new}} \leftarrow \alpha_{\max}$; otherwise, set $\alpha_{\text{new}} \leftarrow N\alpha_{\text{trial}}$. If

$$f_j(P_{\mathcal{B}}(x^k + \alpha_{\text{new}} d^k)) \geq f_j(P_{\mathcal{B}}(x^k + \alpha_{\text{trial}} d^k)), \tag{21}$$

for some $j \in \{1, \dots, m\}$, then set $\alpha_k := \alpha_{\text{trial}}$, $x^{k+1} := P_{\mathcal{B}}(x^k + \alpha_k d^k)$, and go to Step 6. Otherwise, set $\alpha_{\text{trial}} \leftarrow \alpha_{\text{new}}$ and repeat Step 4.3.

Step 5. *Armijo-type backtracking*
If

$$f_j(x^k + \alpha_{\text{trial}} d^k) \leq f_j(x^k) + c_1 \alpha_{\text{trial}} \mathcal{D}(x^k, d^k), \quad \forall j \in \{1, \dots, m\}, \tag{22}$$

then set $\alpha_k := \alpha_{\text{trial}}$, $x^{k+1} := x^k + \alpha_k d^k$, and go to Step 6. Otherwise, compute $\alpha_{\text{new}} \in [\underline{\omega}\alpha_{\text{trial}}, \bar{\omega}\alpha_{\text{trial}}]$, set $\alpha_{\text{trial}} \leftarrow \alpha_{\text{new}}$, and repeat Step 5.

Step 6. *Iteration update*
Set $k \leftarrow k + 1$ and go to Step 1.

The main steps of one iteration of the active-set algorithm are summarized in the flowchart shown in Figure 1. In the following, we discuss some theoretical and algorithmic aspects of the method.

1. *Stopping criterion.* At Step 1, the algorithm terminates when $\theta_{\mathcal{B}}(x^k) = 0$. By Proposition 2.2(c), this condition holds if and only if x^k is Pareto critical for problem (1), thus providing a stationarity test for the multiobjective case.

2. *Relation between $\theta_{\mathcal{B}}(x^k)$ and $\theta_{\overline{\mathcal{F}}}(x^k)$.* Since $(\overline{\mathcal{F}}_{A(x^k)} - x^k) \subset (\mathcal{B} - x^k)$, the feasible region of problem (12) is contained in that of (11). Consequently, the corresponding optimal values satisfy $\theta_{\mathcal{B}}(x^k) \leq \theta_{\overline{\mathcal{F}}}(x^k)$. Using Proposition 2.2 (a), we obtain $|\theta_{\overline{\mathcal{F}}}(x^k)| \leq |\theta_{\mathcal{B}}(x^k)|$, which implies that $|\theta_{\overline{\mathcal{F}}}(x^k)|/|\theta_{\mathcal{B}}(x^k)| \leq 1$. The parameter $\nu \in (0, 1)$ therefore acts as a threshold for this ratio, determining whether the algorithm proceeds with a face-abandoning or a face-exploring iteration.

3. *Relation with the scalar case.* In the single-objective setting ($m = 1$), the vectors $v_{\mathcal{B}}(x^k)$ and $v_{\overline{\mathcal{F}}}(x^k)$ reduce, respectively, to the usual *projected gradient* $g_P(x^k) := P_{\mathcal{B}}(x^k - \nabla F(x^k)) - x^k$ and the *internal gradient* $g_I(x^k) := P_{\mathcal{S}_{A(x^k)}}(g_P(x^k))$. That is, $g_I(x^k)$ coincides with $g_P(x^k)$ on the free variables while its remaining components are zero. Hence, the proposed framework generalizes the classical scalar active-set scheme, recovering its behavior when $m = 1$; see [2].

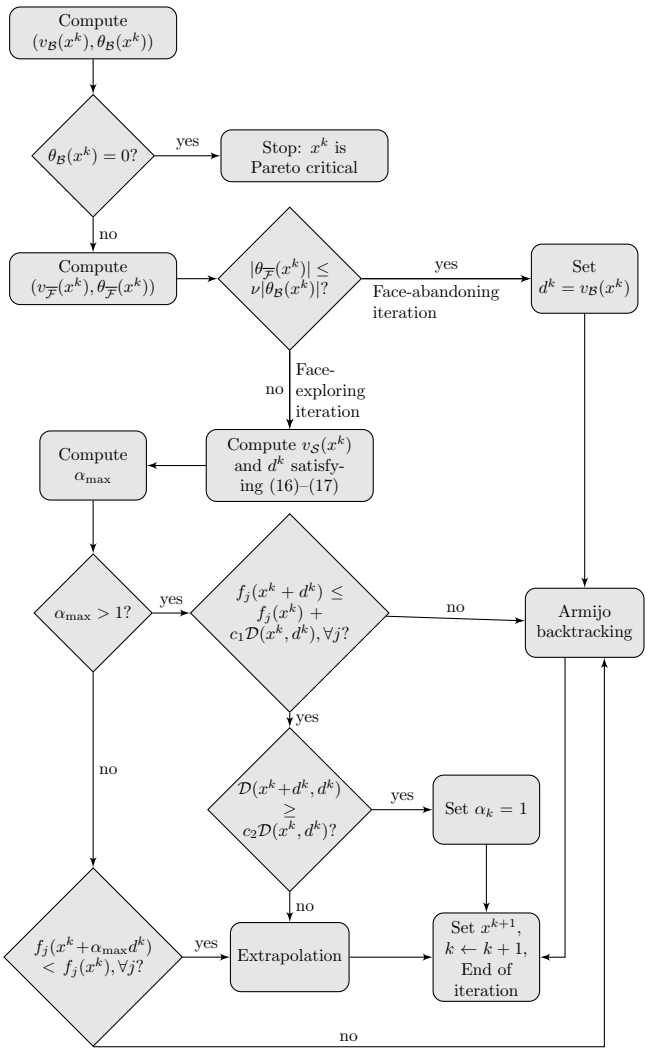

Figure 1: Flowchart of an iteration k of Algorithm 1.

4. *Leaving the current face.* When the search direction is computed at Step 3, the algorithm necessarily leaves the current face. Indeed, assume that condition (13) holds and, by contradiction, that $d^k = v_{\mathcal{B}}(x^k)$ belongs to $\overline{\mathcal{F}}_{A(x^k)} - x^k$. Since $v_{\mathcal{B}}(x^k)$ is feasible for (12) under the assumption, we have

$$\theta_{\overline{\mathcal{F}}}(x^k) \leq \mathcal{D}(x^k, v_{\mathcal{B}}(x^k)) + \tfrac{1}{2}\|v_{\mathcal{B}}(x^k)\|^2 = \theta_{\mathcal{B}}(x^k).$$

Then, by Proposition 2.2 *(a)*, $|\theta_{\mathcal{B}}(x^k)| \leq |\theta_{\overline{\mathcal{F}}}(x^k)|$, and combining this with (13) yields $|\theta_{\mathcal{B}}(x^k)| \leq \nu|\theta_{\mathcal{B}}(x^k)|$, implying $\nu \geq 1$, obtaining a contradiction. Therefore, $d^k \notin \overline{\mathcal{F}}_{A(x^k)} - x^k$, meaning that d^k has nonzero components corresponding to fixed variables and $x^{k+1} \notin \overline{\mathcal{F}}_{A(x^k)}$. In this case, $d^k = v_{\mathcal{B}}(x^k)$ is a descent direction for F at x^k, and the step size α_k is determined by the Armijo rule (22).

5. *Existence of a non-null direction.* In Step 4, a non-null direction d^k satisfying conditions (16)–(17) always exists. Indeed, note first that $v_{\mathcal{S}}(x^k) \neq 0$. If $v_{\mathcal{S}}(x^k) = 0$, then $\theta_{\overline{\mathcal{F}}}(x^k) = 0$, because the feasible region of problem (12) is contained in that of (15). In this case, the test (13) would be satisfied (since $\theta_{\mathcal{B}}(x^k) \neq 0$, i.e., x^k is not critical), and the algorithm would perform a face-abandoning iteration (Step 3) instead of Step 4. Condition (17) was introduced in [17] and generalizes the classical *angle condition* from scalar to multiobjective optimization, ensuring, in particular, that d^k is a descent direction at x^k. It is straightforward to verify that $v_{\mathcal{S}}(x^k)$ itself satisfies the required conditions (16)–(17).

6. *Staying within the closure of the current face.* When the search direction is computed at Step 4, the new iterate remains in the closure of the current face. Condition (16) enforces $d_i^k = 0$ for every index i such that $x_i^k = \ell_i$ or $x_i^k = u_i$, which implies that all variables corresponding to active bounds stay fixed during the update. Consequently, the next iterate satisfies $x_i^{k+1} = x_i^k$ for these indices, so $x^{k+1} \in \overline{\mathcal{F}}_{A(x^k)}$.

7. *Interior trial and Step 4.1.* At Step 4, the test on $\alpha_{\max}$ distinguishes whether the trial step $x^k + d^k$ remains strictly inside the box or reaches its boundary. When $\alpha_{\max} > 1$, the unit step is feasible, meaning that the full direction d^k can be taken without violating the bounds. In this case, the algorithm proceeds to Step 4.1, which attempts the unit step size $\alpha_k = 1$ and verifies the Armijo-type condition (18) together with the curvature condition (19). These two inequalities jointly constitute the *standard Wolfe conditions* for the unit step size in the multiobjective setting, as introduced in [22] and further discussed in [23].

