A linear programming framework and an improved backtracking strategy for multiple-gradient descent

Journal of Computational and Applied Mathematics 482 (2026) 117324

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A linear programming framework and an improved backtracking strategy for multiple-gradient descent

Francesco Della Santa [a,b,*]

[a] Department of Mathematical Sciences, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Turin, Italy
[b] Gruppo Nazionale per il Calcolo Scientifico INdAM, Piazzale Aldo Moro 5, 00185, Rome, Italy

ARTICLE INFO

2020 MSC:
90C29
65K05
90C26

Keywords:
Multiple-gradient descent
Multi-objective optimization
Linear programming

ABSTRACT

This work introduces a method to compute descent directions common to two or more differentiable functions defined over a shared unconstrained domain. Building on this, an alternative Multiple-Gradient Descent procedure for Multi-Objective Optimization problems is proposed. The core of the approach consists of solving a relatively cheap Linear Programming (LP) problem, where the objective and constraints are constructed from the gradients of the functions involved. In particular, the LP formulation is designed such that, when a common descent direction does not exist, it still yields a direction that is perpendicular to all objectives' gradients, if such a direction is available. Additionally, a tailored backtracking strategy is presented, enhancing the performance of Multiple-Gradient Descent methods, especially when paired with the proposed LP-based direction computation, by improving the exploration of the Pareto set and front. Theoretical analysis and experiments on standard benchmark problems are provided to evaluate the effectiveness of the proposed techniques.

1. Introduction

Multi-Objective Optimization (MOO) is the area in optimization that deals with problems that involve multiple, and often competing, objectives that must be optimized simultaneously. The importance of such a kind of optimization problems is evident from the wide MOO applications in real-world scenarios, ranging from mechanical engineering and fluid dynamics [1–3], to energy-saving strategies [4,5], to task allocation strategies [6], and many other topics. These applications have in common the need to find optimal trade-offs between different objectives, typically characterized by competing behaviors with respect to the optimization variables. The trade-off solutions typically are represented by the so-called Pareto set, while their images in the objectives' space by the so-called Pareto front. For more details about the theory of MOO problems, we refer to [7–9].

Often, MOO problems are solved using derivative-free methods, where the most used ones are the nature-inspired methods [10,11]. In particular, naturally inspired methods that were originally developed for single-objective, derivative-free optimization, such as Genetic Algorithms (GAs) [12] and Particle Swarm Optimization (PSO) algorithms [13,14], have, in recent years, been extended to solve MOO problems, becoming the most popular approaches for MOO [10]; see, e.g., [15–17] for GAs and [18] for PSO. The main advantages of these methods are their derivative-free nature and their efficiency in exploring the domain through "populations" of solutions; on the other hand, they lack "realistic" theoretical convergence properties (e.g., see [15]) or they have difficulties in finding accurate solutions, typically due to premature convergence (e.g., see [17,18]). Another common approach for solving MOO problems involves minimizing a single loss function formed by a weighted sum of the objective functions [7] or using specific scalarization

* Corresponding author.
E-mail address: francesco.dellasanta@polito.it

https://doi.org/10.1016/j.cam.2025.117324
Received 15 June 2025; Received in revised form 14 November 2025
Available online 30 December 2025

techniques (e.g., see [19,20]). These methods have stronger theoretical foundations but require multiple runs with varying parameter values to explore the Pareto set/front in depth.

A last approach for MOO problems is represented by Multiple-Gradient Descent (MGD) methods [21–26]. MGD methods aim to build a sequence that converges to the Pareto set by solving sub-problems that determine descent directions shared by all the objective functions. These methods are characterized by robust theoretical properties; nonetheless, similarly to the scalarization techniques, they must be executed multiple times to fully explore the Pareto set/front, starting from different points sampled from the domain. The sub-problems used in MGD methods for computing the shared descent directions are often nonlinear; for example, in [22–26] the sub-problem consists in finding the minimum-norm vector in the anti-gradients' convex hull. Nonetheless, there are some approaches based on Linear Programming (LP) problems, as illustrated in [21]. Even if the methods for solving LP problems are fast and very efficient, often the nonlinear sub-problems are preferred, probably due to the different properties of the directions they return. For example, the anti-gradients' convex hull used in [22–25] restricts the search of the descent directions to a sub-region of all the shared descent directions that, unless of particular cases, is sufficiently distant to the boundary defined by the set of perpendicular directions to one of the gradients; on the other hand, the LP problem described in [21] look for the direction in the whole region of shared descent directions, therefore the solution can be almost perpendicular to some of the objective gradients. More details and theoretical properties of this LP problem will be given in this work since it is used as a baseline for the new MGD method proposed.

In this work, the author proposes a new LP sub-problem that is a trade-off between the cheap LP sub-problem defined in [21] and the nonlinear sub-problems with solutions characterized by "steepness properties". Specifically, we introduce a new LP problem for computing directions in MGD methods such that its solution is a direction that tries to maximize the distance from the boundaries of the descent directions' region and tries to follow as much as possible the direction identified by the sum of the anti-gradients. For doing so, we start from the LP problem described in [21] and we modify its objective function and its constraints to achieve these properties. A theoretical analysis for characterizing the solutions of the new LP problem is performed. In particular, we prove that the new LP formulation admits the null direction as a solution for Pareto critical points only in the following two situations: *i*) all the directions with non-positive dot product with respect to objectives' gradients that are shared by all the objective functions are perpendicular to all the gradients; *ii*) the only direction with non-positive dot product with respect to objectives' gradients that is shared by all the objectives is the null vector. In a third situation, where there is at least one direction of this kind, and it is also a descent direction (i.e., not null) for at least one objective, the LP problem returns one of these vectors and not the null vector. The latter characteristic (not present in the baseline LP problem) is an extra property that the author deliberately incorporates in the new LP problem to enhance its interaction with a new backtracking strategy for MGD methods, which is also proposed in this work.

To the best of the author's knowledge, the backtracking strategies used for MGD methods are always focused on building a sequence of vectors $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ that is strictly decreasing for all the objective functions (e.g., see all the MGD methods cited above). These backtracking strategies can lead the sequence to very good (approximated) Pareto optimal solutions or can stop it "prematurely", i.e., as soon as they do not find a sufficiently small step to decrease all the objectives along the chosen direction. In order to avoid this latter case and improve the possibility of building a sequence that converges toward a Pareto optimal solution, we develop a new backtracking strategy. This new strategy is designed to accept the new point $\boldsymbol{x}^{(k+1)}$ if it is non-dominated by the previous point $\boldsymbol{x}^{(k)}$ when the Armijo condition is not satisfied for all objectives for all the backtracking steps. Additionally, we endow the backtracking strategy with a "storing property"; i.e., in a sequence, we keep apart all the vectors $\boldsymbol{x}^{(k)}$ that do not dominate $\boldsymbol{x}^{(k+1)}$ and that are not dominated by it, since this vectors are candidate Pareto optimals, and not only the last one. A theoretical analysis of MGD methods based on this new backtracking strategy is performed and convergence properties are established.

Through numerical experiments, we analyze the performance of our new methods, both the new LP problem and the new backtracking strategy, and we compare them with the baseline given by the literature in [21]. The experiment results indicate that our backtracking strategy consistently offers only advantages with respect to the "strictly decreasing" ones. While the new LP problem performs well, its benefits are most evident when combined with the new backtracking strategy, outperforming the baseline when applied to MOO problems characterized by large regions of Pareto critical points.

The work is organized as follows. In the next section (Section 2), the main notations and definitions used in MOO are listed. In Section 3, the new LP sub-problem for shared descent directions is presented, whereas in Section 3.2 the theoretical characterization of the sub-problem solutions is described. In Section 4 the new backtracking strategy is introduced and both the theoretical convergence analyses and its implementation pseudo-codes are reported. Section 5 illustrates numerical results where the proposed methods are compared with a baseline on some typical MOO test problems taken from the literature; in particular, the results assess the potential of the new LP sub-problem and the new backtracking strategy. We end the paper with some conclusions drawn in Section 6.

2. Multi-objective optimization setting and notations

In this work, we consider the case of an unconstrained Multi-Objective Optimization (MOO) problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} (f_1(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})), \tag{1}$$

with m differentiable objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$, for each $i = 1, \dots, m$.

In this section, for the reader's convenience, we recall the main definitions related to MOO problems like (1) and Pareto optimality. For a more detailed introduction about MOO, for example, see [7–9].

Notation 2.1 (Order relations and vectors)**.** Let $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^n$. Then, we denote by $\boldsymbol{v} < \boldsymbol{w}$ the relation between $\boldsymbol{v}$ and $\boldsymbol{w}$ such that $v_i < w_i$, for each $i = 1, \dots, n$ (analogous for $\leq$, $>$, and $\geq$). We denote by $\boldsymbol{v} \not< \boldsymbol{w}$ the opposite of the relation $\boldsymbol{v} < \boldsymbol{w}$ (analogous for $\not\leq$, $\not>$, and $\not\geq$); i.e., $\boldsymbol{v} \not< \boldsymbol{w}$ if there is $i \in \{1, \dots, n\}$ such that $v_i \geq w_i$.

Definition 2.1. Let $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ be the function such that $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_m(\boldsymbol{x}))$, where $f_1, \dots, f_m$ are the m objective functions of (1). Then, with respect to (1), we have that:

1. $\boldsymbol{x}_1 \in \mathbb{R}^n$ *dominates* $\boldsymbol{x}_0 \in \mathbb{R}^n$ if $\boldsymbol{f}(\boldsymbol{x}_1) \leq \boldsymbol{f}(\boldsymbol{x}_0)$ and $\boldsymbol{f}(\boldsymbol{x}_1) \neq \boldsymbol{f}(\boldsymbol{x}_0)$. Alternatively, we can say that $\boldsymbol{x}_0$ *is dominated by* $\boldsymbol{x}_1$.
2. $\boldsymbol{x}_0 \in \mathbb{R}^n$ is *non-dominated* by $\boldsymbol{x}_1 \in \mathbb{R}^n$ if $\boldsymbol{x}_1$ does not dominate $\boldsymbol{x}_0$; i.e., $\boldsymbol{f}(\boldsymbol{x}_1) \not\leq \boldsymbol{f}(\boldsymbol{x}_0)$
3. $\boldsymbol{x}^* \in \mathbb{R}^n$ is a *local Pareto optimal* for (1) if there is $\varepsilon > 0$ such that $\boldsymbol{x}^*$ is non-dominated by $\boldsymbol{x}$, for each $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leq \varepsilon$.
4. $\boldsymbol{x}^* \in \mathbb{R}^n$ is a *global Pareto optimal* for (1) if $\boldsymbol{x}^*$ is non-dominated by $\boldsymbol{x}$, for each $\boldsymbol{x} \in \mathbb{R}^n$.
5. The set of all and only the global Pareto optimals is defined as *Pareto set* of (1); its image through $\boldsymbol{f}$ is defined as *Pareto front* of (1).
6. $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ is defined as *Pareto critical point* for (1) if descent directions for all the objective functions $f_1, \dots, f_m$ evaluated in $\hat{\boldsymbol{x}}$ do not exist; i.e, if

$$\boldsymbol{J}_{\boldsymbol{f}}(\hat{\boldsymbol{x}})\boldsymbol{v} \not< \boldsymbol{0}, \tag{2}$$

for each $\boldsymbol{v} \in \mathbb{R}^n$, where $\boldsymbol{J}_{\boldsymbol{f}}$ denotes the Jacobian of $\boldsymbol{f}$. By consequence, $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ is not a Pareto critical point for (1) if a descent direction for all the objective functions $f_1, \dots, f_m$ evaluated in $\hat{\boldsymbol{x}}$ exists; i.e, if $\boldsymbol{p} \in \mathbb{R}^n$ exists such that

$$\boldsymbol{J}_{\boldsymbol{f}}(\hat{\boldsymbol{x}})\boldsymbol{p} < \boldsymbol{0}. \tag{3}$$

Remark 2.1 (Necessary Condition for Local Pareto Optimals)**.** Being a Pareto critical is a necessary condition for being a local Pareto optimal. Specifically, if $\boldsymbol{x}^*$ is a local Pareto optimal, then $\boldsymbol{x}^*$ is a Pareto critical.

After recalling the main entities related to MOO, we introduce the definition of *non-positive dot product regions* for the objective functions. This definition will be useful for studying and analyzing the properties of the proposed MGD method (see Section 3).

Definition 2.2. [Non-positive Dot Product Regions] Let $f_1, \dots, f_m$ be the m objective functions of (1). Let $\boldsymbol{x} \in \mathbb{R}^n$ be fixed. Then, for each $i = 1, \dots, m$, we define *non-positive dot product region* of f_i evaluated at $\boldsymbol{x}$ the set

$$P_i(\boldsymbol{x}) := \{\boldsymbol{p} \in \mathbb{R}^n \mid \nabla f_i(\boldsymbol{x})^T \boldsymbol{p} \leq 0\}; \tag{4}$$

i.e., $P_i(\boldsymbol{x})$ is the union of the descent directions of f_i at $\boldsymbol{x}$ and the directions perpendicular to $\nabla f_i(\boldsymbol{x})$. Moreover, we define as *region of shared non-positive dot product directions* the intersection of all the non-positive dot product regions, i.e., the set $P(\boldsymbol{x}) := \bigcap_{i=1}^m P_i(\boldsymbol{x})$ (it always contains the null vector).

2.1. Descent methods for multi-objective optimization

Descent methods for MOO problems are iterative methods such that

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathbb{R}^n \text{ given} \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \eta^{(k)} \boldsymbol{p}^{(k)}, \ \forall\, k \geq 0 \end{cases}, \tag{5}$$

where $\boldsymbol{p}^{(k)} \in \mathbb{R}^n$ is a descent direction for all the objective functions evaluated at $\boldsymbol{x}^{(k)}$, and $\eta^{(k)} \in \mathbb{R}_+$ is the step-length of the descent method at iteration k.

In literature, one of the favorite approaches used for the computation of a descent direction $\boldsymbol{p}^*$ for m functions $f_1, \dots, f_m$ evaluated at a point $\boldsymbol{x}$ is the one based on finding the minimum-norm vector in the anti-gradients' convex hull [22–26], i.e.:

$$\boldsymbol{p}^* = -\sum_{i=1}^m \alpha_i^* \nabla f_i(\boldsymbol{x}), \tag{6}$$

where

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \left\{ \left\| \sum_{i=1}^m \alpha_i \nabla f_i(\boldsymbol{x}) \right\|^2 \text{ s.t. } \alpha_1, \dots, \alpha_m \geq 0, \ \sum_{i=1}^m \alpha_i = 1 \right\}. \tag{7}$$

Specifically, problem (7) is a quadratic optimization problem and its solutions $\boldsymbol{\alpha}^*$ guarantee that the vector $\boldsymbol{p}^*$ defined by Eq. (6) is either a descent direction for the m objective functions, if such a direction exists, or the null vector, if $\boldsymbol{x}$ is Pareto critical; e.g., see Theorem 2.1 in [22] for a theoretical proof, or see Fig. 1 for a visual example.

