An adaptive line-search-free multiobjective gradient method and its iteration-complexity analysis

Max L.N. Gonçalves[1], Geovani N. Grapiglia[2], Jefferson G. Melo[1]

[1]IME, Universidade Federal de Goiás, Rua Jacarandá, Goiânia, CEP 74001-970, GO, Brazil, Emails: maxlng@ufg.br and jefferson@ufg.br.
[2]ICTEAM/INMA, Université catholique de Louvain, Avenue Georges Lemaître, 4-6, L4.05.01, Louvain-la-Neuve, B-1348, Belgium, Email: geovani.grapiglia@uclouvain.be.

Abstract

This work introduces an Adaptive Line-Search-Free Multiobjective Gradient (AMG) method for solving smooth multiobjective optimization problems. The proposed approach automatically adjusts stepsizes based on steepest descent directions, promoting robustness with respect to stepsize choice while maintaining low computational cost. The method is specifically tailored to the multiobjective setting and does not rely on function evaluations, making it well suited for this scenario. The proposed algorithm admits two variants: (i) a conservative variant, in which the stepsize is monotonically decreasing; and (ii) a flexible variant, which allows occasional increases in the stepsize. From a theoretical standpoint, under standard Lipschitz continuity assumptions on the gradients, we establish iteration-complexity bounds for achieving a Pareto critical point for both variants in the nonconvex setting. In the convex setting, we further derive improved iteration-complexity bounds for the conservative AMG variant. From a practical standpoint, the numerical experiments demonstrate that the flexible AMG performs favorably compared to the steepest descent method with either a fixed stepsize or Armijo line search.

Keywords: Adaptive gradient method; iteration-complexity; Pareto optimality; multiobjective problem.

1 Introduction

In this paper, we consider the following unconstrained multiobjective optimization problem:

$$\min_{x\in\mathbb{R}^n} F(x) := (F_1(x), \ldots, F_m(x)), \tag{1}$$

where, for each $j \in \mathcal{J} := \{1, \ldots, m\}$, the component function $F_j : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, and its gradient ∇F_j is L_j-Lipschitz continuous on $\mathbb{R}^n$. The latter assumption implies, for all $x, y \in \mathbb{R}^n$, that

$$F_j(y) \le F_j(x) + \langle \nabla F_j(x), y - x\rangle + \frac{L_j}{2}\|x - y\|^2. \tag{2}$$

For convenience, we define $L := \max_{j\in\mathcal{J}} L_j$.

Problem (1) provides a unifying modeling framework for decision-making processes involving multiple, often conflicting, performance criteria. Such situations arise naturally in a wide range of real-world applications, including engineering design problems balancing cost, reliability, and efficiency, economic and management models involving trade-offs between risk and return, and medical and biological applications where treatment effectiveness must be weighed against side effects or resource constraints; see, for instance, [10, 26]. The intrinsic presence of competing objectives makes multiobjective optimization fundamentally different from its scalar counterpart and highlights the need for specific algorithmic tools. This practical relevance has motivated extensive research over the past decades on the development of algorithms for general multiobjective and vector optimization problems. A large portion of this literature focuses on extending well-established methods from scalar optimization ($\mathcal{J} = \{1\}$) to the multiobjective setting. In particular, multiobjective variants of steepest descent, conjugate gradient, conditional (FrankWolfe) gradient, projected gradient, proximal gradient, and Newton-type methods have been proposed and analyzed; see, for example, [2, 4–6, 8, 11–16, 18–21, 24] and the references therein.

In recent years, there has been growing interest in adaptive first-order methods that avoid line-search procedures and reduce sensitivity to problem-dependent parameters. Such methods are particularly attractive in applications, where function evaluations are costly and suitable stepsizes are difficult to estimate due to unknown or heterogeneous smoothness constants across objectives. In the scalar case, adaptive gradient methodsmost notably AdaGrad and its variants (see, for example, [9, 25, 27])have proven highly effective by automatically adjusting stepsizes based on accumulated gradient information, leading to robust practical performance with minimal parameter tuning.