   - If both (18) and (19) are satisfied, the unit step size is accepted. In this case, condition (18) guarantees sufficient decrease, and (19) indicates that no significant further improvement can be expected simultaneously in all objectives along d^k. The latter condition also ensures that the unit step size is not unnecessarily conservative.
   - If the Armijo condition (18) fails, the algorithm switches to the backtracking procedure (Step 5), initially trying the unit step size.
   - If only the curvature condition (19) fails, it reveals that all objectives can still be reduced simultaneously by moving further along d^k, which motivates the use of the extrapolation strategy (Step 4.3) to explore larger step sizes.

8. *Boundary trial and Step 4.2.* When $\alpha_{\max} \leq 1$, the trial point $x^k + d^k$ reaches the boundary of the box, so the current direction cannot be followed further without violating feasibility. In this case, Step 4.2 verifies condition (20), which tests whether all objective functions decrease at the boundary point $x^k + \alpha_{\max} d^k$. If this condition holds, it indicates that a simultaneous decrease in all objectives is still possible by moving further along the same direction, motivating the use of the *extrapolation* procedure (Step 4.3). In this situation, the next iterate remains in the closure of the current face, and the algorithm typically moves to a lower-dimensional one in the following iteration. Otherwise, if the boundary point does not yield descent, the algorithm proceeds to the *backtracking* procedure (Step 5), trying $\alpha_{\max}$ first.

9. *Extrapolation (Step 4.3).* The extrapolation procedure aims to explore larger step sizes along the current search direction, either from an interior point (coming from Step 4.1) or from a boundary point (coming from Step 4.2). It performs successive projections of $x^k + \alpha_{\text{trial}} d^k$ onto the box, with increasing values of α_{trial}. Each projected point is accepted as long as all objective functions continue to decrease, and the procedure terminates when no further improvement is observed. Note that if $x^k + \alpha_{\text{trial}} d^k$ is interior but $x^k + N\alpha_{\text{trial}} d^k$ is not, the boundary point $x^k + \alpha_{\max} d^k$ is tested first. When extrapolation is performed, the resulting iterate may lie on the boundary of the current face, potentially activating multiple bound constraints.

10. *Backtracking (Step 5).* The Armijo-type condition (22) is a typical descent condition ensuring sufficient decrease in all objectives. When Step 5 is reached from Step 3, the search direction $v_{\mathcal{B}}(x^k)$ points outside the current face, so the backtracking process leads the next iterate to a different face. In turn, when Step 5 is reached from Step 4, the procedure successively reduces the step size so that all trial points remain in the interior of the current face, and consequently the next iterate also lies in the same face.

4 Convergence analysis

We now establish the convergence properties of Algorithm 1. The following assumption ensures that the iterates remain in a bounded region.

Assumption 4.1. *The level set* $\mathcal{L}(x^0) \coloneqq \{x \in \mathcal{B} \colon F(x) \leq F(x_0)\}$ *is bounded.*

The next result ensures that Algorithm 1 is well-defined, i.e., if the algorithm does not stop at iteration k, then x^{k+1} is computed after finitely many inner steps.

Theorem 4.2. *Suppose that Assumption 4.1 holds. Then, Algorithm 1 is well defined.*

Proof. Subproblems (11), (12), and (15) are particular cases of (4) and, therefore, admit unique solutions. Moreover, a direction satisfying (16)–(17) at Step 4 always exists, since $d^k = v_{\mathcal{S}}(x^k)$ is a trivial choice. Assume that x^k is a nonstationary point. Then, by Lemma 2.2(c), the algorithm does not terminate at iteration k. From (14) and (17), d^k is a descent direction for F at x^k. We next show that, regardless of the branch executed, the algorithm computes x^{k+1} after finitely many internal iterations.

If Step 5 (Armijo-type backtracking) is executed, then by [23, Lemma 1] there exists $\alpha^* > 0$ such that (22) holds for all $\alpha_{\text{trial}} \in (0, \alpha^*)$. Hence, the backtracking procedure terminates finitely. Now consider the case where Step 4.3 (Extrapolation procedure) is executed. If all bounds in $\mathcal{B}$ are finite, there exists $\bar{\alpha} > 0$ such that $P_{\mathcal{B}}(x^k + \alpha d^k)$ coincides with a corner of $\mathcal{B}$ for every $\alpha \geq \bar{\alpha}$. Therefore, for sufficiently large α_{trial} and α_{new}, $f_j(P_{\mathcal{B}}(x^k + \alpha_{\text{new}} d^k)) = f_j(P_{\mathcal{B}}(x^k + \alpha_{\text{trial}} d^k))$ for all $j \in \{1, \ldots, m\}$, and condition (21) holds with equality, ensuring finite termination. If some bounds are infinite, Assumption 4.1 prevents $\alpha_{\text{trial}} \to \infty$ while all objectives decrease strictly. Indeed, otherwise $P_{\mathcal{B}}(x^k + \alpha_{\text{trial}} d^k) \in \mathcal{L}(x^0)$ for all trials, with $\|P_{\mathcal{B}}(x^k + \alpha_{\text{trial}} d^k)\| \to \infty$, contradicting the boundedness of $\mathcal{L}(x^0)$. Hence, in all cases, the internal procedures terminate finitely and x^{k+1} is well defined. □

Algorithm 1 stops if and only if a Pareto critical point is found. Hence, from now on we assume that $\theta_{\mathcal{B}}(x^k) \neq 0$ for all k, so that the algorithm generates an infinite sequence $\{x^k\}$. The following results establish the convergence properties by analyzing the behavior of Algorithm 1 according to whether test (13) is satisfied infinitely often or only finitely many times.

Theorem 4.3. *Suppose that Assumption 4.1 holds and that $\{x^k\}$ is a sequence generated by Algorithm 1. Assume that there exists an infinite sequence of indices $\mathbb{K} \underset{\infty}{\subset} \mathbb{N}$ such that the test* (13) *holds for all $k \in \mathbb{K}$. Then, every limit point of $\{x^k\}_{k \in \mathbb{K}}$ is Pareto critical of* (1)*.*

Proof. The proof follows from [18, Theorem 1]. □

Theorem 4.4. *Suppose that Assumption 4.1 holds and that $\{x^k\}$ is a sequence generated by Algorithm 1. Assume that the test* (13) *is satisfied only in finitely many iterations. Then every limit point of $\{x^k\}$ is Pareto critical of* (1)*.*

Proof. Regardless of the branch executed, the algorithm enforces $F(x^{k+1}) \leq F(x^k)$ for all $k \in \mathbb{N}$. Hence, $\{F(x^k)\}$ is nonincreasing and $\{x^k\} \subset \mathcal{L}(x^0)$. By Assumption 4.1, $\{x^k\}$ has at least one limit point in $\mathcal{B}$. Let $x^* \in \mathcal{B}$ be such a point, and let $\mathbb{K}_1 \underset{\infty}{\subset} \mathbb{N}$ be such that $\lim_{k \in \mathbb{K}_1} x^k = x^*$. Since $\{F(x^k)\}$ is monotone, by continuity, we have $\lim_{k \in \mathbb{N}} F(x^k) = F(x^*)$.

Since the test (13) is satisfied only finitely many times, there exists $k' \in \mathbb{N}$ such that for all $k \geq k'$ the algorithm executes Step 4. Consequently, for $k \geq k'$, x^{k+1} lies on the same face as x^k or on a face of lower dimension. As the number of faces is finite, there exists $k_0 \geq k'$ such that $x^k \in \mathcal{F}_{A(x^{k_0})}$ for all $k \geq k_0$. For these indices, Step 4.3 (Extrapolation) is never executed immediately following Step 4.2, because in that case x^{k+1} would belong to the boundary of $\mathcal{F}_{A(x^{k_0})}$. Without loss of generality, assume $k \geq k_0$ for all $k \in \mathbb{K}_1$.

Define $s^k := x^{k+1} - x^k = \alpha_k d^k$ for all $k \in \mathbb{N}$. We analyze two possibilities:

(A) $\|s^k\| \geq \delta > 0$ for all $k \in \mathbb{K}_1$;

(B) $\liminf_{k \in \mathbb{K}_1} \|s^k\| = 0$.

In both cases we will show that

$$v_{\mathcal{S}}(x^*) = 0. \tag{23}$$

Note first that, by Lemma 2.2(b), $v_{\mathcal{S}}(\cdot)$ is continuous on $\mathcal{F}_{A(x^{k_0})}$.

Consider case (A). Using conditions (18) or (22) together with (17), and applying Lemma 2.1(a), we obtain for each $j \in \{1, \ldots, m\}$

$$f_j(x^{k+1}) \leq f_j(x^k) + c_1\mathcal{D}(x^k, s^k) \leq f_j(x^k) - c_1\Gamma\|v_{\mathcal{S}}(x^k)\|\|s^k\| \leq f_j(x^k) - c_1\Gamma\delta\|v_{\mathcal{S}}(x^k)\|.$$

Hence

$$c_1\Gamma\delta\|v_{\mathcal{S}}(x^k)\| \leq f_j(x^k) - f_j(x^{k+1}), \quad \forall j \in \{1, \ldots, m\}.$$

Taking limits for $k \in \mathbb{K}_1$, by the continuity of f_j and $v_{\mathcal{S}}(\cdot)$, and the convergence of $\{F(x^k)\}$, we obtain (23).