On the other hand, there are methods simpler than (7) for finding a descent direction $\boldsymbol{p}^*$; for example, the method described in [21] and based on solving the Linear Programming (LP) problem

$$\begin{cases} \min_{(\boldsymbol{p},\beta)\in\mathbb{R}^{n+1}} \beta \\ \nabla f_i(\boldsymbol{x})^T \boldsymbol{p} \leq \beta, \quad \forall i = 1, \dots, m \\ \|\boldsymbol{p}\|_\infty \leq 1 \end{cases}. \qquad \text{LP}_{\text{base}}$$

Actually, the LP problem above involves $n+1$ variables: the n variables stored in the vector $\boldsymbol{p} \in \mathbb{R}^n$ and the scalar variable β; therefore, by introducing the variable $\boldsymbol{\rho} := (\boldsymbol{p}, \beta) \in \mathbb{R}^{n+1}$, we can rewrite LP_{base} as the problem

$$\begin{cases} \min_{\boldsymbol{\rho}\in\mathbb{R}^{n+1}} [0, \dots, 0, 1]\, \boldsymbol{\rho} \\ \left[\begin{array}{c|c} \nabla f_1(\boldsymbol{x})^T & -1 \\ \vdots & \vdots \\ \nabla f_m(\boldsymbol{x})^T & -1 \end{array}\right] \boldsymbol{\rho} \leq \boldsymbol{0} \\ -1 \leq \rho_i \leq 1, \ \forall i = 1, \dots, n \end{cases}, \qquad \text{LP}'_{\text{base}}$$

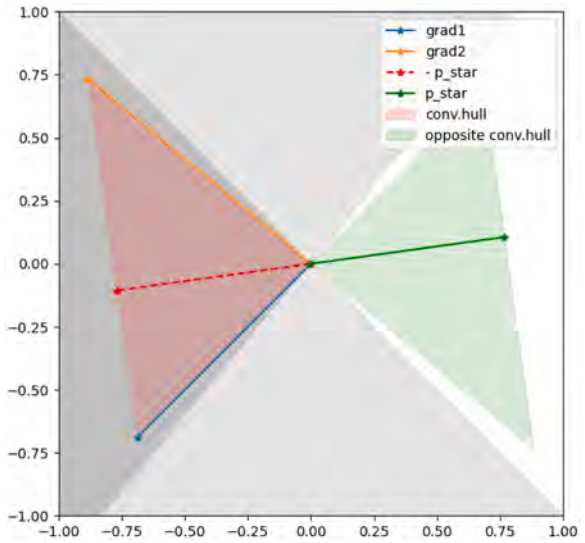


Fig. 1. Example of the direction $\boldsymbol{p}^*$ (green vector) obtained from (6) and (7). Grey color denotes regions of ascent directions for the objective functions; white color denotes the shared regions of descent directions for the objective functions. The red region is the convex hull of the two gradients considered; the green region denotes the convex hull of the anti-gradients.

or more explicitly as

$$\begin{cases} \min_{(\boldsymbol{p},\beta)\in\mathbb{R}^{n+1}} [0,\dots,0,1] \begin{bmatrix} \boldsymbol{p} \\ \beta \end{bmatrix} \\ \left[\begin{array}{c|c} \nabla f_1(\boldsymbol{x})^T & -1 \\ \vdots & \vdots \\ \nabla f_m(\boldsymbol{x})^T & -1 \end{array}\right] \begin{bmatrix} \boldsymbol{p} \\ \beta \end{bmatrix} \leq \boldsymbol{0} \\ -1 \leq p_i \leq 1, \ \forall i = 1,\dots,n \end{cases}$$

Then, the solution $\boldsymbol{\rho}^* = (\boldsymbol{p}^*, \beta^*) \in \mathbb{R}^{n+1}$ of LP'_{base} is the concatenation of the shared descent direction found $\boldsymbol{p}^*$ and the optimal value $\beta^* \leq 0$ found for β; where we have that $\beta^* = 0$, if $\boldsymbol{x}$ is Pareto critical, and that $\beta^* < 0$, otherwise (see Lemma 3.1).

3. A new method based on linear programming

In this paper, one of our focuses is the development of a new LP approach for finding a descent direction shared by multiple functions. In particular, we start from the LP method described in [21] (see LP_{base} and LP'_{base}), and we modify it. More precisely, analogously to [21], the LP we define finds descent directions if they exist; otherwise, directions with non-positive dot product with respect to the objective functions' gradients are returned (see Definition 2.2). Nonetheless, these latter directions are positively used by our MGD method, contrary to other methods in literature (see Section 1); indeed, in this work, we also define a novel and different backtracking strategy for the optimization procedure (5) that is able to exploit them. In this section we focus on the new LP approach, postponing the definition of the new backtracking strategy in Section 4.

First of all, solving LP_{base}, we observe that we obtain a shared descent direction $\boldsymbol{p}^*$ (assuming it exists) such that its scalar product with each gradient is less than the minimum value β^* (always less than or equal to zero). It is worth noting that, in this formulation, the gradients of the m objective functions affect only the inequality constraints that define the feasible region of the LP problem, but not the objective function itself. Consequently, the magnitudes of the gradients have only a limited influence on determining $\boldsymbol{p}^*$; rather, it is mainly their directions that define whether a feasible shared descent direction exists. In other words, problem LP_{base} ensures the computation of a direction $\boldsymbol{p}^*$ that satisfies the descent condition for all objectives, but it does not ensure that a step taken along $\boldsymbol{p}^*$ will decrease all objective functions coherently with the steepness indicated by their gradients or by the average of all the gradients.

In addition, problem LP_{base} has good probabilities of returning the null vector as the solution (stopping the optimization procedure) also in the special case of Pareto critical points characterized by the presence of at least one direction that is a descent direction for at least one objective and perpendicular to other objectives' gradients at most. Nonetheless, such a kind of non-null direction could generate a new step of (5) where $\boldsymbol{x}^{(k+1)}$ is non-dominated by $\boldsymbol{x}^{(k)}$; therefore, under these circumstances, it can be useful to add this option to a MGD optimization procedure, removing the null vector from the solution set of the LP problem.

Given the observations above, the purposes of the proposed new LP problem are the following:

1. Modify problem LP_{base} such that, moving along the direction of $\boldsymbol{p}^*$, all the objective functions decrease as much coherently as possible with their gradient's characteristics;

2. Avoid returning the null vector as a solution for Pareto critical points, if there is at least one direction that is a descent direction for at least one objective and perpendicular to other objectives' gradients at most. We recall that we are interested in these directions too because our new backtracking strategy will exploit them (see Section 4).

Before starting with the formulation of the new LP problem, we introduce some notations. We denote by $\boldsymbol{g}_i(\boldsymbol{x})$ the gradient $\nabla f_i(\boldsymbol{x})$, for each $i = 1, \dots, m$, and by $\boldsymbol{g}(\boldsymbol{x})$ their sum, i.e.:

$$\boldsymbol{g}(\boldsymbol{x}) := \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{x}) . \tag{8}$$

In addition, we denote by $\bar{\boldsymbol{g}}_i(\boldsymbol{x})$ the gradient $\boldsymbol{g}_i$ normalized with respect to the euclidean norm; i.e.,

$$\bar{\boldsymbol{g}}_i(\boldsymbol{x}) := \frac{\boldsymbol{g}_i(\boldsymbol{x})}{\|\boldsymbol{g}_i(\boldsymbol{x})\|_2}, \tag{9}$$

for each $i = 1, \dots, m$. Then, we denote by $G(\boldsymbol{x})$ and $\bar{G}(\boldsymbol{x})$ the matrices with rows defined by the gradients (i.e., the Jacobian of $\boldsymbol{f}$, see Definition 2.1) and the normalized gradients, respectively; specifically:

$$G(\boldsymbol{x}) := \begin{bmatrix} \boldsymbol{g}_1(\boldsymbol{x})^T \\ \vdots \\ \boldsymbol{g}_m(\boldsymbol{x})^T \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad \bar{G}(\boldsymbol{x}) := \begin{bmatrix} \bar{\boldsymbol{g}}_1(\boldsymbol{x})^T \\ \vdots \\ \bar{\boldsymbol{g}}_m(\boldsymbol{x})^T \end{bmatrix} \in \mathbb{R}^{m \times n} . \tag{10}$$

Given (10), the inequality constraints of LP'_{base} defined using the gradients can be rewritten as

$$\left[G(\boldsymbol{x}) \;\middle|\; -\boldsymbol{e} \right] \boldsymbol{\rho} \leq \boldsymbol{0}, \tag{11}$$

where $\boldsymbol{e} := (1, \dots, 1) \in \mathbb{R}^m$.

3.1. Linear programming problem formalization

For improving the influence of the gradient magnitudes according to the purpose illustrated in item 1, we need to involve the gradients both in the objective function and in the boundary values of the variables corresponding to the descent direction. Specifically, we change the objective function of the problem LP_{base} into

$$\boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{p} + c_\beta(\boldsymbol{x})\beta \tag{12}$$

and the boundary values into

$$\|\boldsymbol{p}\|_\infty \leq \gamma(\boldsymbol{x}), \quad \beta \leq 0, \tag{13}$$

where $c_\beta(\boldsymbol{x}) > 0$ (in practice, $c_\beta(\boldsymbol{x}) := \|\boldsymbol{g}(\boldsymbol{x})\|_2 + \varepsilon$, $\varepsilon > 0$, see Section 3.2 later) and

$$\gamma(\boldsymbol{x}) := \max\{ \|\boldsymbol{g}_1(\boldsymbol{x})\|_\infty, \dots, \|\boldsymbol{g}_m(\boldsymbol{x})\|_\infty, \|\boldsymbol{g}(\boldsymbol{x})\|_\infty \} . \tag{14}$$

Clearly, for each $\boldsymbol{p}$ satisfying (11) and $\beta \leq 0$, it holds that $\boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{p} + c_\beta(\boldsymbol{x})\beta \leq m\beta + c_\beta(\boldsymbol{x})\beta \leq 0$.

Using the same notation of problem LP'_{base}, the objective function is now

$$[\boldsymbol{g}(\boldsymbol{x})^T \mid c_\beta(\boldsymbol{x})] \, \boldsymbol{\rho}, \tag{15}$$

while the boundary values are

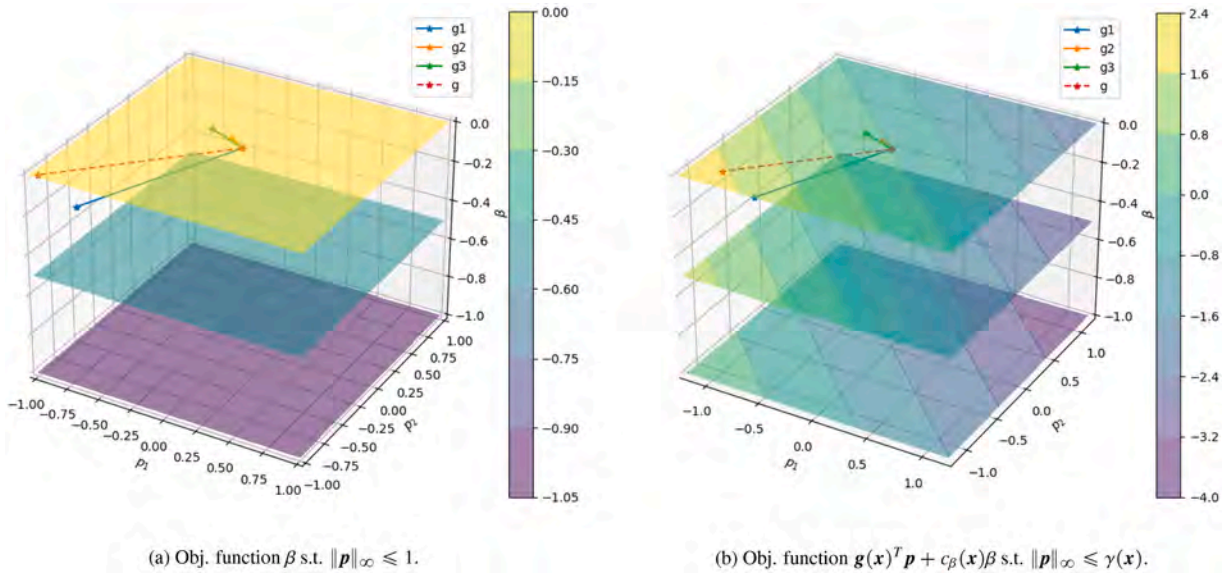
$$-\gamma(\boldsymbol{x}) \leq \rho_i \leq \gamma(\boldsymbol{x}), \; \forall i = 1, \dots, n \quad \text{and} \quad \rho_{n+1} \leq 0 . \tag{16}$$

In Fig. 2(a) we illustrate the proposed objective function (15) with respect to the boundaries (16), in comparison with the corresponding objective function and boundaries of problem LP_{base}, illustrated in Fig. 2(a). Looking at these figures, we see that the objective function for LP_{base} is constant for each fixed value of β; on the other hand, the objective function (15) is not constant for each fixed value of β, because it is decreasing along the direction of the vector $-\boldsymbol{g}(\boldsymbol{x})$. Therefore, the change of objective function is proposed because it forces the solution to find a descent direction that minimizes β but also follows as much as possible the direction identified by the sum of the anti-gradients. On the other hand, the boundary values of the descent directions have been changed in order to maintain at least a weak connection with the norms of the function gradients for a better embedding with an iterative method like (5); this connection is not preserved if we use the condition $\|\boldsymbol{p}\|_\infty \leq 1$ like in LP_{base}.

After the modification of the objective function and the boundary values, we observe that also the inequality constraints (11) can be modified to improve the quality of the descent direction $\boldsymbol{p}^*$.

Ideally, the more parallel a descent direction is to the anti-gradient, the better; then, in the region of the descent directions, the more distant it is from the hyperplane perpendicular to the gradient, the better. Actually, the inequality constraints (11) are implicitly trying to apply the latter reasoning because the distances of $\boldsymbol{p}$ from each one of the hyperplanes perpendicular to the m gradients are lower bounded by a quantity dependent on β. Specifically, denoted by $H_i(\boldsymbol{x})$ the hyperplane perpendicular to $\boldsymbol{g}_i(\boldsymbol{x})$, if $\boldsymbol{\rho} = (\boldsymbol{p}, \beta)$ satisfies (11), then the distance of $\boldsymbol{p}$ from $H_i(\boldsymbol{x})$ is lower bounded by $|\beta| / \|\boldsymbol{g}_i(\boldsymbol{x})\|_2$, i.e.:

$$\text{dist}\big(H_i(\boldsymbol{x}), \boldsymbol{p}\big) = \frac{|\boldsymbol{g}_i(\boldsymbol{x})^T \boldsymbol{p}|}{\|\boldsymbol{g}_i(\boldsymbol{x})\|_2} \geq \frac{|\beta|}{\|\boldsymbol{g}_i(\boldsymbol{x})\|_2}, \quad \forall \, i = 1, \dots, m . \tag{17}$$



(a) Obj. function β s.t. $\|\boldsymbol{p}\|_\infty \leqslant 1$. (b) Obj. function $\boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{p} + c_\beta(\boldsymbol{x})\beta$ s.t. $\|\boldsymbol{p}\|_\infty \leqslant \gamma(\boldsymbol{x})$.

Fig. 2. Visual comparison of the objective function of problem LP_{base} (sub-figure a) and dummyTXdummy– the proposed objective function (15) (sub-figure b). The figures illustrate three function evaluations in $\mathbb{R}^3$ for three fixed values of $\beta = -1, -0.5, 0$, three randomly generated gradients $\boldsymbol{g}_1(\boldsymbol{x}), \boldsymbol{g}_2(\boldsymbol{x}), \boldsymbol{g}_3(\boldsymbol{x}) \in \mathbb{R}^2$, and (for sub-figure b) $c_\beta(\boldsymbol{x}) = \|\boldsymbol{g}(\boldsymbol{x})\|_2 + 0.25$.

Therefore, looking at (17), we observe that a solution $\boldsymbol{p}^*$ can result to be nearly-perpendicular to those gradients characterized by a large norm, because the distance's lower bound tends toward zero. This phenomenon can be a problem for an iterative method like (5), because the step can be almost-perpendicular to the steepest descent direction of one of the steepest objective functions of the problem (see the lightly obscured region varying with β and the vectors $\boldsymbol{\rho}^*, \boldsymbol{p}^*$ in Fig. 3(a)).

Then, we modify the inequality constraints (11) using the matrix $\bar{G}(\boldsymbol{x})$ instead of $G(\boldsymbol{x})$; i.e., we use the inequality constraints

$$\left[\bar{G}(\boldsymbol{x}) \;\middle|\; -\boldsymbol{e}\right]\boldsymbol{\rho} \leq \boldsymbol{0}\,. \tag{18}$$

For a point $\boldsymbol{x}$ that is not Pareto critical, if (18) is used as inequality constraint, we can prove (see Lemma 3.2) that the minimum distance of a solution $\boldsymbol{p}^*$ from all the hyperplanes $H_i(\boldsymbol{x})$ is at least $|\beta^*|$ (see the lightly obscured region varying with β and the vectors $\boldsymbol{\rho}^*, \boldsymbol{p}^*$ in Fig. 3(b)).

In conclusion, applying to LP'_{base} the changes listed above, we obtain the following new LP problem:

$$\begin{cases} \min_{\boldsymbol{\rho}\in\mathbb{R}^{n+1}} [\; \boldsymbol{g}(\boldsymbol{x})^T \mid c_\beta(\boldsymbol{x}) \;]\,\boldsymbol{\rho} \\ \left[\bar{G}(\boldsymbol{x}) \;\middle|\; -\boldsymbol{e}\right]\boldsymbol{\rho} \leq \boldsymbol{0} \\ -\gamma(\boldsymbol{x}) \leq \rho_i \leq \gamma(\boldsymbol{x}),\ \forall i = 1, \dots, n \\ \rho_{n+1} \leq 0 \end{cases} . \qquad LP_{\text{new}}$$

In Fig. 3 we illustrate an example of different solutions obtained by applying LP'_{base} and LP_{new} to the same set of gradients. In particular, in Fig. 3(a), we observe that the direction $\boldsymbol{p}^*$ obtained from the solution $\boldsymbol{\rho}^*$ of LP_{base} is almost perpendicular to $\boldsymbol{g}_1(\boldsymbol{x})$, the gradient with the largest norm, and consequently also to the sum-of-gradients vector $\boldsymbol{g}(\boldsymbol{x})$. This behavior reflects the observations made for the lower bound (17) in LP_{base}, where the lower bound on the distance of a feasible direction $\boldsymbol{p}$ from the hyperplanes orthogonal to the gradients is directly proportional to $|\beta|$ and inversely proportional to the gradient norms (see the lightly shaded regions in Fig. 3(a)). Conversely, in Fig. 3(b), we can see that the direction $\boldsymbol{p}^*$ obtained from the solution $\boldsymbol{\rho}^*$ of LP_{new} is almost parallel to $-\boldsymbol{g}(\boldsymbol{x})$, and the corresponding lower bound on the distance of feasible directions $\boldsymbol{p}$ from the gradient hyperplanes is directly proportional to $|\beta|$ but independent of the gradient norms (see the lightly shaded regions in Fig. 3(b)).