Motivated by these developments, and inspired in particular by the AdaGrad methodology and the adaptive nonmonotone stepsize strategy proposed in [22], we propose and analyze an adaptive line-search-free multiobjective gradient (AMG) method for solving problem (1). The proposed AMG method preserves key advantages of adaptive gradient schemes. More precisely: (i) the method converges for any choice of algorithmic hyperparameters, in contrast to the steepest descent method with a fixed stepsize, which may fail to converge if the stepsize exceeds a threshold depending on the (typically unknown) smoothness Lipschitz constant; and (ii) AMG is free of function evaluationsan especially valuable property in multiobjective optimization, where evaluating the objective function requires computing multiple components. Our algorithm admits two variants: a conservative version with monotonically decreasing stepsizes and a flexible version that occasionally allows stepsize increases, potentially accelerating convergence in practice. The inclusion of the flexible case, however, makes the complexity analysis substantially more involved. While the conservative variant correponds to a multiobjective generalization of the scalar AdaGrad-Norm method, the flexible variant constitutes a novel contribution to the AdaGrad literature, even in the scalar case ($\mathcal{J} = \{1\}$).

From a theoretical standpoint, we establish iteration-complexity guarantees for both variants in the nonconvex setting. Specifically, for a given tolerance $\varepsilon > 0$, the conservative variant requires at most $\mathcal{O}(\varepsilon^{-2})$ iterations (or evaluations of the steepest descent direction) to reach an $\varepsilon-$approximate Pareto critical point (see Remark 1(a)), whereas the flexible variant attains a slightly weaker bound of $\mathcal{O}(\varepsilon^{-2}\log^3(\varepsilon^{-1}))$. In the convex setting, we further show that the conservative variant enjoys an improved iteration-complexity bound of $\mathcal{O}(\varepsilon^{-1})$ iterations to reach an $\varepsilon-$approximate Pareto optimal point, measured by a merit function recently introduced in [17]. From a practical perspective, numerical experiments demonstrate that the flexible AMG performs favorably compared to the multiobjective steepest descent method [12] with either a fixed stepsize or Armijo line search.

The paper is organized as follows. Section 2 contains some preliminaries results of the present work. Section 3 formally describes the AMG method to solve (1), whereas its iteration-complexity bounds for the nonconvex and convex setting are discussed in Section 4. Section 5 contains some numerical experiments. Final remarks are given in Section 6.

2 Preliminaries

This section presents basic properties of the multiobjective steepest descent direction and a key auxiliary result used in the analysis of the proposed method.

In multiobjective optimization, the concept of optimality is replaced by Pareto optimality. A point $x^* \in \mathbb{R}^n$ is considered Pareto optimal if there is no other point $x \in \mathbb{R}^n$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$, where the inequality $\leq$ is understood component-wise. Similarly, a point $x^* \in \mathbb{R}^n$ is called weakly Pareto optimal if there is no point $x \in \mathbb{R}^n$ such that $F(x) < F(x^*)$. A point $x^* \in \mathbb{R}^n$ is locally Pareto optimal (or locally weakly Pareto optimal) if there is a neighborhood $V \subset \mathbb{R}^n$ around x^* where x^* is Pareto optimal (or weakly Pareto optimal) for F restricted to V. A necessary condition for local Pareto optimality of $x^* \in \mathbb{R}^n$ is given by:

$$\max_{j \in \mathcal{J}} \langle \nabla F_j(x^*), y - x^* \rangle \geq 0, \quad \forall y \in \mathbb{R}^n. \tag{3}$$

A point $x^* \in \mathbb{R}^n$ that satisfies this condition is called Pareto critical or stationary. Consequently, if a point x is not Pareto critical, there exists a vector $y \in \mathbb{R}^n$ such that $y - x$ is a descent direction for F at x, meaning that there exists $\varepsilon > 0$ such that $F(x + t(y - x)) < F(x)$ for any $t \in (0, \varepsilon]$.

For a given point $x \in \mathbb{R}^n$, consider the scalar-valued problem:

$$\min_{d\in\mathbb{R}^n} \max_{j\in\mathcal{J}} \langle \nabla F_j(x), d\rangle + \frac{1}{2}\|d\|^2. \tag{4}$$

Since $\max_{j\in\mathcal{J}} \langle \nabla F_j(x), \cdot\rangle$ is a real closed convex function, it follows that (4) has always a unique optimal solution. Denote by $d_{SD}(x)$ the solution of (4) and by $\theta_{SD}(x)$ its optimal value, i.e.,

$$d_{SD}(x) := \arg\min \left\{ \max_{j\in\mathcal{J}} \langle \nabla F_j(x), d\rangle + \frac{\|d\|^2}{2} \mid d \in \mathbb{R}^n \right\}, \tag{5}$$

and

$$\theta_{SD}(x) := \max_{j\in\mathcal{J}} \langle \nabla F_j(x), d_{SD}(x)\rangle + \frac{1}{2}\|d_{SD}(x)\|^2. \tag{6}$$

Direction $d_{SD}(x)$ extends the notion of the steepest descent direction to the multiobjective optimization case. Note that in the single-objective minimization case where $F\colon \mathbb{R}^n \to \mathbb{R}$, we obtain $d_{SD}(x) = -\nabla F(x)$ and $\theta_{SD}(x) = -\|\nabla F(x)\|^2/2$. As is well-known, the direction $d_{SD}(x)$ and the optimum value $\theta_{SD}(x)$ can be used, in particular, to characterize Pareto points of (1).