If (B) takes place, then there exists $\mathbb{K}_2 \underset{\infty}{\subseteq} \mathbb{K}_1$ such that $\lim_{k\in\mathbb{K}_2} s^k = 0$. Let

- $\mathbb{K}_3 \subset \mathbb{K}_2$: indices where α_k is computed at Step 4.1;
- $\mathbb{K}_4 \subset \mathbb{K}_2$: indices where α_k is computed at Step 4.3;
- $\mathbb{K}_5 \subset \mathbb{K}_2$: indices where α_k is computed at Step 5.

At least one of these sets is infinite.

Case (B.1): If $\mathbb{K}_3$ is infinite. From (19) and Lemma 2.1(a),

$$\mathcal{D}\left(x^k + d^k, \tfrac{d^k}{\|d^k\|}\right) \geq c_2\mathcal{D}\left(x^k, \tfrac{d^k}{\|d^k\|}\right), \quad \forall k \in \mathbb{K}_3.$$

Since $\{d^k/\|d^k\|\}$ is bounded, there exist a subsequence $\mathbb{K}_3' \underset{\infty}{\subseteq} \mathbb{K}_3$ and $d^* \in \mathbb{R}^n$ such that $\lim_{k\in\mathbb{K}_3'} d^k/\|d^k\| = d^*$. Taking limits along $\mathbb{K}_3'$ in the above inequality, noting that $\lim_{k\in\mathbb{K}_3} d^k = 0$, and applying Lemma 2.1(b), we obtain $\mathcal{D}(x^*, d^*) \geq c_2\mathcal{D}(x^*, d^*)$, which implies

$$\mathcal{D}(x^*, d^*) \geq 0,$$

because $c_2 < 1$. On the other hand, dividing (17) by $\|d^k\|$ and taking limits along $\mathbb{K}_3'$, we have

$$\mathcal{D}(x^*, d^*) \leq -\Gamma\|v_{\mathcal{S}}(x^*)\|.$$

Combining the two inequalities yields $v_{\mathcal{S}}(x^*) = 0$.

Case (B.2): If $\mathbb{K}_4$ is infinite. Let α^k_{trial} and α^k_{new} denote the last computed α_{trial} and α_{new} at iteration k. Since, for every $k \in \mathbb{K}_1$, x^k lies in a single face, it follows that $\alpha_k = \alpha^k_{\text{trial}} < \alpha_{\max}$ and hence $P_{\mathcal{B}}(x^k + \alpha_k d^k) = x^k + \alpha_k d^k$ for all $k \in \mathbb{K}_4$. Moreover, $\alpha^k_{\text{new}} = \min\{N\alpha_k, \alpha_{\max}\} \in (\alpha_k, N\alpha_k)$, and therefore $P_{\mathcal{B}}(x^k + \alpha^k_{\text{new}} d^k) = x^k + \alpha^k_{\text{new}} d^k$. Consequently, by (21), there exists $j_k \in \{1, \ldots, m\}$ such that

$$f_{j_k}(x^k + \alpha^k_{\text{new}} d^k) \geq f_{j_k}(x^k + \alpha_k d^k), \quad \forall k \in \mathbb{K}_4.$$

By the mean value theorem, there exists $\varepsilon_k \in (\alpha_k, \alpha^k_{\text{new}})$ such that

$$f_{j_k}(x^k + \alpha^k_{\text{new}} d^k) = f_{j_k}(x^k + \alpha_k d^k) + (\alpha^k_{\text{new}} - \alpha_k)\nabla f_{j_k}\left(x^k + \varepsilon_k d^k\right)^\top d^k, \quad \forall k \in \mathbb{K}_4.$$

Combining the two relations and using (3), we obtain

$$\mathcal{D}(x^k + \varepsilon_k d^k, d^k) \geq \nabla f_{j_k}(x^k + \varepsilon_k d^k)^\top d^k \geq 0, \quad \forall k \in \mathbb{K}_4.$$

Since $\lim_{k\in\mathbb{K}_4} \alpha_k d^k = 0$, we also have $\lim_{k\in\mathbb{K}_4} \varepsilon_k d^k = 0$. Thus, dividing the above inequality by $\|d^k\|$ and taking limits along a convergent subsequence of $\{d^k/\|d^k\|\}$, we obtain

$$\mathcal{D}(x^*, d^*) \geq 0.$$

Hence, as in case (B.1), applying (17) yields $v_{\mathcal{S}}(x^*) = 0$.

Case (B.3): If $\mathbb{K}_5$ is infinite. Step 5 is executed after the failure of condition (18) in Step 4.1 or (20) in Step 4.2, which in turn implies that the first trial step size is rejected in the backtracking procedure of Step 5. Thus, for all $k \in \mathbb{K}_5$, there exist $\tilde{\alpha}_k \in (0, \alpha_k/\underline{\omega}]$ and $j_k \in \{1, \ldots, m\}$ such that

$$f_{j_k}(x^k + \tilde{\alpha}_k d^k) > f_{j_k}(x^k) + \tilde{\alpha}_k c_1 \mathcal{D}(x^k, d^k).$$

By the mean value theorem, there exists $\varepsilon_k \in (0, 1)$ such that

$$f_{j_k}(x^k + \tilde{\alpha}_k d^k) = f_{j_k}(x^k) + \tilde{\alpha}_k \nabla f_{j_k}(x^k + \varepsilon_k \tilde{\alpha}_k d^k)^\top d^k, \quad \forall k \in \mathbb{K}_5.$$

Combining the above two inequalities and using (3), we obtain

$$\mathcal{D}(x^k + \varepsilon_k \tilde{\alpha}_k d^k, d^k) \geq \nabla f_{j_k}(x^k + \varepsilon_k \tilde{\alpha}_k d^k)^\top d^k > c_1 \mathcal{D}(x^k, d^k), \quad \forall k \in \mathbb{K}_5.$$

Since $\lim_{k\in\mathbb{K}_5} \alpha_k d^k = 0$, we also have $\lim_{k\in\mathbb{K}_5} \varepsilon_k \tilde{\alpha}_k d^k = 0$. Dividing the above inequality by $\|d^k\|$ and taking limits along a convergent subsequence of $\{d^k/\|d^k\|\}$ yields $\mathcal{D}(x^*, d^*) \geq c_1 \mathcal{D}(x^*, d^*)$, hence

$$\mathcal{D}(x^*, d^*) \geq 0,$$

because $c_1 < 1$. Using (17) as before, we conclude $v_{\mathcal{S}}(x^*) = 0$.

In all cases, (23) holds, and x^* is stationary in the face $\mathcal{F}_{A(x^{k_0})}$. Finally, since $\mathcal{S}_{A(x^k)} = \operatorname{aff}(\mathcal{F}_{A(x^k)}) - x^k$, it follows that $(\overline{\mathcal{F}}_{A(x^k)} - x^k) \subset \mathcal{S}_{A(x^k)}$. Therefore, the optimal value of (15) is no larger than that of (12), i.e.,

$$\mathcal{D}(x^k, v_{\mathcal{S}}(x^k)) + \tfrac{1}{2}\|v_{\mathcal{S}}(x^k)\| \leq \theta_{\overline{\mathcal{F}}}(x^k) \leq 0, \quad \forall k \in \mathbb{K}_1,$$

where the second inequality follows from Proposition 2.2(a). Taking limits along $\mathbb{K}_1$ and using (23) gives $\theta_{\overline{\mathcal{F}}}(x^*) = 0$. Therefore, since $|\theta_{\overline{\mathcal{F}}}(x^k)| > \nu|\theta_{\mathcal{B}}(x^k)|$ for all large k, taking limits along $\mathbb{K}_1$ yields $\theta_{\mathcal{B}}(x^*) = 0$. Hence, by Proposition 2.2(c), x^* is a stationary point for problem (1). □

Next, we present the main convergence result. In essence, it shows that Algorithm 1 is able to compute a Pareto critical point with any desired degree of accuracy.

Theorem 4.5. *Suppose that Assumption 4.1 holds and that $\{x^k\}$ is a sequence generated by Algorithm 1. Then, $\{x^k\}$ admits a limit point that is Pareto critical of* (1).