3.2. Theoretical results and characterization

Analogously to Lemma 3 in [21] for problem LP_{base}, we illustrate some theoretical results for the LP problem LP_{new}. We report the main results of the Lemma for problem LP_{base} in the following, adapted to the notation used in this work.

Lemma 3.1 (Lemma 3 - [21]). *Let V^*, β^* be the solution set and the optimum value of problem LP_{base}, respectively. Then:*

1. *if $\boldsymbol{x}$ is Pareto critical for the functions $f_1, \dots, f_m$, then $\boldsymbol{0} \in V^*$ and $\beta^* = 0$;*
2. *if $\boldsymbol{x}$ is not Pareto critical for the functions $f_1, \dots, f_m$, then $\beta^* < 0$ and any $\boldsymbol{p}^* \in V^*$ is a descent direction for all the functions $f_1, \dots, f_m$;*

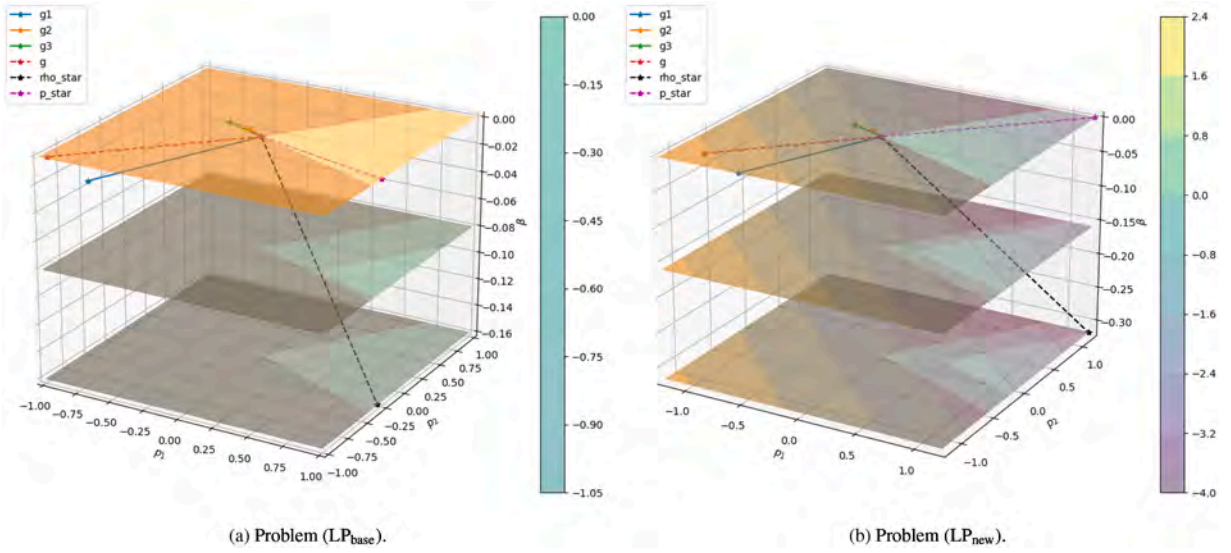


Fig. 3. Visual comparison of the solutions ρ^* and p^* obtained from problem LP_{base} (sub-figure a) and dummyTXdummy– problem LP_{new} (sub-figure b), respectively, given the same gradients $g_1(x), g_2(x), g_3(x) \in \mathbb{R}^2$, and (for sub-figure b) $c_\beta(x) = \|g(x)\|_2 + 0.25$. The figures illustrate three objective function evaluations in $\mathbb{R}^3$ for three fixed values of $\beta = \beta^*, \beta^*/2, 0$. The non-feasible region of each problem is obscured and changes with β (heavily obscured for the region of non-descent directions).

For the theoretical analysis of our problem, we start with a proposition that characterizes the influence of the parameter c_β on the objective function evaluation. In the next results, we will show how this proposition is necessary to guarantee $\beta^* < 0$ for non-null solutions of LP_{new}.

Notation 3.1. For ease of notation, from now on we will drop the dependency on x in the LP problems.

Proposition 3.1. *Let $\rho_0 = (p_0, \beta_0) \in \mathbb{R}^{n+1}$. Let c denotes the vector characterizing the objective function (15); i.e., $c^T := [g^T | c_\beta]$. Then, if $c_\beta > \|g\|_2$, we have that*

$$c^T \rho_0 - c^T \rho_1 > 0 \tag{19}$$

for each $\rho_1 = (p_1, \beta_1)$ such that $\beta_1 < \beta_0$ and $\text{dist}(p_0, p_1) \leq (\beta_0 - \beta_1)$.

Proof. Let α be the angle between g and $(p_0 - p_1)$. Let $\Delta\beta$ be the difference between β_0 and β_1; i.e., $\Delta\beta := \beta_0 - \beta_1 > 0$. Then,

$$\begin{aligned} c^T \rho_0 - c^T \rho_1 &= g^T (p_0 - p_1) + c_\beta \Delta\beta = \\ &= \|g\|_2 \|p_0 - p_1\|_2 \cos\alpha + c_\beta \Delta\beta \,. \end{aligned} \tag{20}$$

If $\cos\alpha \geq 0$, it holds

$$\begin{aligned} c^T \rho_0 - c^T \rho_1 &= \|g\|_2 \|p_0 - p_1\|_2 \cos\alpha + c_\beta \Delta\beta \geq \\ &\geq c_\beta \Delta\beta > \|g\|_2 \Delta\beta \,; \end{aligned} \tag{21}$$

otherwise, if $\cos\alpha < 0$, it holds

$$\begin{aligned} c^T \rho_0 - c^T \rho_1 &= \|g\|_2 \|p_0 - p_1\|_2 \cos\alpha + c_\beta \Delta\beta \geq \\ &\geq -\|g\|_2 \|p_0 - p_1\|_2 + c_\beta \Delta\beta > \\ &> \|g\|_2 (\Delta\beta - \|p_0 - p_1\|_2) \geq \|g\|_2 \,. \end{aligned} \tag{22}$$

In both cases, $\cos\alpha < 0$ or $\cos\alpha \geq 0$, the thesis (19) holds. □

In the following Lemma, we characterize the solutions obtained solving LP_{new}, depending on the Pareto nature of $x \in \mathbb{R}^n$; i.e., depending on x that is a Pareto critical point or not, following the lines of Lemma 3.1.

Lemma 3.2. *Let $\rho^* = (p^*, \beta^*) \in \mathbb{R}^{n+1}$ be a solution of problem LP_{new}, with $c_\beta > \|g\|_2$. Then:*

1. *If x is Pareto critical for the functions $f_1, \dots, f_m$, then $\beta^* = 0$.*
2. *Let x be Pareto critical for the functions $f_1, \dots, f_m$. Let V^* be the solution set, let P_i be the set of non-positive dot product directions of f_i at x, and let P be the intersection of all the sets P_i (see Definition 2.2). Let us denote by Π the set of all the feasible directions in P, according to the constraints of the problem. Then, it holds:*
   2.1 $\mathbf{0} \notin V^*$ if and only if there is $p \in \Pi$ such that $g^T p \neq 0$;

2.2 $V^* = \Pi$ *if and only if* $\boldsymbol{g}^T\boldsymbol{p} = 0$ *for each* $\boldsymbol{p} \in \Pi$
2.3 $V^* = \{\boldsymbol{0}\}$ *if and only if* $\Pi = \{\boldsymbol{0}\}$*;*
3. *If* $\boldsymbol{x}$ *is not Pareto critical for the functions* $f_1, \dots, f_m$*, then* $\beta^* < 0$ *and* $\boldsymbol{p}^*$ *is a descent direction for all the functions* $f_1, \dots, f_m$ *in* $\boldsymbol{x}$*.*
4. *If* $\boldsymbol{x}$ *is not Pareto critical for the functions* $f_1, \dots, f_m$*, then:*
4.1 $\beta^* = \max_{i=1,\dots,m} \tilde{\boldsymbol{g}}_i^T \boldsymbol{p}^*$*;*
4.2 For each $i = 1, \dots, m$*, it holds*

$$\operatorname{dist}\left(H_i, \boldsymbol{p}^*\right) \geq |\beta^*|\,; \tag{23}$$

Proof. In the following, we prove one by one the results listed in the Lemma.

1. The proof is almost straightforward. If $\boldsymbol{x}$ is Pareto critical, no negative values of β permit to satisfy the inequality constraints (see (2)); then, $\beta^* = 0$.
2. If $\boldsymbol{x}$ is Pareto critical, no descent directions are shared by the m objective functions. Then, solutions are contained in the non-positive dot product region; i.e., $V^* \subseteq \Pi$.

   Let us start with item 2.1. We use the proof by contradiction. Let the null vector be a solution and let $\boldsymbol{p} \in \Pi$ be such that $\boldsymbol{g}^T\boldsymbol{p} \neq 0$. Then, $\boldsymbol{g}_i^T\boldsymbol{p} \leq 0$, for each $i = 1, \dots, m$, and there is $j \in \{1, \dots, m\}$ such that $\boldsymbol{g}_j^T\boldsymbol{p} < 0$. By consequence, $\boldsymbol{g}^T\boldsymbol{p} < 0 = \boldsymbol{g}^T\boldsymbol{0}$. Therefore, $\boldsymbol{0}$ is not a solution of problem $\mathrm{LP}_{\mathrm{new}}$ because $(\boldsymbol{p}, 0)$ is a feasible vector with negative objective function value, which is a contradiction; i.e., $\boldsymbol{0}$ is not a solution if there is $\boldsymbol{p} \in \Pi$ such that $\boldsymbol{g}^T\boldsymbol{p} \neq 0$. On the other hand, let us assume that $\boldsymbol{0} \notin V^*$ and, again by contradiction, that there is no $\boldsymbol{p} \in \Pi$ such that $\boldsymbol{g}^T\boldsymbol{p} \neq 0$; therefore, it holds $\boldsymbol{g}^T\boldsymbol{p} = 0$ for each $\boldsymbol{p} \in \Pi$ and, for item 1, any solution $(\boldsymbol{p}^*, \beta^*)$ is such that $\beta^* = 0$. As a consequence, the minimum objective function value can be only 0 and $\boldsymbol{0} \in V^*$, which is a contradiction. Therefore, there is at least one $\boldsymbol{p} \in \Pi$ such that $\boldsymbol{g}^T\boldsymbol{p} \neq 0$ if $\boldsymbol{0} \notin V^*$.

   We now continue with item 2.2. Let us assume that $\boldsymbol{g}^T\boldsymbol{p} = 0$, for each $\boldsymbol{p} \in \Pi$. Since $V^* \subseteq \Pi$, we have that $\boldsymbol{g}^T\boldsymbol{p}^* = 0$ for each solution of the problem. But any vector in Π satisfies the constraints and has the same minimum value for the objective function (i.e., value 0). Then, $\Pi \subseteq V^*$, and $V^* = \Pi$. On the other hand, let $V^* = \Pi$; then, since $\boldsymbol{0} \in \Pi$, we have that $\boldsymbol{0} \in V^*$ and, for item 2.1, there is no $\boldsymbol{p} \in \Pi$ such that $\boldsymbol{g}^T\boldsymbol{p} \neq 0$. Then, the thesis holds.

   Concerning the proof of item 2.3, first of all, it is straightforward to see that $V^* = \{\boldsymbol{0}\}$ if $\Pi = \{\boldsymbol{0}\}$. On the other hand, we prove by contradiction that $\Pi = \{\boldsymbol{0}\}$ if $V^* = \{\boldsymbol{0}\}$. Let us assume that $\Pi \neq \{\boldsymbol{0}\}$ and $V^* = \{\boldsymbol{0}\}$; then, there is a feasible direction $\boldsymbol{p} \in \Pi$, $\boldsymbol{p} \neq \boldsymbol{0}$, such that $\boldsymbol{g}^T\boldsymbol{p} \leq 0 = \boldsymbol{g}^T\boldsymbol{0}$. Therefore, there are two possible situations: *i)* $\boldsymbol{g}^T\boldsymbol{p} < 0$ and $\boldsymbol{0}$ is not a solution; *ii)* $\boldsymbol{g}^T\boldsymbol{p} = 0$ and $\boldsymbol{0}$ is not the unique element of V^*. Both cases are a contradiction of the hypothesis $V^* = \{\boldsymbol{0}\}$, proving the thesis.
3. The proof of this item is done by contradiction. Let us assume that $\mathrm{LP}_{\mathrm{new}}$ has solution $\boldsymbol{\rho}^* = (\boldsymbol{p}^*, \beta^*)$ with $\beta^* = 0$. Since $\boldsymbol{x}$ is not Pareto critical, there is $\widehat{\boldsymbol{p}} \in \mathbb{R}^n$ such that $\widehat{\boldsymbol{p}} \neq \boldsymbol{p}^*$ and $\widehat{\boldsymbol{p}}$ is feasible (i.e., $\tilde{G}\widehat{\boldsymbol{p}} < 0$ and $\|\widehat{\boldsymbol{p}}\|_\infty \leq \gamma$). Let δ denotes the euclidean distance between $\boldsymbol{p}^*$ and $\widehat{\boldsymbol{p}}$ and let $\widehat{\beta}$ be such that

$$\widehat{\beta} := \max\left\{-\delta, \max_{i=1,\dots,m} \tilde{\boldsymbol{g}}_i^T\widehat{\boldsymbol{p}}\right\} < 0\,. \tag{24}$$

   Then, $\widehat{\boldsymbol{\rho}} := (\widehat{\boldsymbol{p}}, \widehat{\beta})$ is a feasible point of the problem that satisfies the hypotheses of Proposition 3.1 with respect to $\boldsymbol{\rho}^*$; i.e., $\widehat{\beta} < \beta^* = 0$, $\operatorname{dist}(\boldsymbol{p}^*, \widehat{\boldsymbol{p}}) = \delta \leq (\beta^* - \widehat{\beta})$, and $c_\beta > \|\boldsymbol{g}\|_2$ (last one is a hypothesis of this Lemma). Therefore, for Proposition 3.1, it holds

$$\boldsymbol{c}^T\boldsymbol{\rho}^* > \boldsymbol{c}^T\widehat{\boldsymbol{\rho}}, \tag{25}$$

   which is a contradiction of the hypothesis that $\boldsymbol{\rho}^*$ is solution of $\mathrm{LP}_{\mathrm{new}}$.
4. Let $\boldsymbol{\rho}^*$ be a solution for the problem $\mathrm{LP}_{\mathrm{new}}$. Then, for item 3 of this Theorem, $\beta^* < 0$ and $\boldsymbol{\rho}^* = (\boldsymbol{p}^*, \boldsymbol{\beta}^*)$ satisfies (18), because $\boldsymbol{x}$ is not Pareto critical. It is easy to prove by contradiction that $\beta^* = \max_{i=1,\dots,m} \tilde{\boldsymbol{g}}_i^T\boldsymbol{p}^* =: \delta^*_{\max}$ (item 4.1). Indeed, assuming $\beta^* > \delta^*_{\max}$, for Proposition 3.1 the feasible vector $(\boldsymbol{p}^*, \delta^*_{\max})$ returns a lower value of the objective function, while any vector $(\boldsymbol{p}^*, \beta)$ with $\beta < \delta^*_{\max}$ is not feasible (i.e., does not satisfy (18)).

   Concerning item 4.2, as previously observed, the distance of $\boldsymbol{p}^*$ from H_i is lower bounded by $|\beta^*|$, for each $i = 1, \dots, m$, because $\boldsymbol{\rho}^*$ satisfies (18); then, (23) holds.

□

Now we point the attention of the readers to item 2 of Lemma 3.2. The three sub-items correspond to the three possible situations that occur for a Pareto critical point $\boldsymbol{x} \in \mathbb{R}^n$; specifically, we have:

- Lemma 3.2 - item 2.3 ($V^* = \Pi = \{\boldsymbol{0}\}$, see Fig. 4(a)): the critical point is such that for each direction $\boldsymbol{p} \in \mathbb{R}^n$, $\boldsymbol{p}$ is an ascent direction for at least one objective function; i.e., there is $j \in \{1, \dots, m\}$ such that $\boldsymbol{g}_j^T\boldsymbol{p} > 0$. This is a particular case of item 2.1 of the Lemma (see below).
- Lemma 3.2 - item 2.2 ($\boldsymbol{g}^T\boldsymbol{p} = 0$ for each $\boldsymbol{p} \in \Pi$, $V^* = \Pi$, see Fig. 4(b)): the critical point is such that: *i)* all the non-positive dot product directions shared by all the objective functions are perpendicular to all the gradients; *ii)* case 2.2 of the Lemma (see above).
- Lemma 3.2 - item 2.1 (there is $\boldsymbol{p} \in \Pi$ such that $\boldsymbol{g}^T\boldsymbol{p} \neq 0$, $\boldsymbol{0} \notin V^*$, see Fig. 4(c)): the critical point is such there is $j \in \{1, \dots, m\}$ and there is a non-positive dot product direction shared by all the objectives such that it is a descent direction for the objective function f_j.