Lemma 1 ([21, Lemma 3.3]) Let $d_{SD} : \mathbb{R}^n \to \mathbb{R}^n$ and $\theta_{SD} : \mathbb{R}^n \to \mathbb{R}$ be as in (5). Then, we have:

(a) if x is Pareto critical, then $d_{SD}(x) = 0$ and $\theta_{SD}(x) = 0$;
(b) if x is not Pareto critical, then we have

$$d_{SD}(x) \neq 0, \quad \theta_{SD}(x) < 0, \quad \max_{j\in\mathcal{J}} \langle \nabla F_j(x), d_{SD}(x)\rangle \leq -\frac{1}{2}\|d_{SD}(x)\|^2 < 0; \tag{7}$$

as a consequence, $d_{SD}(x)$ is a descent direction for F at x;
(c) the mappings $d_{SD}(\cdot)$ and $\theta_{SD}(\cdot)$ are continuous.

Problem (4) can be reformulated as

$$\begin{array}{ll} \min\limits_{(t,d)\in\mathbb{R}\times\mathbb{R}^n} & t + \frac{1}{2}\|d\|^2 \\ \text{s. t.} & \langle \nabla F_j(x), d\rangle \leq t, \quad \forall j \in \mathcal{J}, \end{array} \tag{8}$$

which is a convex quadratic problem with linear inequality constraints. Since problem (8) has a unique solution $(f(x, d_{SD}(x)), d_{SD}(x))$ and its constraints are linear, there exists a multiplier $\lambda^{SD}(x) \in \mathbb{R}^m$ such that the triple $(t, d, \lambda) := (f(x, d_{SD}(x)), d_{SD}(x), \lambda^{SD}(x)) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ satisfies the Karush-Kuhn-Tucker conditions of problem (8) given by:

$$\sum_{j=1}^{m} \lambda_j \left[\nabla F_j(x) + d\right] = 0, \quad \sum_{j=1}^{m} \lambda_j = 1,$$

$$\lambda_j \geq 0, \quad \langle \nabla F_j(x), d \rangle \leq t, \quad \lambda_j \left[\langle \nabla F_j(x), d \rangle - t\right] = 0, \quad \forall j = 1, \ldots, m.$$

Therefore, in particular, we have

$$d_{SD}(x) = -\sum_{j=1}^{m} \lambda_j^{SD}(x) \nabla F_j(x), \tag{9}$$

$$\sum_{j=1}^{m} \lambda_j^{SD}(x) = 1, \quad \lambda_j^{SD}(x) \geq 0, \quad \forall j \in \mathcal{J}, \tag{10}$$

$$\theta_{SD}(x) = -\frac{1}{2}\|d_{SD}(x)\|^2 \quad \text{and} \quad \max_{j \in \mathcal{J}} \langle \nabla F_j(x), d_{SD}(x) \rangle = -\|d_{SD}(x)\|^2. \tag{11}$$

We conclude this section by presenting, for completeness, a result analogous to that in [27, Lemma 3.2], which addresses the convergence properties of the scalar AdaGrad-Norm method.

Lemma 2 For any positive numbers $a_1, \ldots, a_k$, we have

$$\sum_{i=1}^{k} \frac{a_i}{1 + \sum_{t=1}^{i} a_t} \leq \log\left(1 + \sum_{i=1}^{k} a_i\right).$$

Proof Define $a_0 = 1$ and note that since $\{a_i\}_{i=0}^k$ are positive numbers, the partial sums $S_i := \sum_{t=0}^{i} a_t$, for $i = 0, \ldots, k$, are strictly increasing and $S_0 = a_0 = 1$. Thus, $\log S_0 = 0$ and the following relations hold

$$\log S_k = \log S_k - \log S_0 = \sum_{i=1}^{k} \left(\log S_i - \log S_{i-1}\right).$$