Proof. The proof follows straightforwardly from Theorems 4.3 and 4.4. □

We finish this section by showing that, under the assumption that $F : \mathbb{R}^n \to \mathbb{R}^m$ is strongly convex on $\mathcal{B}$, if Algorithm 1 produces an infinite sequence, then the entire sequence $\{x^k\}$ converges to a Pareto optimal point. We say that the mapping $F : \mathbb{R}^n \to \mathbb{R}^m$ is strongly convex on $\mathcal{B}$ if, for each $j \in \{1, \ldots, m\}$, there exists a scalar $\gamma_j > 0$ such that

$$f_j(x) \geq f_j(y) + \nabla f_j(y)^\top (x - y) + \tfrac{\gamma_j}{2}\|x - y\|^2, \quad \forall x, y \in \mathcal{B}. \tag{24}$$

Theorem 4.6. *Assume that $F : \mathbb{R}^n \to \mathbb{R}^m$ is strongly convex on $\mathcal{B}$. Suppose that $\{x^k\}$ is a sequence generated by Algorithm 1. Then, the entire sequence $\{x^k\}$ converges to a Pareto optimal point in $\mathcal{B}$.*

Proof. Since F is strongly convex, the level set $\mathcal{L}(x^0) \coloneqq \{x \in \mathcal{B} \colon F(x) \leq F(x_0)\}$ is bounded. Hence, by Theorem 4.5, the sequence $\{x^k\}$ admits a Pareto critical limit point $x^* \in \mathcal{B}$. From the strongly convexity of F, it follows that x^* is a Pareto optimal point (see, for example, [15, Theorem 3.1]). To show that the entire sequence converges to x^*, observe that $\{F(x^k)\}$ is nonincreasing and $\{x^k\} \subset \mathcal{L}(x^0)$, thus continuity arguments imply that $\lim_{k\in\mathbb{N}} F(x^k) = F(x^*)$. Since x^* is stationary, then for all $x \in \mathcal{B}$ there exists at least one index $j(x) \in \{1, \ldots, m\}$ for which

$$\nabla f_{j(x)}(x^*)^\top (x - x^*) \geq 0.$$

In particular,

$$\nabla f_{j(x^k)}(x^*)^\top (x^k - x^*) \geq 0, \quad \forall k \in \mathbb{N}.$$

Then, by the strong convexity condition (24) of $f_{j(x^k)}$, we obtain

$$f_{j(x^k)}(x^k) - f_{j(x^k)}(x^*) \geq \tfrac{\gamma_{j(x^k)}}{2}\|x^k - x^*\|^2 \geq \tfrac{\gamma_{\min}}{2}\|x^k - x^*\|^2, \quad \forall k \in \mathbb{N},$$

where $\gamma_{\min} \coloneqq \min_{j\in\{1,\ldots,m\}} \gamma_j$. Taking limits for $k \in \mathbb{N}$ in the above inequality and using that $\lim_{k\in\mathbb{N}} f_j(x^k) = f_j(x^*)$ for all $j \in \{1, \ldots, m\}$, we conclude that the entire sequence $\{x^k\}$ converges to x^*. □

5 Dual nondegeneracy and finite active-set identification

In this section, we show that if every Pareto critical point of (1) is *dual-nondegenerate*, then Algorithm 1 identifies in finitely many iterations the face of the feasible region to which a limit point belongs. In other words, the iterates eventually remain on a single face of $\mathcal{B}$. Once the face is identified, the algorithm effectively behaves as an unconstrained multiobjective method restricted to that face.

Definition 5.1. *Let $x^* \in \mathcal{B}$ be a Pareto critical point of* (1)*. Let $w^* \in \mathbb{R}^n$ be any vector satisfying*

$$v_{\mathcal{B}}(x^*) = P_{\mathcal{B}}\big(x^* - w^*\big) - x^*,$$

whose existence follows from Proposition 2.3. We say that x^* is dual-degenerate *if there exists an index $i \in \{1, \ldots, n\}$ such that $x_i^* = \ell_i$ or $x_i^* = u_i$, while $w_i^* = 0$. Otherwise, x^* is called* dual-nondegenerate*.*

Remark 1. *From the KKT conditions of the differentiable formulation* (7) *at a Pareto critical point x^*, one obtains $w^* = \mu^\ell - \mu^u$, where $\mu^\ell, \mu^u \in \mathbb{R}^n_+$ are the Lagrange multipliers associated with the bound constraints $d \geq \ell - x^*$ and $d \leq u - x^*$, respectively. For any index i that is active at x^*, strict complementarity ($x_i^* = \ell_i$ with $\mu_i^\ell > 0$ or $x_i^* = u_i$ with $\mu_i^u > 0$) is equivalent to $w_i^* \neq 0$. Thus, dual-nondegeneracy of x^* is precisely the strict complementarity of the box constraints in the differentiable model* (7)*.*

We now formalize the finite identification result.

Theorem 5.2. *Assume that every Pareto critical point of* (1) *is dual-nondegenerate. Suppose that Assumption 4.1 holds and that* $\{x^k\}$ *is a sequence generated by Algorithm 1. Then the number of iterations for which* (13) *holds is finite. Consequently, after a finite number of iterations all subsequent iterates* x^k *lie on a single face of* $\mathcal{B}$*.*

Proof. Let $\mathbb{K} \subset \mathbb{N}$ be the set of indices for which test (13) holds. Suppose, by contradiction, that $\mathbb{K}$ is infinite. Since the number of faces of $\mathcal{B}$ is finite, there exists a face $\mathcal{F}_A$ and an infinite subset $\mathbb{K}_1 \underset{\infty}{\subset} \mathbb{K}$ such that $x^k \in \mathcal{F}_A$ and $x^{k+1} \notin \overline{\mathcal{F}}_A$ for all $k \in \mathbb{K}_1$. Let $x^* \in \mathcal{B}$ be a limit point of the subsequence $\{x^k\}_{k\in\mathbb{K}_1}$. By Theorem 4.3, it follows that x^* is a stationary point and hence $v_{\mathcal{B}}(x^*) = 0$.

For each k, from Proposition 2.3, there exists a vector $w^k \in \operatorname{conv}\{\nabla f_j(x^k)\}_{j=1}^m$ such that

$$v_{\mathcal{B}}(x^k) = P_{\mathcal{B}}(x^k - w^k) - x^k. \tag{25}$$

Writing $w^k = \sum_{j=1}^m \lambda_j^k \nabla f_j(x^k)$ with $\lambda^k \in \Delta_m$, compactness of Δ_m ensures that there exist $\mathbb{K}_2 \underset{\infty}{\subset} \mathbb{K}_1$ and $\lambda^* \in \Delta_m$ such that $\lim_{k\in\mathbb{K}_2} \lambda^k = \lambda^*$. Let $w^* := \sum_{j=1}^m \lambda_j^* \nabla f_j(x^*)$. Taking limits along $\mathbb{K}_2$ in (25), and using the continuity of both $v_{\mathcal{B}}(\cdot)$ and the projection operator, we obtain

$$v_{\mathcal{B}}(x^*) = P_{\mathcal{B}}(x^* - w^*) - x^*.$$

Without loss of generality, assume that on $\mathcal{F}_A$ the first p coordinates of x are free and the remaining coordinates satisfy $x_i = \ell_i$, $i \in \{p+1, \ldots, n\}$. Dual nondegeneracy of x^* implies $w_i^* > 0$ for all $i \in \{p+1, \ldots, n\}$, see Remark 1. Hence, for all sufficiently large $k \in \mathbb{K}_2$, we also have $w_i^k > 0$ for these indices. Since $x_i^k = \ell_i$ and $w_i^k > 0$, we have $x_i^k - w_i^k < \ell_i$ and hence $[P_{\mathcal{B}}(x^k - w^k)]_i = \ell_i$ for all $i \in \{p+1, \ldots, n\}$, which implies

$$[v_{\mathcal{B}}(x^k)]_i = 0, \quad \forall i \in \{p+1, \ldots, n\}.$$

Therefore, for all sufficiently large $k \in \mathbb{K}_2$,

$$v_{\mathcal{B}}(x^k) \in \overline{\mathcal{F}}_A - x^k.$$

By the definitions of $\theta_{\mathcal{B}}(x^k)$ and $\theta_{\overline{\mathcal{F}}}(x^k)$, the above inclusion implies

$$\theta_{\overline{\mathcal{F}}}(x^k) \leq \theta_{\mathcal{B}}(x^k), \qquad \text{for all sufficiently large } k \in \mathbb{K}_2.$$

On the other hand, since it is always true that $\theta_{\mathcal{B}}(x^k) \leq \theta_{\overline{\mathcal{F}}}(x^k)$, we obtain

$$\theta_{\overline{\mathcal{F}}}(x^k) = \theta_{\mathcal{B}}(x^k), \qquad \text{for all sufficiently large } k \in \mathbb{K}_2.$$

For such indices, test (13) cannot hold, contradicting the fact that $\mathbb{K}_2 \subset \mathbb{K}$. Therefore, $\mathbb{K}$ must be finite, and from some iteration onward the algorithm never abandons the current face. Thus all subsequent iterates lie on a single face of $\mathcal{B}$. □

6 Implementation aspects

We now describe the key implementation components of Algorithm 1, covering the projected gradient subproblems, the face-abandoning and face-exploring directions, and the line search procedure.

6.1 Projected gradient subproblems

We first observe that, when $\Omega = \mathbb{R}^n$, problem (4) reduces to the *steepest descent subproblem*. Introducing multipliers $\lambda \in \Delta_m$ yields the dual problem

$$\min_{\lambda} \ \tfrac{1}{2}\Big\|\sum_{j=1}^{m} \lambda_j \nabla f_j(x)\Big\|^2 \quad \text{s.t.} \qquad \lambda \in \Delta_m, \tag{26}$$

whose solution λ^{SD} gives $d_{\mathrm{SD}}(x) = -\sum_{j=1}^{m} \lambda_j^{\mathrm{SD}} \nabla f_j(x)$. In practice, this dual form is preferable to the primal formulation (7), especially for large n; moreover, for $m = 2$ it admits a closed-form solution.