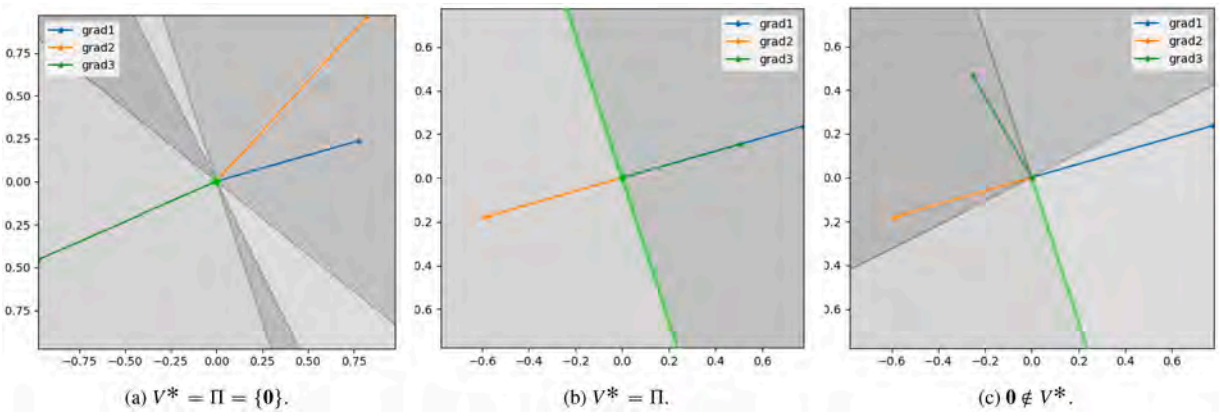


Fig. 4. Visual example in $\mathbb{R}^2$, $m = 3$ objectives, of the three possible situations that occur for a Pareto critical point in LP_{new}. Grey color denotes regions of ascent directions for the objective functions (there are no shared descent directions because $\boldsymbol{x}$ is Pareto critical); the darker the grey color, the more its objectives consider it as an ascent direction. Dark grey lines denote the regions of directions perpendicular to the gradients. We highlight in green color the solution set V^* of the LP problem.

One of the main differences between LP_{base} and LP_{new} is that for LP_{base} we have that $\mathbf{0} \in V^*$ for all the three situations listed above and illustrated in Fig. 4 (see Lemma 3.1). Therefore, LP_{base} can return anytime a null direction $\boldsymbol{p}^* = \mathbf{0}$ if $\boldsymbol{x}$ is Pareto critical, stopping the optimization procedure as a consequence. On the contrary, the new LP problem LP_{new} returns a null direction only if there are no directions in Π that are descent directions for at least one objective; therefore, if there is a direction in Π that is a descent direction for at least one objective, LP_{new} returns a non-null solution $\boldsymbol{p}^*$ and the optimization procedure (5) can continue looking for point $\boldsymbol{x}^{(k+1)}$ along the direction $\boldsymbol{p}^*$ such that $\boldsymbol{x}^{(k+1)}$ is non-dominated by $\boldsymbol{x}^{(k)} = \boldsymbol{x}$. Moreover, it is possible to find $\boldsymbol{x}^{(k+1)}$ with decreased values for the objective functions $f_{j_1}, \dots f_{j_M}$ for which it holds $\boldsymbol{g}_j^T \boldsymbol{p}^* < 0$, $j = j_1, \dots, j_M$. In this situation, in order to find such a kind of new point for the sequence, we define a new backtracking strategy in the next section. Coupling LP_{new} with this new backtracking strategy, we are able to increase the possibility of detecting and/or exploring the Pareto set of a MOO problem, as illustrated by the results of the numerical experiments of Section 5.

Remark 3.1 (Case of two objective functions)**.** We observe that if $m = 2$, i.e., the MOO problem is characterized by two objective functions only, the case $\mathbf{0} \notin V^*$ is not possible for $\boldsymbol{x} \in \mathbb{R}^n$ Pareto critical. Indeed, only the cases corresponding to items 2.1 and 2.2 of Lemma 3.2 are possible. Therefore, when $m = 2$, the differences between LP_{base} and LP_{new} are in the orientation and norm of the non-null solutions $\boldsymbol{p}^*$.

4. New backtracking strategy and convergence analysis

In this section, we describe how to better exploit a MGD method (5) that is based on descent directions evaluated with LP_{new}. In particular, we use a new line-search method for the step lengths $\{\eta_k\}_{k\in\mathbb{N}}$, in order to guarantee that the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ is such that $\boldsymbol{x}^{(k+1)}$ is non-dominated by $\boldsymbol{x}^{(k)}$, for each $k \in \mathbb{N}$. Therefore, we observe that the main difference between our line-search method and the ones typically adopted in literature (e.g., see [21,23–26]) is in the nature of the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$: all the other methods look for a sequence strictly decreasing for all the objective functions; on the contrary, we relax this requirement, accepting in general vectors $\boldsymbol{x}^{(k+1)}$ that are non-dominated by $\boldsymbol{x}^{(k)}$ (but giving priority to the ones that decrease the values of all the objectives).

One of the simplest but most effective implementations of line-search methods for one-objective optimization problems is the backtracking strategy with respect to the Armijo condition [27–30]. This condition can be easily extended to MOO for building the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ strictly decreasing with respect to all the objective functions [21,23–26]. Then, in this work, we extend even further this approach for modifying the properties of the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ as stated above.

Let $\alpha \in (0, 1)$ be the factor of the backtracking strategy and let $c_1 \in (0, 1)$ be the parameter of the Armijo condition. Then, at each step $k > 0$, we check if the Armijo condition

$$f_i(\boldsymbol{x}^{(k)} + \eta_0^{(k)} \boldsymbol{p}^{*(k)}) \leq f_i(\boldsymbol{x}^{(k)}) + c_1 \eta_0^{(k)} \boldsymbol{g}_i(\boldsymbol{x}^{(k)})^T \boldsymbol{p}^{*(k)}, \tag{26}$$

is satisfied for the base-value of the step-size $\eta^{(k)} = \eta_0^{(k)}$ and for each $i = 1, \dots, m$. If it is not satisfied, we decrease the step-size by the factor α until the condition

$$f_i(\boldsymbol{x}^{(k)} + \eta_t^{(k)} \boldsymbol{p}^{*(k)}) \leq f_i(\boldsymbol{x}^{(k)}) + c_1 \eta_t^{(k)} \boldsymbol{g}_i(\boldsymbol{x}^{(k)})^T \boldsymbol{p}^{*(k)}, \qquad \text{A}_i^t$$

is satisfied for the step-size $\eta_t^{(k)} := \eta_{t-1}^{(k)} \alpha = \eta_0^{(k)} \alpha^t$, $t > 0$, and for each $i = 1, \dots, m$.

However, it is possible that $\eta_t^{(k)}$ does not satisfy A_i^t, for each $t \geq 0$; in particular, it happens when $\boldsymbol{x}^{(k)}$ is Pareto critical and, therefore, $\boldsymbol{p}^{*(k)}$ is not a descent direction for all the objectives, but only a non-positive dot product direction or the null vector (see

Lemma 3.2). Under these circumstances, instead of stopping the sequence like the other MOO methods (e.g., see [21]), if $\boldsymbol{p}^{*(k)} \neq \boldsymbol{0}$ we make a last check:

$$\exists\, i \in \{1, \dots, m\} \quad s.t. \quad f_i(\boldsymbol{x}^{(k)}) > f_i(\boldsymbol{x}^{(k)} + \hat{\eta}\, \boldsymbol{p}^{*(k)}) \tag{27}$$

where $\hat{\eta} > 0$ is a fixed arbitrary small parameter. Therefore, if (27) is satisfied, setting $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \hat{\eta}\, \boldsymbol{p}^{*(k)}$ returns a new point $\boldsymbol{x}^{(k+1)}$ for the sequence that is non-dominated by $\boldsymbol{x}^{(k)}$, and the sequence can continue.

Then, we can summarize the sequence generated by the line search method described above as

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathbb{R}^n,\ \{\eta_0^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^+,\ \hat{\eta} > 0 & \text{given} \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \eta^{(k)} \boldsymbol{p}^{*(k)}, & k \geq 0 \end{cases}, \qquad \text{BT}_{\text{new}}$$

where

$$\eta^{(k)} = \begin{cases} \eta_\tau^{(k)} = \eta_0^{(k)} \alpha^\tau & \text{if } \boldsymbol{x}^{(k)} \text{ is not Par. crit. } (\tau, \text{1st value of } t \geq 0 \text{ satisfying } \text{A}_i^t) \\ \hat{\eta} & \text{if } \boldsymbol{x}^{(k)} \text{ is Par. crit. and does not dominate } \boldsymbol{x}^{(k)} + \hat{\eta}\, \boldsymbol{p}^{*(k)} \\ 0 & \text{otherwise (i.e., } \boldsymbol{x}^{(k)} \text{ is Par. crit. and dominates } \boldsymbol{x}^{(k)} + \hat{\eta}\, \boldsymbol{p}^{*(k)}) \end{cases}. \tag{28}$$

The new backtracking strategy can be used in general with any MGD method, independently of the type of sub-problems used to compute the directions $\boldsymbol{p}^{*(k)}$; therefore, it can be used also with a MGD method based on LP_{base}. Nonetheless, we recall that BT_{new} has been specifically designed to take advantage of the properties of LP_{new}; in particular, we recall the observations related to item 2.3 of Lemma 3.2.

Remark 4.1 (Strictly decreasing backtracking as a special case of the new one)**.** Line search methods designed for building strictly decreasing sequences for all the objectives can be interpreted as a special case of the new type of backtracking strategies, where $\hat{\eta} = 0$.

4.1. Convergence results

If we look at all the possible sequences $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ determined by the new backtracking strategy BT_{new}, we can identify three main types of sub-sequences of consecutive elements:

1. $\{\boldsymbol{x}^{(k+m)}\}_{m=0}^{M}$, $k, M \geq 0$, such that $\boldsymbol{x}^{(k+M)}$ is Pareto critical, $\eta^{(k+M)} = \hat{\eta}$, and $\boldsymbol{x}^{(k+m)}$ is not Pareto critical for each $m < M$.
   We define this type of sub-sequence as *$\hat{\eta}$-Pareto Critical end* ($\text{PC}_{\hat{\eta}}$);
2. $\{\boldsymbol{x}^{(k+m)}\}_{m\in\mathbb{N}}$, $k \geq 0$, such that there is $M \in \mathbb{N}$ for which $\boldsymbol{x}^{(k+m)}$ is Pareto critical and $\eta^{(k+m)} = 0$, for each $m \geq M$, and $\boldsymbol{x}^{(k+m)}$ is not Pareto critical for each $m < M$.
   We define this type of sub-sequence as *0-Pareto Critical end* (PC_0);
3. $\{\boldsymbol{x}^{(k+m)}\}_{m\in\mathbb{N}}$, $k \geq 0$, such that $\boldsymbol{x}^{(k+m)}$ is not Pareto critical for each $m \geq 0$.
   We define this type of sub-sequence as *non-Pareto Critical* (nPC);

More precisely, we have that any sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ has one of the following structures:

1. $N \in \mathbb{N}$ consecutive $\text{PC}_{\hat{\eta}}$ sub-sequences, followed by one PC_0 sub-sequence;
2. $N \in \mathbb{N}$ consecutive $\text{PC}_{\hat{\eta}}$ sub-sequences, followed by one nPC sub-sequence;
3. Infinite consecutive $\text{PC}_{\hat{\eta}}$ sub-sequences.

On the other hand, a sequence generated for being strictly decreasing for all the objectives coincides with a sub-sequence of type nPC or, if it reaches a Pareto critical point, with a sub-sequence of type PC_0. In [21], the authors state that when such a kind of sequences are of the nPC type, assuming proper limitation hypotheses on the objective functions, they have at least one accumulation point, and each accumulation point of the sequence is a Pareto critical point. Below, we report this theorem ([21, Theorem 1]), adapted to the notation used in this work.

Theorem 4.1 (Theorem 1 - [21])**.** *Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be a sequence defined by BT_{new}, but with $\hat{\eta} = 0$ (see Remark 4.1). Let $\boldsymbol{x}^{(k)}$ be not Pareto critical, for each $k \in \mathbb{N}$. Then:*

1. *$\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ stays bounded and has at least one accumulation point, if the function $\boldsymbol{f}$ (see Definition 2.1) has bounded level sets in the sense that $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{x}^{(0)})\}$ is bounded;*
2. *Every accumulation point of the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ is a Pareto critical point.*

Now, we state a theorem for characterizing the convergence properties of the sequences obtained using the new backtracking strategy proposed; i.e., sequences determined by BT_{new}. In this theorem, we exclude the case of a sequence ending with a PC_0 sub-sequence because, in that case, the results are trivial.

Theorem 4.2. *Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be a sequence defined by BT_{new} and let C denotes the set of all the Pareto critical points of $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^n$ (see Definition 2.1). Let us assume the sequence does not end with a PC_0 sub-sequence. Then:*

1. $\liminf_{k\to+\infty} \text{dist}(\boldsymbol{x}^{(k)}, C) = 0$;

2. *if* $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ *ends with a sub-sequence* $\{\boldsymbol{x}^{(k_0+m)}\}_{m\in\mathbb{N}}$ *of nPC type,* $k_0 \geq 0$ *fixed, and* $\{\boldsymbol{x}\in\mathbb{R}^n \mid \boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{x}^{(k_0)})\}$ *is bounded, then* $\{\boldsymbol{x}^{(k_0+m)}\}_{m\in\mathbb{N}}$ *is bounded, admits at least one accumulation point and all its accumulation points are Pareto critical;*

Proof. The structure of the sequence can be of only two types:

(i) $N \in \mathbb{N}$ consecutive $\mathrm{PC}_{\hat{\eta}}$ sub-sequences, followed by one nPC sub-sequence;
(ii) Infinite consecutive $\mathrm{PC}_{\hat{\eta}}$ sub-sequences.

In the first case (i), we have to prove both the theses of the theorem. Nonetheless, using Theorem 4.1 [Theorem 1 - [21]] the proof of item 2 is trivial and, as a consequence, also the proof of item 1.

Now, for case (ii), the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ is made of infinite consecutive $\mathrm{PC}_{\hat{\eta}}$ sub-sequences, i.e.:

$$\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}} = \{\boldsymbol{x}^{(k_1+m)}\}_{m=0}^{M_1} \cup \{\boldsymbol{x}^{(k_2+m)}\}_{m=0}^{M_2} \cup \cdots \cup \{\boldsymbol{x}^{(k_i+m)}\}_{m=0}^{M_i} \cup \cdots = \\ = \bigcup_{i\geq 1} \{\boldsymbol{x}^{(k_i+m)}\}_{m=0}^{M_i},$$

where $k_1 = 1$, $k_i = k_{i-1} + M_{i-1} + 1$, and $M_{i-1} \in \mathbb{N}$ for each $i \geq 2$. Moreover, by definition, each one of these $\mathrm{PC}_{\hat{\eta}}$ sub-sequences $\{\boldsymbol{x}^{(k_i+m)}\}_{m=0}^{M_i}$ is such that $\boldsymbol{x}^{(k_i+M_i)}$ is Pareto critical.

Then, for each $k \geq 0$, there is $i \geq 1$ such that $h_i = k_i + M_i \geq k$, and $\boldsymbol{x}^{(h_i)}$ is Pareto critical; therefore, it holds

$$\inf_{h\geq k} \mathrm{dist}(\boldsymbol{x}^{(h)}, C) = \mathrm{dist}(\boldsymbol{x}^{(h_i)}, C) = 0$$

and

$$\liminf_{k\to+\infty} \mathrm{dist}(\boldsymbol{x}^{(k)}, C) = \sup_{k\geq 0}\{\inf_{h\geq k} \mathrm{dist}(\boldsymbol{x}^{(h)}, C)\} = 0\,.$$

□

4.2. Implementation

Here, we report the pseudocode of an algorithm (Algorithm 1) that implements a MGD procedure with respect to the backtracking strategy defined in this section. We assume a maximum number of iterations $K \in \mathbb{N}$, a maximum number of backtracking steps $\Theta \in \mathbb{N}$, and, for simplicity, we set $\hat{\eta} = \eta_0\alpha^{\Theta}$, where $\eta_0 \in \mathbb{R}_+$ is the base value of the step-size for each iteration. The direction $\boldsymbol{p}^*$ described in the algorithm is the vector obtained solving the LP problem $\mathrm{LP}_{\mathrm{base}}$ of [21], the LP problem $\mathrm{LP}_{\mathrm{new}}$ proposed in this work, or any other problem defined with the same scope.