Apply, for $i = 1, \ldots, k$, the mean value theorem to the function $f(x) = \log x$ on the interval $[S_{i-1}, S_i]$, and noting that $S_i = S_{i-1} + a_i$ and $f'(x) = 1/x$, we have that there exists $\xi_i \in (S_{i-1}, S_i)$ such that

$$\log S_i - \log S_{i-1} = \frac{1}{\xi_i}(S_i - S_{i-1}) = \frac{a_i}{\xi_i}.$$

Since $\xi_i < S_i$ and $a_i > 0$, we have

$$\frac{a_i}{S_i} < \frac{a_i}{\xi_i} = \log S_i - \log S_{i-1}.$$

Summing over $i = 1$ to k, we get

$$\sum_{i=1}^{k} \frac{a_i}{S_i} \leq \sum_{i=1}^{k} (\log S_i - \log S_{i-1}) = \log S_k - \log S_0 = \log S_k.$$

Therefore, the desired result follows from the fact that $S_i = 1 + \sum_{t=1}^{i} a_t$, for each $i = 1, \ldots, k$. □

3 An adaptive line-search-free multiobjective gradient method

In this section, we present the adaptive line-search-free multiobjective gradient method for solving problem (1) and provide remarks on its main features.

The algorithm is formally stated next.

Algorithm 1: AMG method.

(0) Let $x^0 \in \mathbb{R}^n$, $\omega_0 > 0$, $b_{\max} \geq b_0 \geq b_{\min} > 0$, $\alpha \in [0, 1)$, and $\eta > 0$ be given. Set $k \leftarrow 0$.

(1) Compute $d_{SD}(x^k)$ as in (5), and set $d^k := d_{SD}(x^k)$.

(2) If $d^k = 0$, then stop and return x^k.

(3) Update b_{k+1} and ω_{k+1} as follows:

$$(b_{k+1}, \omega_{k+1}) = \begin{cases} (\hat{b}_k, \|d^k\|), & \text{if } k \geq 1 \ \text{ and } \ \|d^k\| \leq \alpha\omega_k, \\ \left(\sqrt{b_k^2 + \|d^k\|^2}, \omega_k\right), & \text{otherwise,} \end{cases} \tag{12}$$

where

$$\hat{b}_k \in [b_{\min}, \min\{b_k, b_{\max}\}]\,; \tag{13}$$

(4) Compute

$$x^{k+1} := x^k + \frac{\eta}{b_{k+1}} d^k. \tag{14}$$

Set $k \leftarrow k + 1$ and return to Step (1).

Remark 1 (a) If the algorithm terminates at some iteration k, then Lemma 1 ensures that x^k is a Pareto critical point of problem (1). For a given tolerance $\varepsilon > 0$, our theoretical goal will be to establish a bound on the number of iterations (or evaluations of the steepest descent direction) required to obtain an εapproximate Pareto critical point, that is, an iterate x^k satisfying $\|d^k\| = (2|\theta^k|)^{\frac{1}{2}} \leq \varepsilon$.

(b) Note that the stepsize in (14) is given by η/b_{k+1}. Hence, Algorithm 1 consists of two types of iterations: one with decreasing stepsize, where $b_{k+1}^2 = b_k^2 + \|d^k\|^2 > b_k^2$, and another with nondecreasing stepsize, where $b_{k+1} = \hat{b}_k \leq b_k$. The former case is regarded as conservative, and the latter as flexible. This nonmonotone stepsize rule is inspired by the one proposed in [22] in the context of trust-region methods.

(c) The sequence $\{\omega_k\}$ is used to identify when a flexible iteration occurs: it remains constant throughout a cycle of conservative iterations and decreases during a cycle of flexible ones. In particular, $\{\omega_k\}$ is nonincreasing and satisfies $\omega_k \leq \omega_0$ for all $k \in \mathbb{N}$.

(d) The parameter α is used to control the type of iterations performed by the method. It is worth noting that when $\alpha = 0$, the method executes only conservative iterations.

(e) Algorithm 1 does not prescribe a specific rule for selecting the stepsize $\hat{b}_k$ in (13). An effective and practical choice is given by

$$\hat{b}_k = \begin{cases} \min\{b_{\max}, \max\{b_{\min}, b_k/2\}\}, & \text{if } \|d^k\| > \frac{\eta}{2b_k}, \\ \min\{b_{\max}, b_k\}, & \text{otherwise.} \end{cases} \tag{15}$$

with $b_{\max} := \|d^0\|$. This update rule guarantees that (13) holds. Moreover, it adapts the magnitude of $\hat{b}_k$and consequently of b_{k+1}based on the relationship between the stepsize η/b_k and the norm of the descent direction d^k.