A practical shortcut for computing the projected gradient direction $v(x)$ follows from Proposition 2.3; see also [13, Remark 3]. After evaluating $d_{\mathrm{SD}}(x)$ via the dual formulation (26), we define

$$v_{\mathrm{trial}}(x) \coloneqq P_\Omega\big(x + d_{\mathrm{SD}}(x)\big) - x.$$

By Proposition 2.3, $v(x) = v_{\mathrm{trial}}(x)$ if and only if

$$\{j \in \{1, \dots, m\} : \lambda_j^{\mathrm{SD}} > 0\} = J\big(x, v_{\mathrm{trial}}(x)\big),$$

where $J(\cdot, \cdot)$ is defined in (8). In particular, this condition always holds when $x + d_{\mathrm{SD}}(x) \in \Omega$. Otherwise, the projected gradient $v(x)$ is obtained by solving the primal differentiable formulation (7). Thus, in many iterations $v(x)$ is computed from a single projection, avoiding the solution of (7) and reducing the computational cost.

The projected gradient directions required at Steps 3 and 4 of Algorithm 1 are obtained by solving instances of problem (4) for different feasible sets Ω:

- In the *full-space subproblem* (11), $\Omega = \mathcal{B}$, so all n components of d are decision variables.
- In the *face-restricted subproblem* (12), $\Omega = \overline{\mathcal{F}}_{A(x^k)}$, and only the free coordinates d_i, $i \in I(x^k)$, are decision variables; the remaining components are set to zero.
- In the *affine-restricted subproblem* (15), $\Omega = \mathrm{aff}(\mathcal{F}_{A(x^k)})$, so the optimization is unconstrained in the free coordinates; again, the full vector is obtained by inserting zeros in the active coordinates. Subproblem (15) is precisely the unconstrained version of (12).

Subproblem (15) is always solved using the dual steepest descent formulation (26), after restricting each gradient $\nabla f_j(x^k)$ to the free index set $I(x^k)$. The resulting direction is then embedded into $\mathbb{R}^n$ by setting $d_i = 0$ for $i \notin I(x^k)$, yielding the affine-restricted steepest descent direction $v_{\mathcal{S}}(x^k)$. For the full-space and face-restricted subproblems (11) and (12), the shortcut described above is employed: we compute d_{SD}, project it, and verify the active set condition; if it fails, we solve the primal differentiable formulation (7).

6.2 Face-abandoning direction: Barzilai–Borwein spectral variant

For the face-abandoning step (Step 3), instead of always using the standard projected gradient direction $v_{\mathcal{B}}(x^k)$, we employ a *Barzilai–Borwein spectral* variant of the projected gradient subproblem, following [1, 31]; see also [4]. Spectral scalings are known to substantially accelerate gradient-based schemes, and this idea has been used successfully in active-set algorithms for scalar minimization [2, 19]. More recently, it has also been applied in the multiobjective setting [6, 7, 26].

We compute the Barzilai–Borwein parameter as

$$\beta_k = \begin{cases} \dfrac{s_k^\top s_k}{\mathcal{D}(x^k, s_k) - \mathcal{D}(x^{k-1}, s_k)}, & \text{if } \mathcal{D}(x^k, s_k) - \mathcal{D}(x^{k-1}, s_k) > 0, \\ \max\{1,\ \|x^k\|_\infty / \|v_{\mathcal{B}}(x^k)\|_\infty\}, & \text{otherwise,} \end{cases}$$

where $s_k \coloneqq x^k - x^{k-1}$. The parameter β_k is then safeguarded by projection onto the interval $[\beta_{\min}, \beta_{\max}]$, with $0 < \beta_{\min} \leq 1 \leq \beta_{\max}$. We then solve

$$d_{\mathrm{BB}}(x^k) \coloneqq \operatorname*{argmin}_{d \in \mathcal{B} - x^k} \ \beta_k \, \mathcal{D}(x^k, d) + \tfrac{1}{2}\|d\|^2. \tag{27}$$

We define the face-abandoning direction as

$$d^k \coloneqq \begin{cases} d_{\mathrm{BB}}(x^k), & \text{if } \exists\, i \notin I(x^k) \text{ such that } [d_{\mathrm{BB}}(x^k)]_i \neq 0, \\ v_{\mathcal{B}}(x^k), & \text{otherwise.} \end{cases}$$

Thus the algorithm uses the spectral direction whenever it produces a component pointing outside the current face; otherwise, it falls back to the classical projected gradient direction.

Note that when $\beta_k = 1$, problem (27) reduces to the standard projected gradient subproblem (11). Moreover, for any value of β_k the structure of the projected gradient subproblem is preserved, so the same procedure described in Section 6.1 applies directly here. Finally, the spectral strategy preserves all theoretical properties required for our analysis. In particular, the convergence results for the projected gradient framework remain valid, and Theorem 4.3 follows directly from [14, Theorem 1].

6.3 Face-exploring direction: truncated Newton-Gradient

In Step 4, we compute the search direction as a *truncated Newton-Gradient* direction. The Newton-Gradient direction, introduced in [17], combines the steepest-descent Lagrange multipliers of subproblem (15) with second-order information of the objective functions. In order to guarantee that the search direction satisfies the angle condition (17), we adopt a truncated variant of this scheme, in the spirit of the truncated Newton approach of [2], as discussed next.

Let $\lambda \in \Delta_m$ be the Lagrange multiplier associated with the solution of subproblem (15) (see Section 6.1). Recall that $v_{\mathcal{S}}(x^k) = -\sum_{j=1}^m \lambda_j \nabla f_j(x^k)$, and $[v_{\mathcal{S}}(x^k)]_i = 0$ for every $i \notin I(x^k)$ (fixed variables). Using the same multipliers, we build the scalarized gradient and Hessian

$$g \coloneqq \sum_{j=1}^m \lambda_j \nabla f_j(x^k), \qquad H \coloneqq \sum_{j=1}^m \lambda_j \nabla^2 f_j(x^k).$$

Restricting these quantities to the free variables, we define

$$g_{\mathrm{red}} \coloneqq (g_i)_{\,i\in I(x^k)}, \qquad H_{\mathrm{red}} \coloneqq (H_{ij})_{\,i,j\in I(x^k)}.$$

We then consider the reduced quadratic model

$$\min_s \; q_k(s) \coloneqq \tfrac{1}{2} s^\top H_{\mathrm{red}} s + g_{\mathrm{red}}^\top s.$$

This problem is solved approximately by a truncated conjugate-gradient (CG) procedure as in [2, Algorithm 4.1], initialized with $s = 0$. The first CG iterate is a multiple of $-g_{\mathrm{red}}$, and since $-g_{\mathrm{red}} = [v_{\mathcal{S}}(x^k)]_{\,i\in I(x^k)}$, the angle condition (17) holds trivially. During the CG iterations we test condition (17): if it is violated, the CG process is truncated and the last iterate satisfying the condition is taken as the reduced search direction. Finally, the search direction d^k is obtained by extending the final CG iterate to $\mathbb{R}^n$ by setting the components corresponding to the fixed variables to zero. By construction, this guarantees that condition (16) holds automatically.

6.4 Armijo backtracking line search

The backtracking procedure used in Step 5 employs a safeguarded quadratic interpolation strategy. Given a trial step size α_{trial}, the Armijo condition (22) is evaluated componentwise. If all components satisfy (22), the step size is accepted. Otherwise, let j be an index for which (22) fails, and define $\varphi(\alpha) \coloneqq f_j(x^k + \alpha d^k)$. We compute the minimizer of the quadratic model that interpolates $\varphi(0)$, $\varphi'(0)$, and $\varphi(\alpha_{\mathrm{trial}})$, which yields

$$\alpha_q = -\tfrac{1}{2}\varphi'(0)\,\alpha_{\mathrm{trial}}^2 / \left[\varphi(\alpha_{\mathrm{trial}}) - \varphi(0) - \varphi'(0)\,\alpha_{\mathrm{trial}}\right].$$

If $\alpha_q \in [\underline{\omega}\,\alpha_{\mathrm{trial}}, \bar{\omega}\,\alpha_{\mathrm{trial}}]$, we set $\alpha_{\mathrm{new}} = \alpha_q$; otherwise, we set $\alpha_{\mathrm{new}} = \alpha_{\mathrm{trial}}/2$. The trial stepsize is then replaced by α_{new}. Once such an index j is selected, the corresponding function f_j is kept as the reference during the backtracking loop, and the step size is refined until this function satisfies the Armijo condition, after which all objectives are retested.

7 Numerical experiments

This section presents numerical results aimed at assessing the reliability and practical performance of MAMBO. All experiments were conducted using the Julia programming language (version 1.12.3) on a computer equipped with a 3.7 GHz Intel Core i5 6-Core processor and 8 GB of 2667 MHz DDR4 RAM, running macOS Sequoia 15.7.2. All source codes used in the experiments are publicly available `https://github.com/lfprudente/MAMBO-paper`.

The following methods are considered in the reported numerical tests:

- **MAMBO**: the active-set algorithm described in Algorithm 1, incorporating all implementation aspects discussed in Section 6.
- **PG**: the classical projected gradient method with Armijo line search, originally proposed in [18]. We use our own implementation.

- **PG-BB**: the Barzilai–Borwein projected gradient method with Armijo line search, proposed in [6]. We use our own implementation.
- **NSGA-II**: the Nondominated Sorting Genetic Algorithm II proposed in [10]. It is a population-based evolutionary metaheuristic widely regarded as a state-of-the-art method in multiobjective optimization. We employ the Julia implementation provided by the `Metaheuristics` package [9].