4.2.1. Storage of non-dominated intermediate points

Until now, we focused on the advantages of finding a new non-dominated point $\boldsymbol{x}^{(k+1)}$ even if $\boldsymbol{x}^{(k)}$ is Pareto critical. Nonetheless, since in MOO it is crucial to identify as many Pareto optimals as possible, it is better if we do not "forget" $\boldsymbol{x}^{(k)}$ when it is not dominated by $\boldsymbol{x}^{(k+1)}$; indeed, we recall that is possible to have both $\boldsymbol{x}^{(k+1)}$ non-dominated by $\boldsymbol{x}^{(k)}$, and vice-versa.

Therefore, we can modify Algorithm 1 adding a "domination-check" for $\boldsymbol{x}^{(k)}$ with respect to $\boldsymbol{x}^{(k+1)}$; specifically, if $\boldsymbol{x}^{(k)}$ is not dominated by $\boldsymbol{x}^{(k+1)}$, we store it into a list C. Then, at the end of the procedure, the algorithm returns the last vector computed for the sequence, denoted by $\hat{\boldsymbol{x}}$, and all the $\boldsymbol{x} \in C$ that are non-dominated by all the other vectors in $C \cup \{\hat{\boldsymbol{x}}\}$; this set of vectors will represent the (approximation of) Pareto optimals discovered by the optimization method during its run (not necessarily global Pareto optimals, due to the local nature of the method).

The result of these modifications is Algorithm 2.

4.2.2. Blockwise implementation for searching global pareto optimals

We recall that MGD methods have a local nature. Therefore, all the Pareto critical points returned by the procedures are candidates to be approximations of global and/or local Pareto optimals, mainly depending on the starting point $\boldsymbol{x}^{(0)}$. Therefore, a basic multi-start approach for the optimization procedure, made with respect to a set of $N \in \mathbb{N}$ random starting points

$$X^{(0)} := \{\boldsymbol{x}_1^{(0)}, \ldots, \boldsymbol{x}_N^{(0)}\} \subset \mathbb{R}^n,$$

can be a simple but efficient method for exploring and detecting as much as possible the Pareto set and the Pareto front of a MOO problem.

Algorithm 1 Let us consider the unconstrained MOO problem (1). Then, we define the following MGD algorithm for the implementation of the descent method BT_{new}.

Data: $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ starting point for (5); $c_1, \alpha \in (0,1)$ parameters for the Armijo condition; η_0 starting value for the step length; $\Theta \in \mathbb{N}$ maximum number of backtracking steps; $K \in \mathbb{N}$ maximum number of iterations; $\mathcal{P}$ sub-problem for computing $\boldsymbol{p}^*$ feasible direction with respect to $\boldsymbol{x}$.

Procedure:

for $k = 0, \dots, (K-1)$ **do**
  State $\boldsymbol{p}^{*(k)} \leftarrow$ solve $\mathcal{P}$ with respect to $\boldsymbol{x}^{(k)}$ ▷ e.g., solve $\mathcal{P}$ =LP_{base} or $\mathcal{P}$ =LP_{new} for finding $\boldsymbol{p}^{*(k)}$
  $\eta^{(k)} \leftarrow \eta_0$
  for $t = 0, \dots, (\Theta-1)$ **do**
    if A_i^t is true for each $i = 1, \dots, m$ **then**
      break
    else
      $\eta^{(k)} \leftarrow \eta^{(k)}\alpha$
    end if
  end for
  $\widetilde{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(k)} + \eta^{(k)}\boldsymbol{p}^{*(k)}$
  if $\widetilde{\boldsymbol{x}}$ is dominated by $\boldsymbol{x}^{(k)}$ **then** ▷ Always false, if A_i^t is true $\forall\, i = 1, \dots, m$
    break
  else
    $\boldsymbol{x}^{(k+1)} \leftarrow \widetilde{\boldsymbol{x}}$
  end if
end for
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(k)}$
return : $\hat{\boldsymbol{x}}$

Algorithm 2 Let us consider the unconstrained MOO problem (1). Then, define the following MGD algorithm for the implementation of the descent method (5).

Data: $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ starting point for (5); $c_1, \alpha \in (0,1)$ parameters for the Armijo condition; η_0 starting value for the step length; $\Theta \in \mathbb{N}$ maximum number of backtracking steps; $K \in \mathbb{N}$ maximum number of iterations; $\mathcal{P}$ sub-problem for computing $\boldsymbol{p}^*$ feasible direction with respect to $\boldsymbol{x}$.

Procedure:

$C \leftarrow$ empty list ▷ Initialization of the list of $\boldsymbol{x}^{(k)}$ non-dominated by $\boldsymbol{x}^{(k+1)}$
for $k = 0, \dots, (K-1)$ **do**
  $\boldsymbol{p}^{*(k)} \leftarrow$ solve $\mathcal{P}$ with respect to $\boldsymbol{x}^{(k)}$ ▷ e.g., solve $\mathcal{P}$ =LP_{base} or $\mathcal{P}$ =LP_{new} for finding $\boldsymbol{p}^{*(k)}$
  $\eta^{(k)} \leftarrow \eta_0$
  for $t = 0, \dots, (\Theta-1)$ **do**
    if A_i^t is true for each $i = 1, \dots, m$ **then**
      break
    else
      $\eta^{(k)} \leftarrow \eta^{(k)}\alpha$
    end if
  end for
  $\widetilde{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(k)} + \eta^{(k)}\boldsymbol{p}^{*(k)}$
  if $\widetilde{\boldsymbol{x}}$ is dominated by $\boldsymbol{x}^{(k)}$ **then**
    break
  else
    $\boldsymbol{x}^{(k+1)} \leftarrow \widetilde{\boldsymbol{x}}$
  end if
  if $\boldsymbol{x}^{(k)}$ is not dominated by $\boldsymbol{x}^{(k+1)}$ **then**
    Add $\boldsymbol{x}^{(k)}$ to C
  end if
end for
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(k)}$
$\bar{C} \leftarrow \{\boldsymbol{x} \in C \mid \boldsymbol{x}$ *non-dom. by* $\boldsymbol{y},\ \forall \boldsymbol{y} \in C \cup \{\hat{\boldsymbol{x}}\},\ \boldsymbol{y} \neq \boldsymbol{x}\}$
return : $\hat{\boldsymbol{x}}, \bar{C}$

Concerning a multi-start approach, it is interesting to observe that the solutions $\boldsymbol{\rho}_1^*, \dots, \boldsymbol{\rho}_N^*$ of N LP problems like LP_{new}, if concatenated into a vector $\boldsymbol{\rho}^* = (\boldsymbol{\rho}_1^*, \dots, \boldsymbol{\rho}_N^*) = ((\boldsymbol{p}_1^*, \beta_1^*), \dots, (\boldsymbol{p}_N^*, \beta_N^*)) \in \mathbb{R}^{N(n+1)}$, are solution of the LP problem

$$\begin{cases} \min_{\boldsymbol{\rho} \in \mathbb{R}^{N(n+1)}} [\, \boldsymbol{c}_1^T \mid \cdots \mid \boldsymbol{c}_N^T \,]\, \boldsymbol{\rho} \\ \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{bmatrix} \boldsymbol{\rho} \leq \boldsymbol{0} \\ -\gamma_j \boldsymbol{e} \leq \boldsymbol{p}_j \leq \gamma_j \boldsymbol{e}, \qquad \forall j = 1, \dots, N \\ \beta_j \leq 0, \qquad \forall j = 1, \dots, N \end{cases}, \tag{29}$$

and vice-versa; where $\boldsymbol{c}_j \in \mathbb{R}^{n+1}$, $A_j \in \mathbb{R}^{m \times (n+1)}$, and $\gamma_j \in \mathbb{R}_+$ are: the vector of the linear objective function, the matrix of inequality constraints, and the scalar for boundary values of $\boldsymbol{p}_j$, respectively, of the jth LP problem, for each $j = 1, \dots, N$.

Then, if the user cannot directly parallelize the multi-start procedure, they can still exploit (29) for a fast computation of the N directions $\boldsymbol{p}_1^*, \dots, \boldsymbol{p}_N^*$. Of course, techniques for fastening the evaluation of the Jacobians $G_1, \dots, G_N$ are suggested, but their discussion does not fall within the scope of this work.

A similar block-wise parallelization can be implemented also for problem LP_{base}.

5. Numerical experiments

In this section, we study and analyze the behavior of the new method LP_{new} for computing the shared descent direction of multiple objectives, using the new backtracking strategy BT_{new} for MGD methods. Specifically, we run Algorithm 2, using LP_{new} for computing the direction $\boldsymbol{p}^{*(k)}$, with parameter $c_\beta^{(k)} = c_\beta(\boldsymbol{x}^{(k)}) = \|\boldsymbol{g}(\boldsymbol{x}^{(k)})\|_2 + 1$. Then, the analysis is made through a comparison with [21], representing the baseline; i.e., the baseline is represented by

LP_{base}

, using a backtracking strategy designed for building a sequence strictly decreasing for all the objectives; we report the pseudocode of a MGD implementation using this backtracking strategy in Appendix A (see Algorithm Appendix A). Nonetheless, we study also how the behavior of LP_{base} changes if BT_{new} is used (i.e., we run Algorithm 1 with respect to LP_{base}) and how the behavior of LP_{new} changes if the backtracking strategy for strictly decreasing objectives is used (i.e., we run Algorithm Appendix A with respect to LP_{new}).

Summarizing, we analyze the behavior of:

- Algorithm Appendix A and Algorithm 2 run with respect to LP_{base}. We denote them as BT_{base}-LP_{base} and BT_{new}–LP_{base}, respectively;
- Algorithm Appendix A and Algorithm 2 run with respect to LP_{new}. We denote them as BT_{base}-LP_{new} and BT_{new}–LP_{new}, respectively;.

The baseline is represented by BT_{base}-LP_{base} (see [21]), while the new method we propose is represented by BT_{new}–LP_{new}. All the other "intermediate" cases are added to the analysis to better understand the effects of using different backtracking strategies and different shared descent directions in a MGD method.

All the cases are compared analyzing the results obtained by applying them to three well-known MOO test problems: the Fonseca-Fleming problem [31], the Kursawe problem [32], and the Viennet problem [33]. For all these problems, all the algorithms are executed with respect to $N = 500$ randomly sampled starting points (exploiting a blockwise implementation, see Section 4.2.2); the backtracking strategy is characterized by a maximum number of steps $\Theta = 40$ and parameters $c_1 = 10^{-9}$, $\alpha = 0.8$, and $\eta_0 = 1$ for the Armijo condition A_i^t. For more details about the test problems, the starting point samplings, etc., see Section 5.1.

The comparison between the methods is performed by measuring the ratio of sequences, among all the N ones, that have detected at least one Pareto critical that is non-dominated by all the other Pareto critical points detected by itself and the other sequences. Indeed, such a kind of Pareto critical points can be considered as a good approximation of a global Pareto optimal and, therefore, it means that the sequence reached at least one point near to the Pareto set of the problem.

Specifically, let $\bar{C}_j$ be the set containing the outputs of a MGD algorithm, starting from $\boldsymbol{x}_j^{(0)}$, for each $j = 1, \dots, N$; i.e., $\bar{C}_j = \{\hat{\boldsymbol{x}}\}$ for Algorithm Appendix A, while $\bar{C}_j$ contains $\hat{\boldsymbol{x}}$ and all the non-dominated critical points of the sequence for Algorithm 2 (see the algorithm's pseudocode). Let $\bar{C}^N$ be the union of all the sets $\bar{C}_j$, i.e., $\bar{C}^N := \bigcup_{j=1}^N \bar{C}_j$, and let $\widetilde{C}_j$ be the set points in $\bar{C}_j$ non-dominated by all the other points in $\bar{C}^N$, i.e.

$$\widetilde{C}_j := \{\boldsymbol{x} \in \bar{C}_j \mid \boldsymbol{x} \text{ non-dom. by } \boldsymbol{y},\ \forall \boldsymbol{y} \in \bar{C}^N, \boldsymbol{y} \neq \boldsymbol{x}\}. \tag{30}$$

Then, for a MGD algorithm, we define the index

$$\text{P}^N := \frac{\#\text{ of non-empty } \widetilde{C}_j}{N} = \frac{\left|\{j \mid \widetilde{C}_j \neq \emptyset,\ j = 1, \dots, N\}\right|}{N}, \tag{31}$$

where the superscript N denotes the number of sequences used for computing the index. We define this index as the *global Pareto ratio* of the MGD algorithm, with respect to N runs.

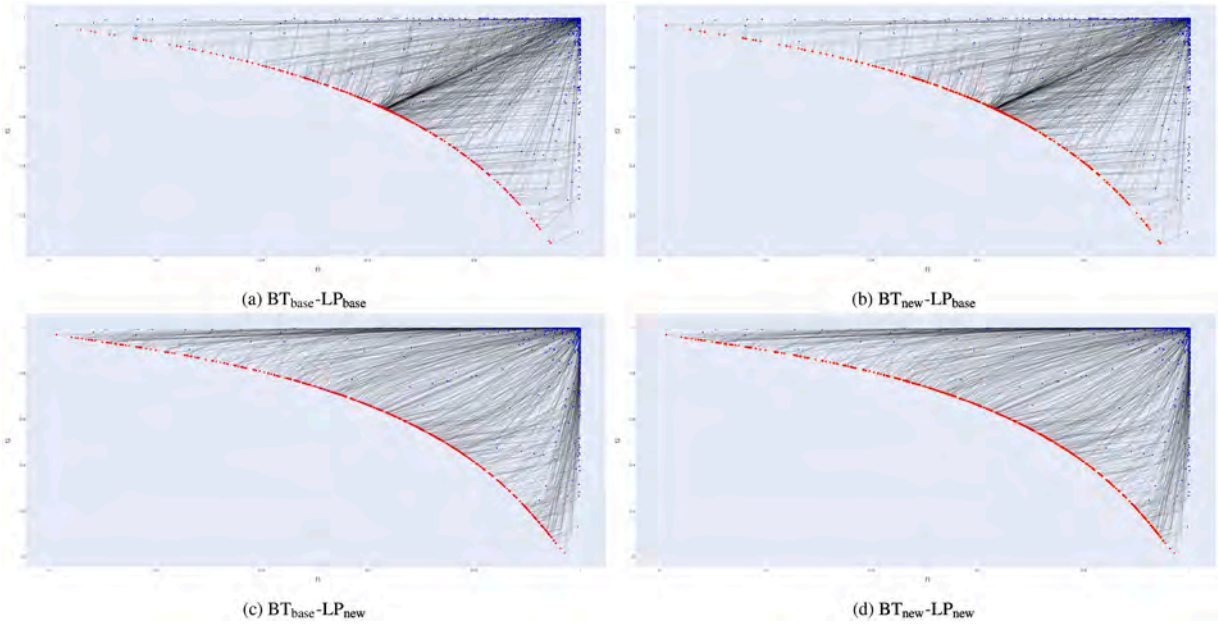


Fig. 5. Fonseca-Flemig test case. Movement of the $N = 500$ sequences in the objectives' space. The blue dots are the starting points' images $\boldsymbol{f}(\boldsymbol{x}_j^{(0)}) \in \mathbb{R}^2$, $j = 1, \dots, N$, while the red dots are the last points of the sequence. The black, dotted, and piece-wise linear curves describe the movement of each sequence from its starting point to its last element. Critical points stored during Algorithm 4.2 are not represented because they are (almost) coincident with red dots.

Remark 5.1 (About the global Pareto ratio index)**.** We introduce index (31) in order to analyze the performance of different MGD algorithms independently on the availability of a known Pareto set and Pareto front. Alternative indices can be used, such as the Hypervolume (HV) indicator, which is able to measure the "quality" of a Pareto set/front through the hyper-volume of the dominated region in the objectives' space (see [34,35]). Nonetheless, the HV indicator (and the other MOO indicators) focuses too much on the spread properties of the detected global Pareto optimals, with the drawback of finding often situations where very similar indicator values are measured, despite the very different number of points in the Pareto front; for this reason. we prefer instead to use index (31) for our analyses, because it is better for quantifying the ability of a MGD algorithm to reach the Pareto set of the problem, varying the starting point. However, we will report also the normalized HV indicator for the numerical experiments of this section, and not only the index P^N.

For a better comparison of the methods, we conclude the beginning of this section with a discussion on their computational costs. Actually, the computational costs of LP_{base} and LP_{new} can be considered equivalent, since for both the problems we have to compute m gradients (one for each objective function), while the computation of $\boldsymbol{g}$ and γ (see Eqs. (8) and (14), respectively) are negligible; moreover, also the cost of the solution of the LP problem is the same. On the other hand, there are instead differences in the costs of Algorithm Appendix A (backtracking strategy BT_{base}) and Algorithm 2 (backtracking strategy BT_{new}); even if the computational costs in terms of time and operations are almost the same for the iterative procedure that builds the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$, Algorithm 2 is characterized by these additional costs: (*i*) it requires a greater amount of memory to store in the list C all the vectors $\boldsymbol{x}^{(k)}$ that are not dominated by $\boldsymbol{x}^{(k+1)}$; (*ii*) if the cardinality of the list C is large, the computation of the set $\bar{C}$ at the end of the iterative procedure (see line 23 of the algorithm) can be not negligible, requiring more computational efforts.

In general, we notice that a possible limitation to the scalability of all the methods is the cost of the m gradient computations (i.e., the computation of the Jacobian of $\boldsymbol{f}$, $\boldsymbol{f} = (f_1, \dots, f_m)$); indeed, when both $n \gg 1$ and $m \gg 1$, the costs of derivative computations can be significant if repeated for each iteration, especially in case of derivatives approximations (e.g., finite differences).

5.1. Multi-objective test problems

In this section, we report the formulations of the three MOO test problems used in the numerical experiments, together with information about the selection of starting points and the maximum number of steps K used.

- **Fonseca-Fleming** [31,36]**.** The objective functions of the MOO problem are

$$f_1 = 1 - e^{-\sum i=1^n (x_i - 1/\sqrt{n})^2}$$
$$f_2 = 1 - e^{-\sum i=1^n (x_i + 1/\sqrt{n})^2}$$

The starting points for the MGD methods are sampled with random uniform distribution $\mathcal{U}([-2, 2]^n)$, $n = 3$. The maximum number of steps used for the MGD methods is $K = 250$.

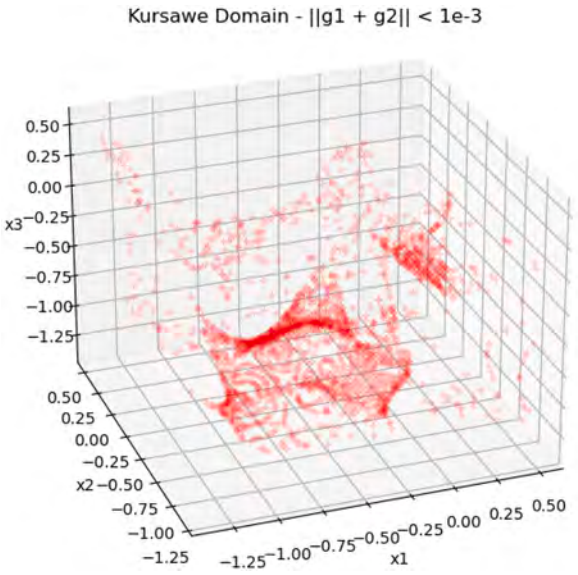


Fig. 6. Kursawe test case. The red dots are points in the cube $[-1.5, 0.5]^3$ (see Section 5.1) where the normalized gradients are (almost) opposite (i.e., $\|\bar{g}_1 + \bar{g}_2\|_2 < 10^{-3}$).

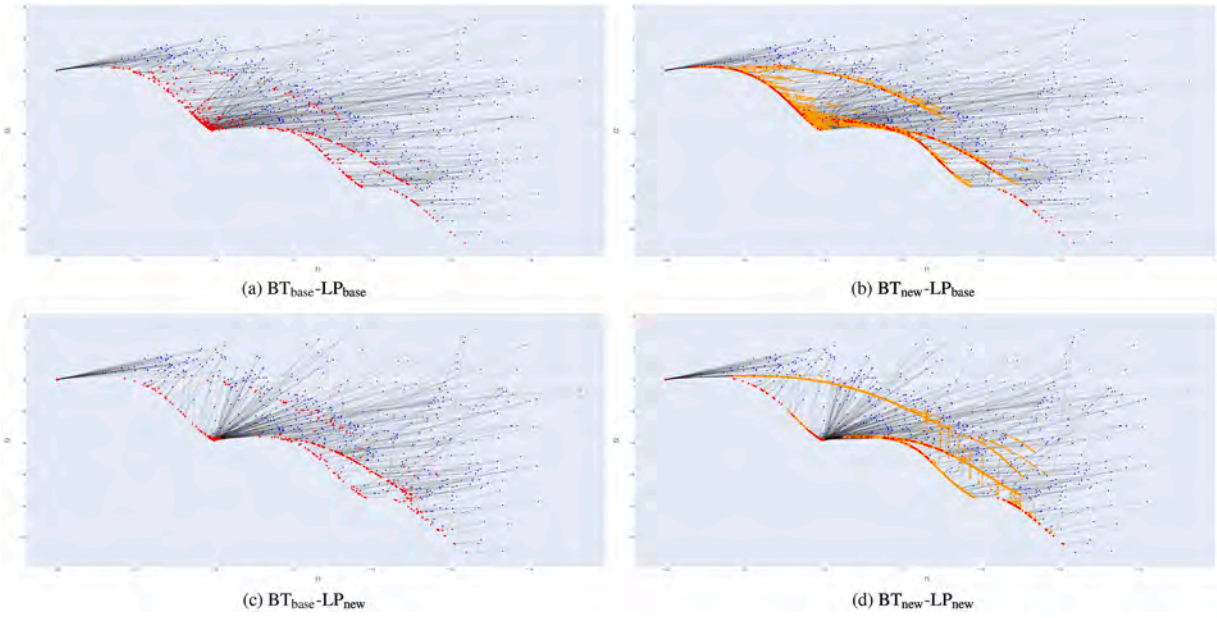


Fig. 7. Kursawe test case. Movement of the $N = 500$ sequences in the objectives' space. The blue dots are the starting points' images $\boldsymbol{f}(\boldsymbol{x}_j^{(0)}) \in \mathbb{R}^2$, $j = 1, \dots, N$, while the red dots are the last points of the sequence. The orange dots are the critical points stored during Algorithm 4.2, before the final pruning (see the pseudocode). The black, dotted, and piece-wise linear curves describe the movement of each sequence from its starting point to its last element.

- **Kursawe** [32]**.** The objective functions of the MOO problem are

$$f_1 = \sum_{i=1}^{2} -10\, e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}$$

$$f_2 = \sum_{i=1}^{3} (|x_i|^{0.8} + 5\sin(x_i^3))$$

The starting points for the MGD methods are sampled with random uniform distribution $\mathcal{U}([-1.5, 0.5]^n)$, $n = 3$. The maximum number of steps used for the MGD methods is $K = 1500$.

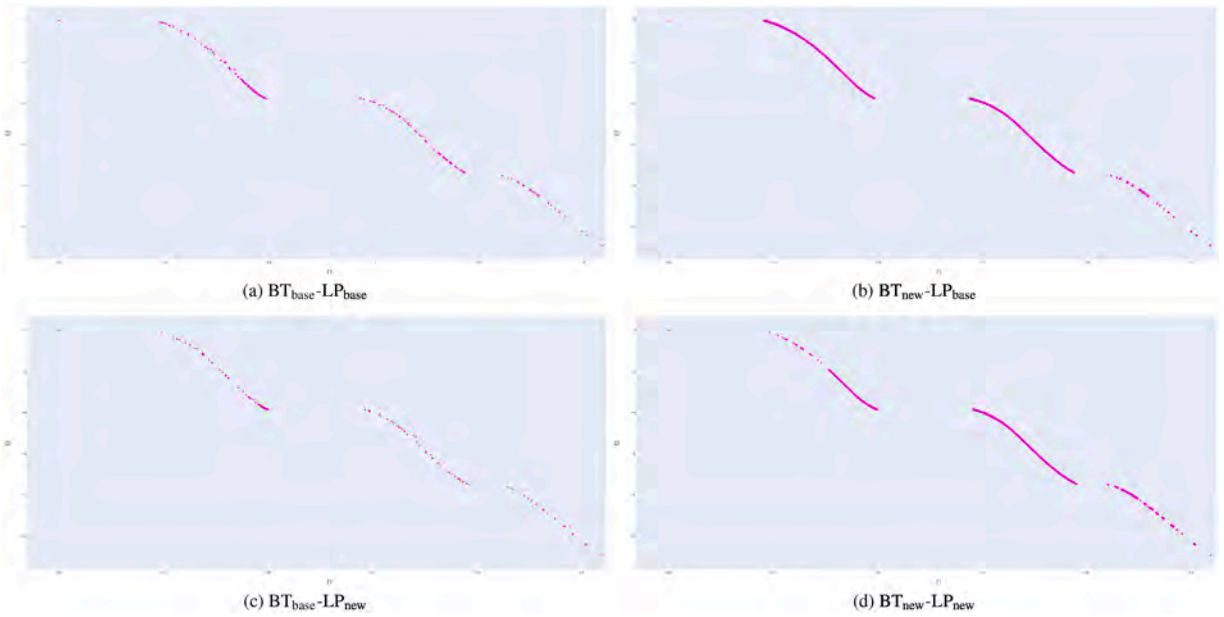


Fig. 8. Kursawe test case. Images in the objectves' space of the non-dominated points among all the outputs returned by the MGD algorithms; i.e., images of the points in $\bigcup_{j=1}^{N} \widetilde{C}_j$ (see (30)).

- **Viennet** [33]**.** The objective functions of the MOO problem are

$$f_1 = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$$
$$f_2 = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 + x_2 + 1)^2}{27} + 15$$
$$f_3 = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1\, e^{-(x_1^2 + x_2^2)}$$

The starting points for the MGD methods are sampled with random uniform distribution $\mathcal{U}([-3, 1.5]^n)$, $n = 2$. The maximum number of steps used for the MGD methods is $K = 7500$.

5.2. Numerical results and analyses

In this subsection, we analyze and study the results obtained by the four given MGD algorithms applied to the three test problems illustrated in the previous subsection (Fonseca-Fleming, Kursawe, and Viennet). We start analyzing the results for the Fonseca-Fleming problem, focusing on the different paths generated by using directions that are solutions of LP_{base} or LP_{new}; in this case, the type of backtracking strategy is not crucial, due to the characteristics of the objective functions. On the contrary, when we continue to the results for the Kursawe problem, we observe how the usage of BT_{new} (i.e., Algorithm 2) improves concretely the global Pareto ratio P^N. In the end, we analyze the results for the Viennet problem; for this latter case, we observe that only the MGD method BT_{new}–LP_{new} is able to return a good P^N value, outperforming all the other methods. A summary of the global Pareto ratio values for all the methods, with respect to all the problems, is reported in Table 1.

In Table 1, we report also the normalized HV indicator computed with respect to the non-dominated final points of the converged sequences. According to the comments in Remark 5.1, we notice that even with very different quantities of non-dominated points, the normalized HV indicator does not change consistently, because this measure focuses on the spread of the points in the objective space; for example, in the Viennet problem we have 464 non-dominated points (92.80% of 500 sequences) for the case BT_{new}–LP_{new} and 174 non-dominated points (34.80% of 500 sequences) for the case BT_{base}-LP_{new}, but both the cases have a normalized HV indicator that is almost the same (0.7710 and 0.7712, respectively). For this reason, we will focus our analyses on the comparison of the index P^N.

5.2.1. Fonseca-fleming

With respect to the Fonseca-Fleming test problem, all the MGD methods return 100% of P^N (see Table 1). Looking only at these values, we deduce that the sequences built with LP_{new} are a valuable alternative to the ones built with LP_{base}. We do not see differences in the performances while changing the backtracking strategy because the characteristics of the problem are such that A_i^t is very easy to be satisfied in regions far from the Pareto set (e.g., both the objective functions are convex); therefore, all the sequences are decreasing with respect to all the objectives except when an approximation of a global Pareto optimal has already been reached.

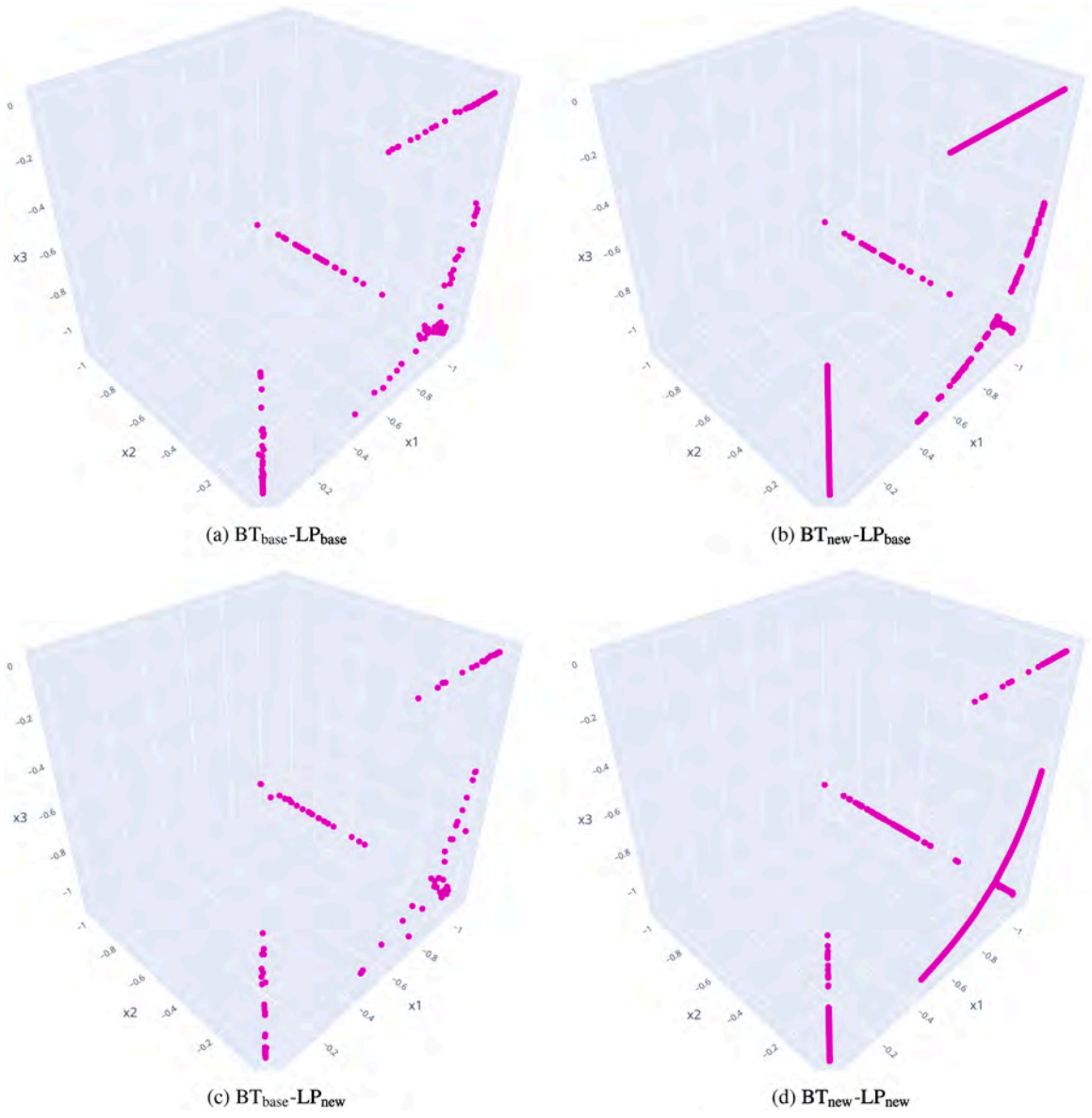


Fig. 9. Kursawe test case. Non-dominated points among all the outputs returned by the MGD algorithms; i.e., the points in $\bigcup_{j=1}^{N} \widetilde{C}_j$ (see (30)).

However, looking at the paths of the sequences' images in the objectives' space in Fig. 5, we observe different behaviors for LP_{base} and LP_{new}. For both the methods, the paths end (approximately) on the Pareto front of the problem, but the paths generated with LP_{base} are attracted more by the "center" of the front (Fig. 5(a) and (b)), while the paths generated with LP_{new} are distributed more uniformly on the front (Fig. 5(c) and (d)).

Summarizing, with these results we show that LP_{new} is able to generate good sequences for a MGD algorithm, but the sequences we obtain are characterized by a different behavior from the ones generated with LP_{base} (as expected, by construction of LP_{new}).

5.2.2. Kursawe

The Kursawe problem is more complicated than the Fonseca-Fleming one. In this case, analogously to the Fonseca-Fleming test, we observe similar performances but different path behaviors for LP_{base} and LP_{new} (given the same backtracking strategy); actually, the performances of methods based on LP_{new} are a bit smaller than the performances of methods based on LP_{base} (see Table 1). The different path behaviors of the two methods are evident if Fig. 7; in particular, we have that LP_{new} generates sequences attracted more by the "center" of the front, while LP_{base} generates sequences mainly focused in reducing the value of f_1 (i.e., "leftward movement" in the objectives' space).