All deterministic algorithms (MAMBO, PG, and PG-BB) employ the same stopping criterion to declare convergence. Specifically, an iterate x^k is considered Pareto critical when

$$|\theta_{\mathcal{B}}(x^k)| \leq 5 \times \sqrt{\texttt{eps}},$$

where $\texttt{eps} \approx 2.22 \times 10^{-16}$ denotes the machine precision. In addition, a maximum of 2000 iterations is imposed. If this limit is reached, the algorithm is declared to have failed.

The projected gradient subproblems arising in the algorithms, either in their primal or dual formulations, were solved using the `RipQP` package [27] from the JuliaSmoothOptimizers organization. The algorithmic parameters used by MAMBO (and, when applicable, by PG and PG-BB) are fixed throughout all experiments and are given by: $\nu = 0.1$, $\Gamma = 10^{-6}$, $c_1 = 10^{-4}$, $c_2 = 0.5$, $N = 2$, $\underline{\omega} = 0.05$, and $\bar{\omega} = 0.95$.

7.1 Test problems

The numerical evaluation was carried out on a collection of 63 box-constrained multiobjective optimization problems commonly used in the literature. Except for the problems of the F1–F9 family (see [20]), the test problems considered in this work are summarized in [30], where their original references are provided. For conciseness, individual references are omitted here. Table 1 summarizes the test problems used in the experiments, reporting the number of variables n, the number of objectives m, and the corresponding box constraints. Some problems in the F1–F9 and ZDT test suites are not differentiable at $x_1 = 0$. Therefore, since the deterministic methods rely on first-order information, the lower bound of the first variable was slightly modified to $x_1 \in [10^{-6}, 1]$.

7.2 Performance metrics

The performance of the algorithms was assessed using standard metrics for evaluating approximations of the Pareto front. To account for numerical inaccuracies, dominance relations were evaluated using combined relative and absolute tolerances. Given two objective vectors $u, v \in \mathbb{R}^m$, u is said to dominate v if $u_i \leq v_i + \varepsilon_i$ for all $i \in \{1, \ldots, m\}$ and $u_j < v_j - \varepsilon_j$ for some $j \in \{1, \ldots, m\}$, where $\varepsilon_i \coloneqq \sqrt{\texttt{eps}} \times \max(1, \max\{|u_i|, |v_i|\})$. Vectors satisfying $|u_i - v_i| \leq \varepsilon_i$ for all i are considered numerically equivalent.

Purity Purity measures the proportion of solutions produced by an algorithm that belong to a reference non-dominated set. Given a set of solutions A generated by an algorithm and a reference set R, Purity is defined as

$$\text{Purity}(A, R) = |A \cap R|/|A|.$$

Problem	n	m	Bounds
AP1	2	3	$[-10, 10]^n$
AP3	2	2	$[-100, 100]^n$
AP4	3	3	$[-10, 10]^n$
DD1	5	2	$[-20, 20]^n$
F1	10	2	$[10^{-6}, 1] \times [0, 1]^{n-1}$
F2	30	2	$[10^{-6}, 1] \times [-1, 1]^{n-1}$
F3	30	2	$[10^{-6}, 1] \times [-1, 1]^{n-1}$
F4	30	2	$[10^{-6}, 1] \times [-1, 1]^{n-1}$
F5	30	2	$[10^{-6}, 1] \times [-1, 1]^{n-1}$
F6	10	3	$[0, 1]^2 \times [-2, 2]^{n-2}$
F7	10	2	$[10^{-6}, 1]^n$
F9	30	2	$[0, 1] \times [-1, 1]^{n-1}$
FA1	3	3	$[0, 1]^n$
Far1	2	2	$[-1, 1]^n$
FDS	200	3	$[-2, 2]^n$
FF1	2	2	$[-1, 1]^n$
Hil1	2	2	$[0, 1]^n$
IKK1	2	3	$[-50, 50]^n$
IM1	2	2	$[1, 4] \times [1, 2]$
JOS1	100	2	$[-100, 100]^n$
KW2	2	2	$[-3, 3]^n$
LE1	2	2	$[-5, 10]^n$
Lov1	2	2	$[-10, 10]^n$
Lov2	2	2	$[-0.75, 0.75]^n$
Lov3	2	2	$[-20, 20]^n$
Lov4	2	2	$[-20, 20]^n$
Lov5	3	2	$[-2, 2]^n$
Lov6	2	2	$[0.1, 0.425] \times [-0.16, 0.16]^{n-1}$
LTDZ	3	3	$[0, 1]^n$
MGH9	3	15	$[-2, 2]^n$
MGH16	4	5	$[-25, 25] \times [-5, 5]^2 \times [-1, 1]$
MGH26	4	4	$[-1, 1]^n$

Problem	n	m	Bounds
MGH33	10	10	$[-1, 1]^n$
MHHM2	2	3	$[0, 1]^n$
MLF2	2	2	$[-100, 100]^n$
MMR1	2	2	$[0.1, 1] \times [0, 1]$
MMR3	2	2	$[-1, 1]^n$
MMR4	3	2	$[0, 4]^n$
MOP2	2	2	$[-1, 1]^n$
MOP3	2	2	$[-\pi, \pi]^n$
MOP5	2	3	$[-1, 1]^n$
MOP6	2	2	$[0, 1]^n$
MOP7	2	3	$[-400, 400]^n$
PNR	2	2	$[-2, 2]^n$
QV1	100	2	$[-5, 5]^n$
SD	4	2	$[1, 3] \times [\sqrt{2}, 3]^2 \times [1, 3]$
SK2	4	2	$[-10, 10]^4$
SLCDT1	2	2	$[-1.5, 1.5]^n$
SLCDT2	100	3	$[-1, 1]^n$
SP1	2	2	$[-100, 100]^n$
TKLY1	4	2	$[0.1, 1] \times [0, 1]^{n-1}$
Toi4	4	2	$[-2, 5]^4$
Toi8	3	3	$[-1, 1]^4$
Toi9	4	4	$[-1, 1]^4$
Toi10	4	3	$[-2, 2]^4$
VU1	2	2	$[-3, 3]^n$
VU2	2	2	$[-3, 3]^n$
ZDT1	30	2	$[10^{-6}, 1] \times [0, 1]^{n-1}$
ZDT2	30	2	$[0, 1]^n$
ZDT3	30	2	$[10^{-6}, 1] \times [0, 1]^{n-1}$
ZDT4	10	2	$[10^{-6}, 1] \times [-5, 5]^{n-1}$
ZDT6	10	2	$[0, 1]^n$
ZLT1	100	5	$[-1000, 1000]^n$

Table 1: List of test problems.

In our experiments, R is defined as the collection of all non-dominated points obtained by all considered algorithms.

Γ-Spread The Γ-Spread metric evaluates the uniformity of the distribution of solutions along the Pareto front. It is computed as the maximum gap between consecutive solutions after sorting them according to one objective.

Hypervolume The Hypervolume (HV) metric quantifies the volume of the objective space dominated by a given approximation of the Pareto front and bounded by a reference point. As commonly adopted, a normalized hypervolume was considered. Let z^{ideal} and z^{nadir} denote the *ideal* and *nadir* points estimated from the reference non-dominated set. Each objective vector z was normalized componentwise as

$$\tilde{z}_j = (z_j - z_j^{\text{ideal}})/(z_j^{\text{nadir}} - z_j^{\text{ideal}}), \quad j = 1, \ldots, m.$$

The reference point was set to $r = (1.1, \ldots, 1.1)$, and the resulting hypervolume value was divided by 1.1^m, ensuring that the final value lies in $[0, 1]$.

Covering metric The Covering metric, also known as the *C-metric*, compares the dominance relations between two approximation sets. Given two sets A and B, the covering of B by A is defined as

$$C(A, B) = |\{\, b \in B : \exists a \in A \text{ such that } a \text{ dominates } b \,\}|/|B|.$$

The metric is asymmetric and quantifies the dominance of A over B.

7.3 Comparison with projected gradient methods

This subsection compares MAMBO with two classical deterministic descent-type algorithms, namely the projected gradient method (PG) and its Barzilai–Borwein variant (PG-BB). For each test problem, all three algorithms were executed from the same set of 300 initial points uniformly sampled from the feasible box. Performance profiles in

the sense of Dolan and Moré [12] were employed to compare the algorithms across all problems and initial points.

Performance profiles based on CPU time were constructed by comparing the algorithms on a run-to-run for each initial point. For a given problem and initial point, an algorithm was considered to have successfully solved the instance if the final solution obtained was not dominated by those produced by the other algorithms starting from the same initial point. Figure 2 reports the resulting performance profile with respect to CPU time. The results show that MAMBO consistently outperforms the classical projected gradient method and its Barzilai–Borwein variant in terms of computational efficiency, exhibiting a higher probability of being the fastest method across the test set. Moreover, the right-hand side of the performance profile indicates that MAMBO also achieves the highest overall success rate, meaning that it most frequently produces final solutions that are not dominated by those obtained by the competing methods. The PG-BB method generally improves upon the classical PG approach but remains less competitive than MAMBO both in terms of speed and success rate in most instances.

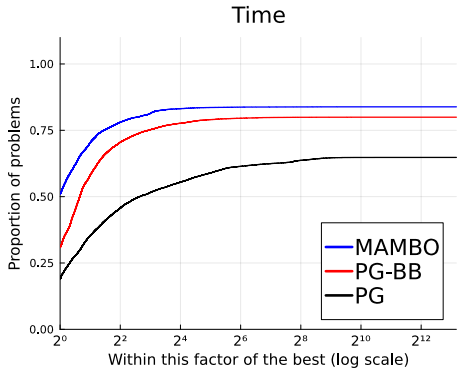

Figure 2: Performance profile comparing MAMBO, PG, and PG-BB with respect to CPU time.

For each algorithm and each test problem, the approximation of the Pareto front was defined as the union of all non-dominated solutions obtained over the 300 runs of the algorithm. A reference Pareto front was then constructed as the union of all non-dominated solutions generated by MAMBO, PG, and PG-BB across all runs. The quality of the approximated Pareto fronts was assessed using the Purity, Γ-Spread, and Hypervolume metrics. Since larger values of Purity and Hypervolume indicate better performance, their reciprocals were considered when constructing the corresponding performance profiles.

The results in Figure 3 indicate that MAMBO achieves superior or comparable Pareto front quality relative to PG and PG-BB. In particular, MAMBO tends to generate approximation sets with higher purity and larger hypervolume. With respect to the Γ-Spread metric, MAMBO and PG-BB exhibit very similar performance and both outperform the classical projected gradient method, indicating a more uniform distribution of solutions along the Pareto front. Although not reported explicitly, we also evaluated the Δ-Spread metric as an alternative measure of distribution. For this metric, all three algorithms exhibited comparable performance across the test set, suggesting that differences in solution spread are mainly captured by the Γ-Spread results.

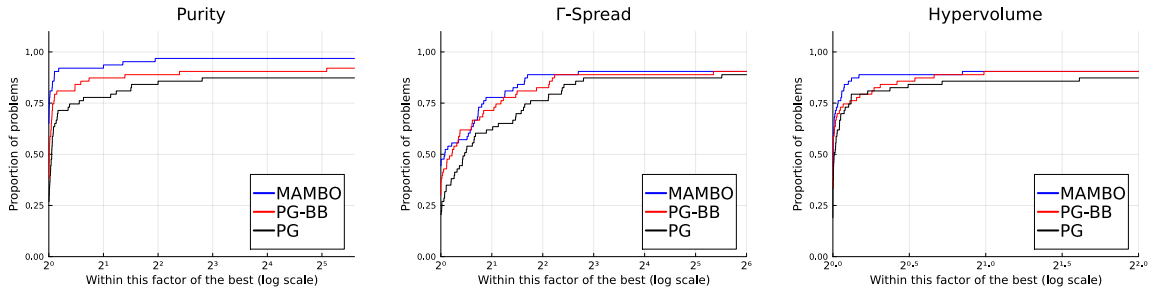

Figure 3: Performance profiles comparing MAMBO, PG, and PG-BB with respect to Purity, Γ-Spread, and Hypervolume. For Purity and Hypervolume, inverse values are used so that smaller values indicate better performance.

In order to provide a qualitative illustration of the behavior of MAMBO, Figure 4 depicts the Pareto front approximations obtained for the problems F1, Hil1, and ZDT3. In the graphics, a filled circle represents a final iterate, while the starting point of each straight segment corresponds to the associated initial point. Pareto-optimal solutions are highlighted using square markers. The plots illustrate how different initial points are driven toward the Pareto front, yielding well-distributed non-dominated solutions across the objective space.

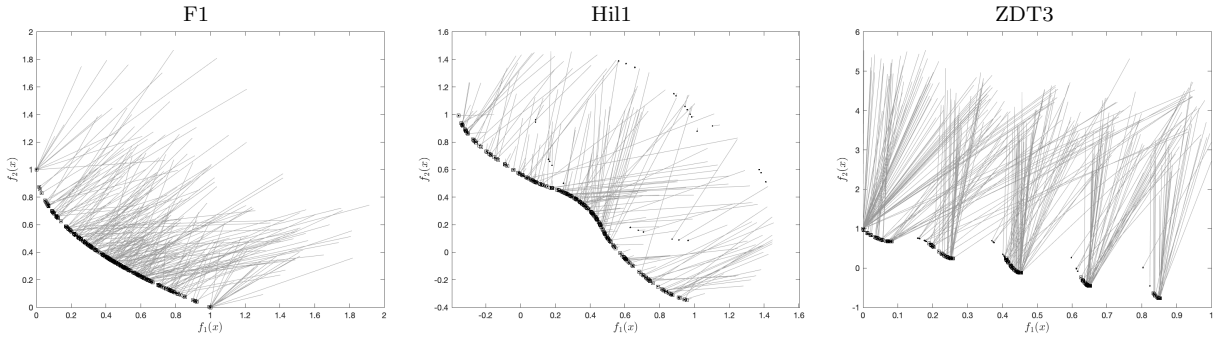

Figure 4: Illustrative Pareto front approximations obtained by MAMBO for F1, Hil1, and ZDT3 test problems.

Overall, the performance profile analysis indicates that MAMBO attains a favorable balance between computational efficiency and solution quality when compared with classical projected gradient methods. While PG-BB improves upon the basic projected gradient scheme through the use of spectral step-size information, it remains less competitive in several instances. The observed performance of MAMBO reflects the combined effect of its algorithmic components, including active-set identification and the exploitation of second-order information, which together contribute to its robustness and efficiency across a wide range of problems.

7.4 Comparison with NSGA-II

This section compares MAMBO with the evolutionary algorithm NSGA-II, which represents a fundamentally different class of methods. While MAMBO is a deterministic descent-type algorithm, NSGA-II is a population-based stochastic metaheuristic. As a

consequence, the comparison is necessarily indirect and focuses on the quality of the Pareto front approximations obtained within comparable computational budgets.

NSGA-II was executed using the default parameter settings provided by the `Metaheuristics` package, including an initial population size of 100 individuals and 500 generations. Due to its stochastic nature, NSGA-II was run five independent times for each selected test problem, producing different populations at each execution. All solutions generated throughout the five runs were collected, including all intermediate populations, rather than restricting the analysis to the final populations.

The final Pareto front approximation associated with NSGA-II was then defined as the set of non-dominated points obtained from the union of these solutions. The total CPU time reported for NSGA-II corresponds to the sum of the execution times of the five independent runs. In order to ensure a fair comparison in terms of computational effort, MAMBO was then executed for the same total CPU time using multiple initial points sampled from the feasible box.

The comparison was carried out on a subset of test problems selected from Table 1, focusing on problems originally designed for multiobjective optimization and with dimension $n \geq 10$. In particular, the selected instances include well-known benchmark problems from the F1–F9 and ZDT test suites, which are widely used in the literature and are known to pose difficulties for deterministic descent-type methods.

Table 2 reports the comparative results between MAMBO and NSGA-II in terms of Purity, Γ-Spread, Hypervolume, and the Covering metric. All values are reported in percentage form, except for the Γ-Spread metric, for which smaller values indicate a more uniform distribution of solutions.

Problem	Purity (%)		Γ-Spread		Hypervolume (%)		Covering (%)	
	MAMBO	NSGA-II	MAMBO	NSGA-II	MAMBO	NSGA-II	(M,N)	(N,M)
F1	**100.00**	0.21	**0.005**	0.025	**72.42**	72.09	**99.79**	0.00
F2	**100.00**	4.81	**0.022**	0.157	**72.39**	69.22	**95.19**	0.00
F3	**100.00**	0.00	**0.004**	0.108	**72.41**	69.99	**100.00**	0.00
F4	**100.00**	0.00	**0.006**	0.220	**72.41**	69.50	**100.00**	0.00
F5	**100.00**	0.00	**0.004**	0.101	**72.41**	69.95	**100.00**	0.00
F6	**98.75**	88.36	3.833	**2.773**	**92.08**	69.47	**11.64**	1.25
F7	10.00	**100.00**	**0.825**	1.535	82.17	**94.27**	0.00	**90.00**
F9	**100.00**	15.52	**0.006**	0.266	**44.86**	40.19	**84.48**	0.00
FDS	**100.00**	77.79	**4.292e+5**	2.665e+6	**41.50**	28.84	**22.21**	0.00
JOS1	**100.00**	0.00	**0.237**	2.238	**85.80**	68.15	**100.00**	0.00
QV1	**100.00**	75.26	0.380	**0.005**	25.95	**43.47**	**24.74**	0.00
SLCDT2	**100.00**	9.96	10.162	**9.935**	**49.52**	36.61	**90.04**	0.00
ZDT1	80.00	**99.75**	0.983	**0.002**	20.71	**72.36**	0.25	**20.00**
ZDT2	**100.00**	99.86	1.000	**0.009**	9.09	**44.85**	**0.14**	0.00
ZDT3	**99.86**	44.69	0.171	**0.171**	**60.08**	60.06	**55.31**	0.14
ZDT4	0.00	**100.00**	30.420	**0.002**	0.00	**72.39**	0.00	**100.00**
ZDT6	33.33	**99.73**	2.729	**0.003**	23.25	**50.87**	0.27	**66.67**

Table 2: Comparison between MAMBO and NSGA-II using Purity, Γ-Spread, Hypervolume, and the Covering metric. For the Covering metric, (M, N) denotes the percentage of solutions generated by NSGA-II that are dominated by solutions obtained by MAMBO, while (N, M) denotes the converse. Best values are highlighted in bold.