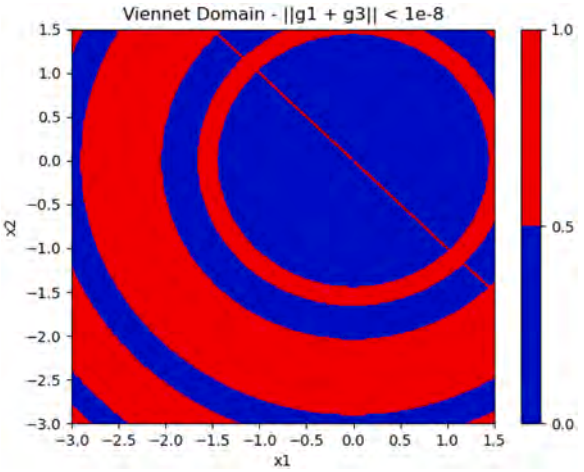


Fig. 10. Viennet test case. The red color denotes the points in the square $[-3, 1.5]^2$ (see Section 5.1) where the normalized gradient of the first objective function is (almost) equal to minus the normalized gradient of the third objective function (i.e., $\|\bar{g}_1 + \bar{g}_3\|_2 < 10^{-8}$); blue color, otherwise. In particular, red regions are regions of critical points of the same type illustrated in Fig. 4(c).

Table 1
For each case, with respect to $N = 500$ sequences $\{x_j^{(k)}\}_{k\in\mathbb{N}}$, $j = 1, \dots, N$, we report the global Pareto convergence ratios (P^N, represented as percentages) and the normalized HV indicator of the non-dominated final points of the converged sequences. The HV indicator is computed with respect to the reference vectors: $(1, 1)$ for Fonseca-Fleming, $(-14, -0.1)$ for Kursawe, and $(9, 36, 0.3)$ for Viennet. The HV indicator is normalized with respect to the hypervolume delimited by the reference vector and an opposite vector that is: $(0, 0)$ for Fonseca-Fleming, $(-20.5, -11)$ for Kursawe, and $(0, 14.5, -0.2)$ for Viennet..

		BT_{base} Algorithm 1		BT_{new} Algorithm 2	
Problem	LP Problem	P^N	norm. HV	P^N	norm. HV
Fonseca-Fleming	LP_{base}	100.00%	0.3349	100.00%	0.3349
	LP_{new}	100.00%	0.3395	100.00%	0.3395
Kursawe	LP_{base}	31.20%	0.4211	66.40%	0.4155
	LP_{new}	24.00%	0.4221	63.60%	0.4044
Viennet	LP_{base}	37.00%	0.7722	42.00%	0.7722
	LP_{new}	34.80%	0.7712	92.80%	0.7710

With respect to the global Pareto ratio, the real difference is made by BT_{new}. Indeed, independently on the LP problem used for computing the directions, the adoption of the new backtracking strategy guarantees a $P^N \simeq 65\%$; more than twice the value obtained using Algorithm Appendix A (see Table 1 and the approximated Pareto front/set in Figs. 8 and 9, respectively). We can explain this phenomenon looking at the orange dots in Fig. 7(b) and (d); the orange dots are points $x_j^{(k)}$ with respect to which the backtracking strategy cannot find a point along the direction $p^{*(k)}$ that is able to decrease all the objectives, but BT_{new} finds $x_j^{(k+1)}$ such that it is not dominated by $x_j^{(k)}$ and vice-versa; such a kind of points are Pareto critical points or points very near to them (see Fig. 6). In Fig. 7(b) and (d), we clearly see that there are paths of (almost) Pareto critical orange dots that (often) end with an approximated global Pareto optimal; i.e., BT_{new} permits the movement along trajectories made of Pareto critical points, increasing the probability of reaching the Pareto set/front of the problem. On the contrary, BT_{base} stops the iterations as soon as a Pareto critical point is reached, reducing the probability of finding a global Pareto optimal from a "non-suitable" starting point; an example is represented by the dominated red dots in Fig. 7(a) and (c).

We conclude the analysis of the Kursawe problem focusing again on Fig. 6. We observe that the Kursawe problem is characterized by $m = 2$ objective functions; therefore, for the observations in Remark 3.1 we have that the only difference between LP_{base} and LP_{new} is in the properties of the directions evaluated for the points that are not Pareto critical. For Pareto critical points, the only difference is in the norm of p^*, if it is not equal to the null vector.

5.2.3. Viennet

The last problem we consider is the Viennet problem. This problem is particularly difficult for MGD methods, because it is characterized by regions where the gradient g_1 of the first objective function is parallel and opposite to the gradient g_3 of the third objective function (i.e., $\bar{g}_1 = -\bar{g}_3$, see Fig. 10). Therefore, all these points are Pareto critical, and shared descent directions do not

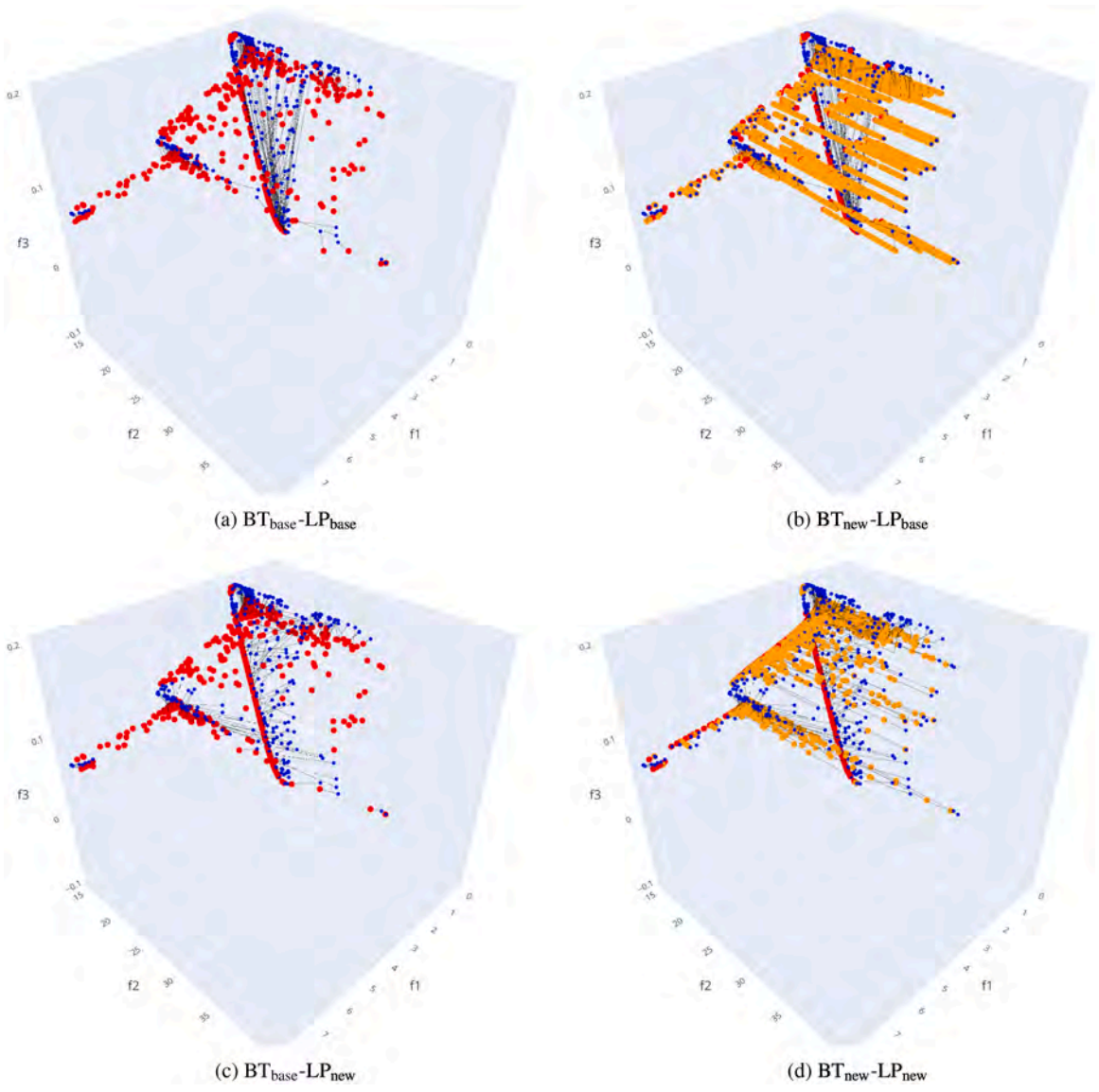


Fig. 11. Viennet test case. Movement of the $N = 500$ sequences in the objectives' space. The blue dots are the starting points' images $\boldsymbol{f}(\boldsymbol{x}_j^{(0)}) \in \mathbb{R}^3$, $j = 1, \dots, N$, while the red dots are the last points of the sequence. The orange dots are the critical points stored during Algorithm 4.2, before the final pruning (see the pseudocode). The black, dotted, and piece-wise linear curves describe the movement of each sequence from its starting point to its last element.

exist for the objectives in those points; at most, there are non-positive dot product directions perpendicular to $\boldsymbol{g}_1$ and $\boldsymbol{g}_3$ that are descent directions for the second objective function (e.g., see Fig. 4(c)).

Under these circumstances, we clearly see how the pairing of BT_{new} with LP_{new} is crucial for increasing the possibility of reaching the Pareto set/front of the problem, starting from a random point. Indeed, looking at the values of Table 1, we see that only the MGD method BT_{new}-LP_{new} reaches a good P^N value of 92.80%; all the other MGD methods reaches a value of P^N that is approximately between 35% and 45%. Nonetheless, some of the observations made for the Fonseca-Fleming and the Kursawe problems still hold. Specifically, we see that BT_{new} improves the performances in general, while the usage of LP_{base} or LP_{new} generates directions with different characteristics that, using Algorithm Appendix A as backtracking strategy, does not result in consistent differences in the performances. We can see the effects of these observations looking also at the approximated Pareto front/set in Figs. 12 and 14, respectively.

The performance improvements obtained using BT_{new} depend on the fact that this backtracking strategy helps to push the sequences along and/or beyond the "static regions" of Pareto critical points (see Fig. 10). This behavior is evident looking at the paths made of orange dots in Fig. 11(b) and (d) (objectives' space) and Fig. 13(b) and (d) (domain); in particular, see how the paths in Fig. 13(b) and (d) corresponds to the red regions in Fig. 10. On the other hand, since BT_{base} cannot help to overcome this kind of difficulty,

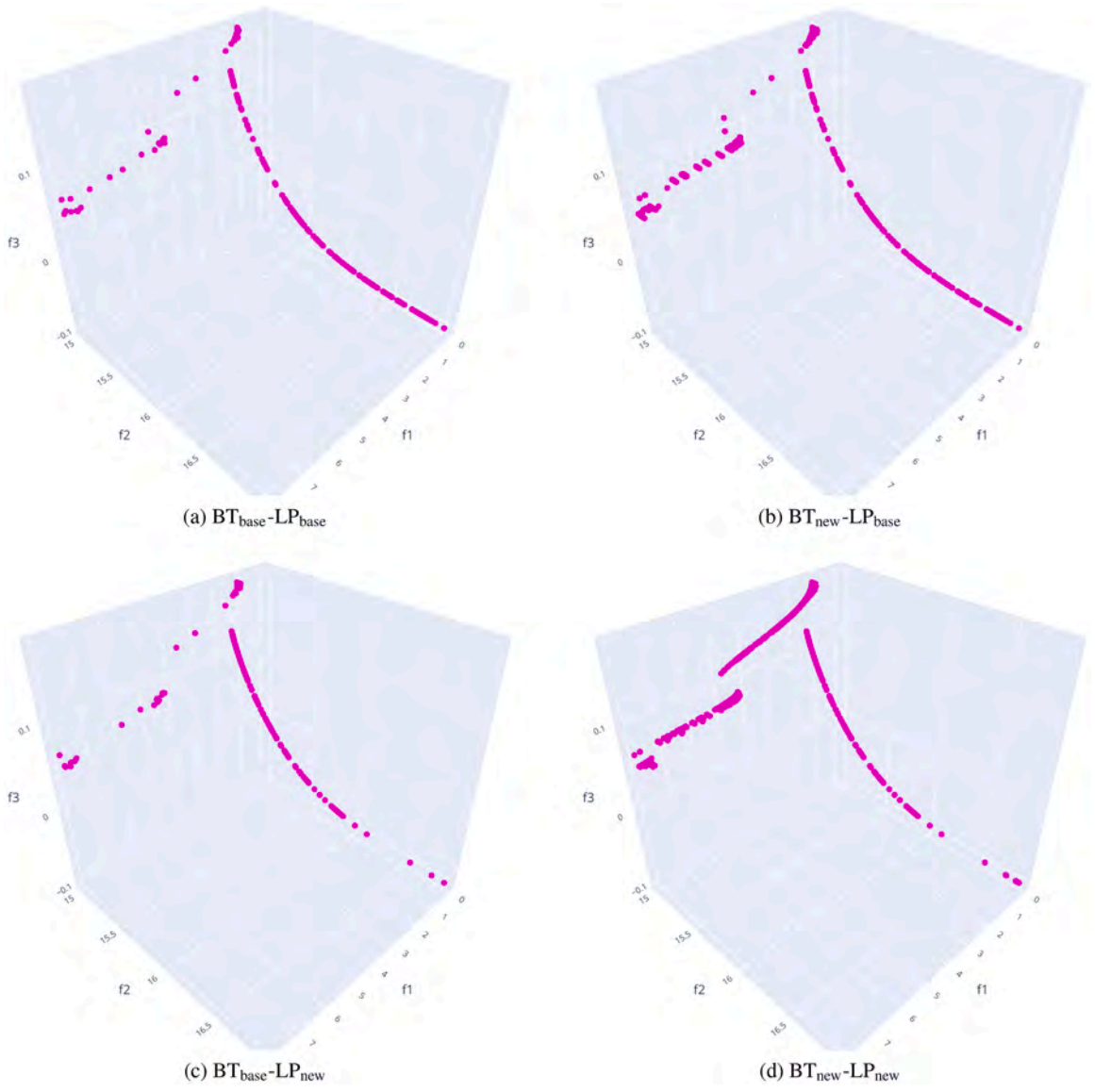


Fig. 12. Viennet test case. Images in the objectves' space of the non-dominated points among all the outputs returned by the MGD algorithms; i.e., images of the points in $\bigcup_{j=1}^{N} \widetilde{C}_j$ (see (30)).

instead of paths made of orange dots, for these regions we observe "frozen" red dots (i.e., sequences that stops at the beginning); see Fig. 11(a) and (c) (objectives' space) and Fig. 13(a) and (c) (domain). We observed a similar phenomenon in the Kursawe problem too, but less explicit.

Moreover, it is evident from Table 1 that the best performances are obtained when we use BT_{new} and the sequences are built with respect to LP_{new}. The reasons for these better performances depend on the different properties of the solutions returned by LP_{new} and the solutions returned by LP_{base}, with respect to the Pareto critical points belonging to the red regions illustrated in Fig. 10. Indeed, LP_{base} can return a null direction anytime the point of the sequence is Pareto critical (see Lemma 3.1), de-facto stopping the sequence itself; moreover, the infinite-norm of the computed direction is at most 1, generating steps that are not related to the gradients' order of magnitude (in this case, too small steps). On the other hand, LP_{new} returns a non-null direction for sure for all the points in the red regions illustrated in Fig. 10 where $\boldsymbol{g}_2 \neq \boldsymbol{0}$, because of Lemma 3.2 - item 2.3; moreover, the upper bound of the infinite-norm of the computed direction depends on the infinite-norms of the gradients of the objective functions (see (14)), granting longer steps when objectives are steeper.

From these observations, we deduce that the combination of LP_{new} with BT_{new} is particularly effective for the MOO problems like the Viennet problem, because it generates search directions and step sizes that prevent the sequences from stalling in the static regions of Pareto critical points. Indeed, LP_{new} does not return the null direction in these regions (as long as $\boldsymbol{g}_2 \neq \boldsymbol{0}$), ensuring that

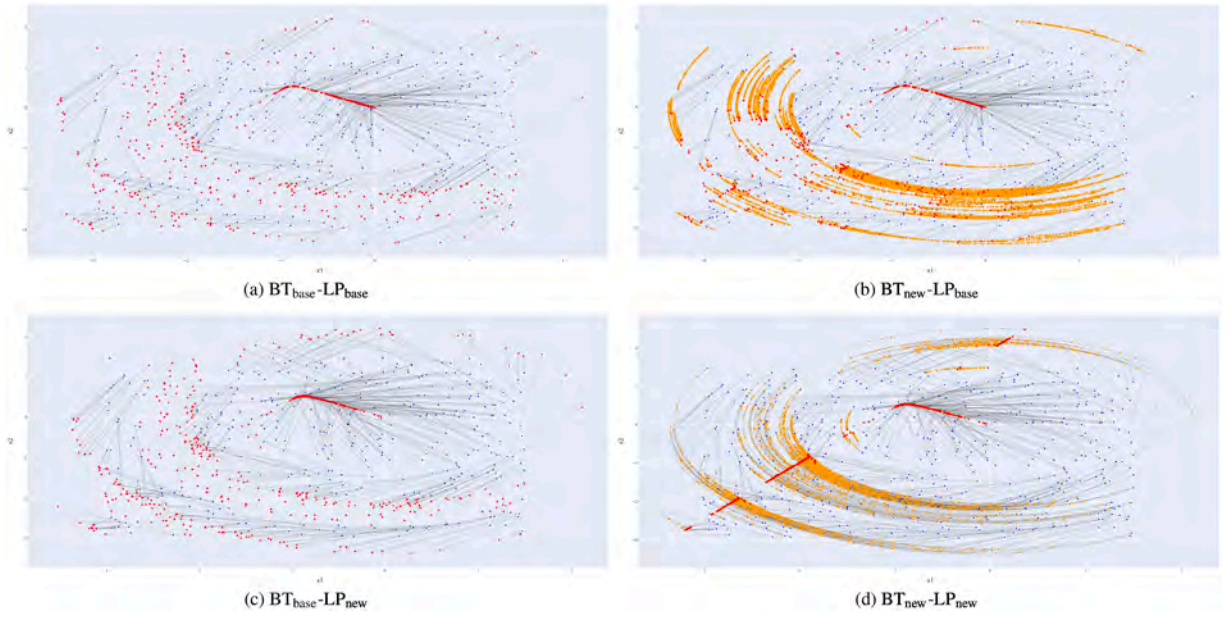


Fig. 13. Viennet test case. Movement of the $N = 500$ sequences. The blue dots are the starting points $\boldsymbol{x}_j^{(0)} \in \mathbb{R}^2$, $j = 1, \dots, N$, while the red dots are the last points of the sequence. The orange dots are the critical points stored during Algorithm 4.2, before the final pruning (see the pseudocode). The black, dotted, and piece-wise linear curves describe the movement of each sequence from its starting point to its last element.