Overall, the results indicate that MAMBO is able to produce Pareto front approximations that are competitive with those obtained by NSGA-II within the same computational time budget. In several instances, MAMBO attains higher Purity and Hypervolume values, suggesting a closer agreement with the reference non-dominated set. The Covering metric further shows that a significant portion of the solutions generated by NSGA-II are dominated by those obtained by MAMBO in a number of test problems.

The favorable performance of MAMBO in terms of Purity and Covering is consistent with its theoretical properties. MAMBO is designed to converge to Pareto critical

points, whereas NSGA-II, as a stochastic evolutionary algorithm, does not explicitly enforce first-order optimality conditions. As a consequence, although NSGA-II is effective at approximating the geometry of the Pareto front, the solutions it generates are not guaranteed to satisfy Pareto criticality. This distinction is naturally reflected in metrics such as Purity and Covering, which favor solutions that lie closer to the set of Pareto critical points.

With respect to solution distribution, the results obtained using the Γ-Spread metric indicate a balanced performance between MAMBO and NSGA-II across the test set. This outcome is noteworthy, given that NSGA-II explicitly incorporates diversity-preservation mechanisms as part of its population-based framework, whereas the diversity of the solutions generated by MAMBO in the present experiments is solely induced by the initial points, which were sampled in a naive manner from the feasible box.

It is also worth noting that MAMBO exhibits more limited performance on some ZDT test problems, with the exception of ZDT3. These problems are known to admit a large number of Pareto critical and weakly Pareto-optimal points. While MAMBO correctly converges to Pareto critical solutions in accordance with the theoretical results, many of these points are subsequently discarded during the dominance filtering step used to construct the final Pareto front approximation. As a result, the quality of the final non-dominated set may be adversely affected, despite the correct convergence behavior of the algorithm.

Taken together, these results suggest that, despite relying on fundamentally different optimization paradigms, MAMBO is capable of generating high-quality Pareto front approximations within a comparable computational effort, even on problems traditionally considered challenging for deterministic methods. At the same time, the results highlight the importance of initialization strategies for descent-type multiobjective algorithms, motivating the investigation of more effective ways to generate initial points, potentially by combining heuristic exploration mechanisms with subsequent refinement using MAMBO.

8 Final remarks

In this paper, we proposed a deterministic active-set framework for smooth multiobjective optimization problems subject to box constraints, referred to as the *Multiobjective Active-set Method for Box-constrained Optimization (MAMBO)*. The method alternates between face-exploring and face-abandoning iterations, combining first and second-order information with adaptive working-set updates. Global convergence to Pareto critical points was established, and finite identification of the active set was proved under a dual nondegeneracy assumption. Numerical experiments demonstrated that MAMBO achieves a favorable balance between computational efficiency and the quality of the approximated Pareto fronts when compared with classical projected gradient methods and NSGA-II.

An important consequence of the convergence analysis, particularly of Theorem 4.4, is the modular nature of the proposed framework. Specifically, any method employed during face-exploring iterations that satisfies the following property can be incorporated into MAMBO: if $\{x^k\}$ is an infinite sequence of iterates generated on a fixed face $\mathcal{F}_A(x^k)$ by an internal unconstrained algorithm, then $\lim_{k\to\infty} v_S(x^k) = 0$, where v_S is defined in (15). This observation implies that advances in unconstrained multi-

objective optimization methods can be directly incorporated into MAMBO to improve its performance, without requiring modifications to its overall structure. The proposed algorithm is already flexible with respect to the choice of search directions during face-exploring iterations, as characterized by conditions (16)–(17). While a truncated Newton–gradient-type direction was used in the present implementation, alternative first or second-order strategies satisfying these conditions may be explored.

Several directions for future research naturally arise from this work. One promising extension is the development of an augmented Lagrangian-type method for multiobjective optimization with general constraints, using MAMBO as an internal solver for the resulting box-constrained subproblems. Such an approach would extend the applicability of the proposed framework beyond simple bound constraints. In addition, the numerical results highlight that, as a deterministic descent-type method, the diversity of the Pareto front approximations generated by MAMBO is inherently influenced by the choice of initial points, particularly for problems admitting a large number of Pareto critical or weakly Pareto-optimal solutions. This motivates the investigation of more effective initialization strategies. A particularly interesting direction is the combination of heuristic exploration mechanisms, such as those employed by NSGA-II, with subsequent refinement using MAMBO. Hybrid strategies of this type may offer an effective way to exploit the complementary strengths of deterministic descent-based methods and population-based evolutionary algorithms.

References

[1] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.

[2] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Comput. Optim. Appl.*, 23(1):101–125, 2002.

[3] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. SIAM, Philadelphia, 2014.

[4] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211, 2000.

[5] G. A. Carrizo, N. S. Fazzio, M. D. Sánchez, and M. L. Schuverdt. Scaled-pakkt sequential optimality condition for multiobjective problems and its application to an augmented lagrangian method. *Comput. Optim. Appl.*, 89(3):769–803, 2024.

[6] G. A. Carrizo, N. S. Fazzio, and M. L. Schuverdt. A nonmonotone projected gradient method for multiobjective problems on convex sets. *J. Oper. Res. Soc. China*, 12(2):410–427, 2024.

[7] J. Chen, L. Tang, and X. Yang. A Barzilai-Borwein descent method for multiobjective optimization problems. *Eur. J. Oper. Res.*, 311(1):196–209, 2023.

[8] G. Cocchi, G. Liuzzi, A. Papini, and M. Sciandrone. An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints. *Comput. Optim. Appl.*, 69(2):267–296, 2018.

[9] J.-A. M. de Dios and E. Mezura-Montes. Metaheuristics: A julia package for single- and multi-objective optimization. *J. Open Source Softw.*, 7(78):4723, 2022.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002.

[11] R. S. Dembo and U. Tulowitzki. *On the minimization of quadratic functions subject to box constraints*. Yale University, Department of Computer Science, 1984.

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

[13] N. Fazzio, L. F. Prudente, and M. L. Schuverdt. The projected gradient direction in vector optimization: continuity and dual characterization, 2025.

[14] N. S. Fazzio and M. L. Schuverdt. Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems. *Optim. Lett.*, 13(6):1365–1379, 2019.

[15] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM J. Optim.*, 20(2):602–626, 2009.

[16] E. H. Fukuda and L. M. Graña Drummond. On the convergence of the projected gradient method for vector optimization. *Optimization*, 60(8-9):1009–1021, 2011.

[17] M. L. N. Gonçalves, F. S. Lima, and L. F. Prudente. Globally convergent newton-type methods for multiobjective optimization. *Comput. Optim. Appl.*, 83(2):403–434, 2022.

[18] L. M. Graña Drummond and A. N. Iusem. A projected gradient method for vector optimization problems. *Comput. Optim. Appl.*, 28(1):5–29, 2004.

[19] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. Optim.*, 17(2):526–557, 2006.

[20] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE T. Evolut. Comput.*, 10(5):477–506, 2006.

[21] M. Lapucci, P. Mansueto, and F. Schoen. A memetic procedure for global multi-objective optimization. *Math. Program. Comput.*, 15(2):227–267, 2023.

[22] L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM J. Optim.*, 28(3):2690–2720, 2018.

[23] L. R. Lucambio Pérez and L. F. Prudente. A Wolfe line search algorithm for vector optimization. *ACM Trans. Math. Softw.*, 45(4):23, 2019.

[24] K. Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

[25] E. Miglierina, E. Molho, and M. Recchioni. Box-constrained multi-objective optimization: A gradient-like method without a priori scalarization. *Eur. J. Oper. Res.*, 188(3):662–682, 2008.

[26] V. Morovati, L. Pourkarimi, and H. Basirzadeh. Barzilai and Borwein's method for multiobjective optimization problems. *Numer. Algorithms*, 72(3):539–604, 2016.

[27] D. Orban and G. Leconte. RipQP.jl: Regularized interior point solver for quadratic problems. https://github.com/JuliaSmoothOptimizers/RipQP.jl, November 2020.

[28] A. Pascoletti and P. Serafini. Scalarizing vector optimization problems. *J. Optim. Theory Appl.*, 42(4):499–524, 1984.

[29] B. Polyak. The conjugate gradient method in extremal problems. *USSR Comput. Math. & Math. Phys.*, 9(4):94–112, 1969.

[30] L. F. Prudente and D. R. Souza. A quasi-Newton method with Wolfe line searches for multiobjective optimization. *J. Optim. Theory Appl*, 194(3):1107–1140, 2022.

[31] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.*, 7(1):26–33, 1997.

[32] M. C. Recchioni. A path following method for box-constrained multiobjective optimization with applications to goal programming problems. *Math. Methods Oper. Res.*, 58(1):69–85, 2003.

[33] A. Singh and M. A. T. Ansary. A hybrid approach for box constrained multiobjective optimization problems combining quasi-Newton and NSGA-II methods. *Authorea Preprints*, 2025.

[34] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. Di Stefano. Real-world applications of multiobjective optimization. In J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 285–327. Springer, Berlin, Heidelberg, 2008.