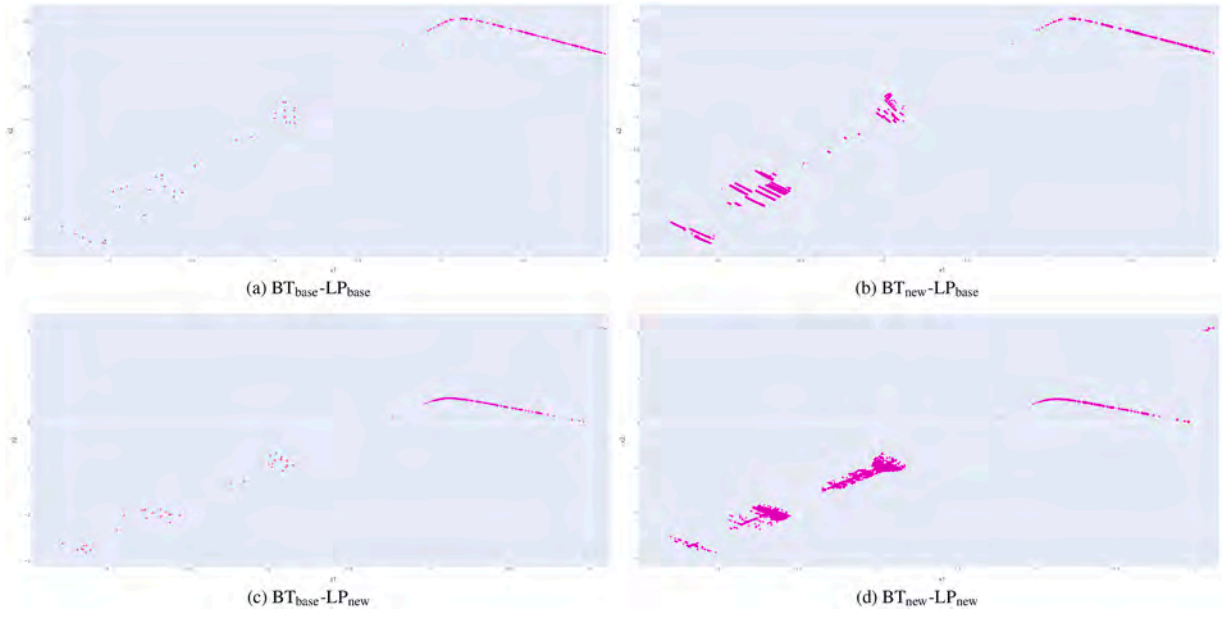


Fig. 14. Viennet test case. Non-dominated points among all the outputs returned by the MGD algorithms; i.e., the points in $\bigcup_{j=1}^{N} \widetilde{C}_j$ (see (30)).

the iterations continue to move. As a consequence, the resulting sequences are able to progress along the critical regions, instead of stopping at their boundaries, thereby increasing the probability of reaching a good global Pareto approximation. See how the paths generated by orange/critical Points in Fig. 13(d) (but also Fig. 13(b)) follow the circular shapes of the red regions of Fig. 10, while in the objectives' space their movement actually results mostly in a reduction of the second objective function (see Fig. 11(d), but also Fig. 11(b)).

6. Conclusion

In this paper, we introduced LP_{new}, a novel LP problem specifically designed for computing directions in MGD methods. This new LP problem is derived from LP_{base}, introduced by Fliege and Svaiter in [21]. Through rigorous theoretical analysis (see Lemma 3.2), we demonstrated that our proposed LP formulation surely returns non-null directions if the point $\boldsymbol{x}$ of the sequence is Pareto critical but under particular conditions; namely, when at $\boldsymbol{x}$ there is at least one non-positive dot product direction that is a descent direction for at least one objective function. This is one of the main properties that distinguishes the new LP problem from the one in [21], which can return null directions anytime the point of the sequence we consider is a Pareto critical point.

Additionally, we developed a new backtracking strategy for MGD methods (see BT_{new}). This strategy is characterized by the acceptance of a new point $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \widehat{\eta}\, \boldsymbol{p}^{*(k)}$ at the last backtracking step if it is non-dominated by $\boldsymbol{x}^{(k)}$, even if the Armijo condition is not satisfied for all objectives. Furthermore, we introduced a "storing property" within this strategy (see Algorithm 2), ensuring that all points $\boldsymbol{x}^{(k)}$ which do not dominate $\boldsymbol{x}^{(k+1)}$, and vice-versa, are stored. This innovation aims to improve the efficiency and robustness of the backtracking process, improving the probability of any sequence to reach the Pareto set. We provided theoretical proof of the convergence properties for a MGD method that incorporates our new backtracking strategy.

To validate our theoretical findings, we conducted numerical experiments to evaluate the performance of the new methods compared to the baseline method taken from [21]. The numerical experiments confirmed the theoretical findings and indicated that the new backtracking strategy consistently improves performance over the baseline BT_{base}. Independently of the LP problem used to compute directions, adopting BT_{new} significantly increases the probability that a sequence reaches the Pareto set, reducing premature stagnation near non-optimal Pareto critical points.

The experiments also clarify the role of the new LP problem. While LP_{new} does not uniformly outperform the formulation in [21], the difference between the two becomes substantial when the objective landscape contains large and diffuse regions of Pareto critical points. In such situations, exemplified by the Viennet problem, combining LP_{new} with the new backtracking strategy produces search directions that prevent stalling and enable sequences to move along or beyond these "static regions". This leads to higher probabilities of reaching good global Pareto approximations, whereas using LP_{base} may still result in early stagnation. Consequently, MGD methods based on BT_{new} are preferable to their BT_{base} counterparts, and LP_{new} becomes particularly advantageous for problems where extended sets of Pareto critical points are expected.

Future work will focus on extending our new methods to constrained MOO and on applying them to real-world problems. This will enable further validation of their practical utility and potential for broader adoption in various application domains.

Code availability

The code for MGD methods illustrated in this paper is available at: https://github.com/Fra0013To/MGD.

Data availability

No data was used for the research described in the article.

Acknowledgement

This study was carried out within the FAIR-Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR)-MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3—D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions; neither the European Union nor the European Commission can be considered responsible for them.

Appendix A. Strictly Decreasing Backtracking for MGD

In this appendix section, we report the pseudocode of the standard MGD algorithm used in Section 5; i.e., the algorithm that implements a backtracking strategy looking for a decreasing sequence for all the objectives. Actually, this algorithm is equivalent to a simplified version of Algorithm 1, where the algorithm stops if the Armijo condition A_i^t is not satisfied for all the objectives, for all $t = 0, \dots, \Theta$.

Algorithm 3 Let us consider the unconstrained MOO problem (1). Then, we define the following MGD algorithm for the implementation of the classic descent method based on a backtracking strategy looking for a decreasing sequence for all the objectives.

Data: $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ starting point for (5); $c_1, \alpha \in (0,1)$ parameters for the Armijo condition; η_0 starting value for the step length; $\Theta \in \mathbb{N}$ maximum number of backtracking steps; $K \in \mathbb{N}$ maximum number of iterations; $\mathcal{P}$ sub-problem for computing $\boldsymbol{p}^*$ feasible direction with respect to $\boldsymbol{x}$ (e.g., an LP problem like LP_{base} and LP_{new}).

Procedure:

for $k = 0, \dots, (K-1)$ **do**
  $\boldsymbol{p}^{*(k)} \leftarrow$ solve $\mathcal{P}$ with respect to $\boldsymbol{x}^{(k)}$ ▷ e.g., solve $\mathcal{P}$ =LP_{base} or $\mathcal{P}$ =LP_{new} for finding $\boldsymbol{p}^{*(k)}$
  $\eta^{(k)} \leftarrow \eta_0$
  for $t = 0, \dots, (\Theta-1)$ **do**
    if A_i^t is true for each $i = 1, \dots, m$ **then**
      break
    else
      $\eta^{(k)} \leftarrow \eta^{(k)}\alpha$
    end if
  end for
  if $\eta^{(k)} = \eta_0\alpha^{\Theta}$ and there is $i \in \{1, \dots, m\}$ s.t. A_i^t is false **then**
    break
  else
    $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} + \eta^{(k)}\boldsymbol{p}^{*(k)}$
  end if
end for
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(k)}$
return : $\hat{\boldsymbol{x}}$

References

[1] Multi-objective optimisation of herringbone grooved gas bearings supporting a high speed rotor, taking into account rarefied gas and real gas effects, vol. Volume 3: Dynamic Systems and Controls, Symposium on Design and Analysis of Advanced Structures, and Tribology Engineering Systems Design and Analysis, 2006. https://doi.org/10.1115/ESDA2006-95085

[2] K.S. Zadeh, Multi-objective optimization in variably saturated fluid flow, J. Comput. Appl. Math. 223 (2009) 801–819. https://doi.org/10.1016/j.cam.2008.03.005

[3] F. Colombo, F. Della Santa, S. Pieraccini, Multi-objective optimisation of an aerostatic pad: design of position, number and diameter of the supply holes, J. Mech. 36 (3) (2019) 347–360. https://doi.org/10.1017/jmech.2019.41

[4] Y. Cui, Z. Geng, Q. Zhu, Y. Han, Review: multi-objective optimization methods and application in energy saving, Energy 125 (2017) 681–704. https://doi.org/10.1016/j.energy.2017.02.174

[5] W. Wu, J. Guo, J. Li, H. Hou, Q. Meng, W. Wang, A multi-objective optimization design method in zero energy building study: a case study concerning small mass buildings in cold district of china, Energy Build 158 (2018) 1613–1624. https://doi.org/10.1016/j.enbuild.2017.10.102

[6] L. Chen, W.L. Liu, J. Zhong, An efficient multi-objective ant colony optimization for task allocation of heterogeneous unmanned aerial vehicles, J. Comput. Sci. 58 (2022). https://doi.org/10.1016/j.jocs.2021.101545

[7] K. Miettinen, Nonlinear Multiobjective Optimization, Springer New York, NY, 1998. https://doi.org/10.1007/978-1-4615-5563-6

[8] M. Ehrgott, Multicriteria Optimization, Springer Berlin, Heidelberg, 2005. https://doi.org/10.1007/3-540-27659-9

[9] M. Ehrgott, M.M. Wiecek, Mutiobjective Programming, 2005, pp. 667–708. https://doi.org/10.1007/0-387-23081-5_17

[10] X.-S. Yang, Chapter 15 - multi-objective optimization, in: X.-S. Yang (Ed.), Nature-Inspired Optimization Algorithms (Second Edition), Academic Press, second edition edition, 2021, pp. 221–237. https://doi.org/10.1016/B978-0-12-821986-7.00022-6

[11] J. Larson, M. Menickelly, S.M. Wild, Derivative-free optimization methods, Acta Numer. 28 (2019) 287–404. https://doi.org/10.1017/S0962492919000060

[12] M. Mitchell, Elements of Generic Algorithms - An Introduction to Generic Algorithms, 1998. https://mitpress.mit.edu/books/introduction-genetic-algorithms.

[13] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, 4, 1995, pp. 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968

[14] X.-S. Yang, Chapter 8 - particle swarm optimization, in: X.-S. Yang (Ed.), Nature-Inspired Optimization Algorithms (Second Edition), Academic Press, second edition edition, 2021, pp. 111–121. https://doi.org/10.1016/B978-0-12-821986-7.00015-9

[15] X.-S. Yang, Chapter 6 - genetic algorithms, in: X.-S. Yang (Ed.), Nature-Inspired Optimization Algorithms (Second Edition), Academic Press, second edition edition, 2021, pp. 91–100. https://doi.org/10.1016/B978-0-12-821986-7.00013-5

[16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEE Trans. Evol. Comput. 6 (2002) 182–197. https://doi.org/10.1016/j.ejor.2017.07.027

[17] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia Tools Appl. 80 (2021) 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

[18] G. Xu, Y.Q. Yang, B.B. Liu, Y.H. Xu, A.J. Wu, An efficient hybrid multi-objective particle swarm optimization with a multi-objective dichotomy line search, J. Comput. Appl. Math. 280 (2015) 310–326. https://doi.org/10.1016/j.cam.2014.11.056

[19] A. Pascoletti, P. Serafini, Scalarizing vector optimization problems, J. Optim. Theory Appl. 42 (1984) 499–524. https://doi.org/10.1007/BF00934564

[20] E. Khorram, K. Khaledian, M. Khaledyan, A numerical method for constructing the pareto front of multi-objective optimization problems, J. Comput. Appl. Math. 261 (2014) 158–171. https://doi.org/10.1016/j.cam.2013.11.007

[21] J. Fliege, B.F. Svaiter, Steepest descent methods for multicriteria optimization, Math. Methods Oper. Res. 51 (3) (2000) 479–494. https://doi.org/10.1007/s001860000043

[22] S. Schäffler, R. Schultz, K. Weinzierl, Stochastic method for the solution of unconstrained vector optimization problems, J. Optim. Theory Appl. 114 (1) (2002) 209–222. https://doi.org/10.1023/A:1015472306888

[23] J.A. Désidéri, Multiple-Gradient Descent Algorithm (MGDA), Technical Report, HAL inria, 2009. https://inria.hal.science/inria-00389811.

[24] J.A. Désidéri, MGDA II: A Direct Method for Calculating a Descent Direction Common to Several Criteria, Technical Report April, 2012. https://inria.hal.science/hal-00685762.
[25] J.A. Désidéri, Multiple-gradient descent algorithm (MGDA) for multiobjective optimization, C.R. Math. 350 (5–6) (2012) 313–318. https://doi.org/10.1016/j.crma.2012.03.014
[26] S. Peitz, M. Dellnitz, Gradient-based multiobjective optimization with uncertainties, Stud. Comput. Intell. 731 (2018) 159–182. https://doi.org/10.1007/978-3-319-64063-1_7
[27] L. Armijo, Minimization of functions having lipschitz continuous first partial derivatives, Pac. J. Math. 16 (1) (1966) 1–3. https://doi.org/10.2140/pjm.1966.16.1
[28] P. Wolfe, Convergence conditions for ascent methods, SIAM Rev. 11 (2) (1969) 226–235. https://doi.org/10.1137/1011036
[29] P. Wolfe, Convergence conditions for ascent methods. II: some corrections, SIAM Rev. 13 (2) (1971) 185–188. https://doi.org/10.1137/1013035
[30] J. Nocedal, S.J. Wright, Advances in Industrial Control, Numerical Optimization, 9781447122234, Springer, 2nd edition, 2012. https://doi.org/10.1007/978-1-4471-2224-1_2
[31] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput. 3 (1995) 1–16. https://doi.org/10.1162/evco.1995.3.1.1
[32] F. Kursawe, A Variant of Evolution Strategies for Vector Optimization, Springer-Verlag, (2025), pp. 193–197. https://doi.org/10.1007/BFb0029752
[33] R. Viennet, C. Fonteix, I. Marc, Multicriteria optimization using a genetic algorithm for determining a pareto set, Int. J. Syst. Sci. 27 (1996) 255–260. https://doi.org/10.1080/00207729608929211
[34] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2000) 173–195. https://doi.org/10.1162/106365600568202
[35] C. Audet, J. Bigeon, D. Cartier, S.L. Digabel, L. Salomon, Performance indicators in multiobjective optimization, Eur. J. Oper. Res. 292 (2021) 397–422. https://doi.org/10.1016/j.ejor.2020.11.016
[36] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation, IEEE Trans. Syst. Man Cybern. Part A Syst. Humans 28 (1998) 26–37. https://doi.org/10.1109/3468.650319