We next proceed to the iteration-complexity analysis of Algorithm 1.

4 Iteration-complexity analysis of Algorithm 1

In this section, we analyze the iteration-complexity of the AMG method. Our goal is to quantify the number of iterations required to reach approximate Pareto critical point of problem (1). In the nonconvex case, we establish worst-case complexity bounds for both the conservative and flexible variants of AMG. In the convex case, we show that the conservative variant enjoys improved iteration-complexity guarantees.

4.1 Nonconvex case

We first consider the general nonconvex setting, where no convexity assumptions are imposed on the component functions F_j.

Given $\ell \geq 0$, define the set of indices

$$D_\ell := \left\{k \in \{1, 2, ..., \ell\} : \|d^k\| \leq \alpha\omega_k\right\} = \{k_1, k_2, \dots, k_{|D_\ell|}\}, \tag{16}$$

where $|A|$ denotes the cardinality of a set $A \subset \mathbb{N}$, and, by convention, $|\emptyset| = 0$. This set is used to track when a flexible iteration is performed up to the ℓ-th iteration. Note that $I_0 := \{0, 1, \dots, k_1\}$ is the first cycle of conservative iterations. Moreover, for every t such that $k_{t+1} > k_t + 1$, we have a new cycle of conservative iterations $I_t := \{k_t + 2, \dots, k_{t+1}\}$.

The following result shows that the sequence $\{d^{k_i}\}$ decreases geometrically with a rate α. This result will be used subsequently to estimate the size of the set D_ℓ.

Lemma 3 Let $\{(x^k, d^k, \omega_k)\}_{k\geq 0}$ be generated by Algorithm 1. Suppose that $D_\ell \neq \emptyset$ for some $\ell \geq 1$, and let $D_\ell = \{k_i\}_{i=1}^{|D_\ell|}$ with $k_i < k_j$ whenever $i < j$. Then, the following inequality holds

$$\|d^{k_i}\| \leq \alpha^i\omega_0, \quad \forall i = 1, \dots, |D_\ell|.$$

Proof From the definitions of ω_k and D_l in (12) and (16), respectively, we have

$$\begin{aligned} \omega_k &= \omega_0 \quad \text{for } k = 0, \dots, k_1, \\ \omega_k &= \|d^{k_i}\| \quad \text{for } k = k_i + 1, \dots, k_{i+1}, \\ \omega_{k_i+1} &= \|d^{k_i}\| \leq \alpha\omega_{k_i} \quad \text{for } i = 1, \dots, |D_\ell|. \end{aligned} \tag{17}$$

It follows from the first relation and the last inequality above that the statement of the lemma holds for $i = 1$. Now assume that the statement of the lemma holds for some $i = 1, \dots, |D_l| - 1$. It follows from the last two relations in (17) that

$$\omega_{k_{i+1}+1} = \|d^{k_{i+1}}\| \leq \alpha\omega_{k_{i+1}} = \alpha\|d^{k_i}\| \leq \alpha^{i+1}\omega_0,$$

where the last inequality is due to the induction assumption. Therefore, the proof of the lemma follows by an induction argument. □

The following result provides, in particular, two estimates. The first one is an upper bound on $|D_\ell|$, and the second one estimates the number of iterations ℓ in terms of $|D_\ell|$ and the sizes of the conservative cycles $|I_t|$, $t = 0, \dots, |D_\ell|$. Both estimates assume that $\|d^k\|$, for $k = 0, 1, \dots, \ell$, are not sufficiently small.

Lemma 4 Let $\{(x^k, d^k)\}_{k=0}^{\ell}$ be generated by Algorithm 1, and consider D_ℓ as in (16). Given $\varepsilon > 0$, assume that $\|d^k\| > \varepsilon$ for every $k = 0, 1, \dots, \ell$. Then, the following statements hold:

a) if $\alpha = 0$, then $|D_\ell| = 0$ (equivalently, $D_\ell = \emptyset$);
b) if $\alpha \in (0, 1)$, then

$$|D_\ell| \le \left| \frac{\log\left(\omega_0 \varepsilon^{-1}\right)}{\log(\alpha)} \right|; \tag{18}$$

c) the number ℓ of iterations of Algorithm 1 is bounded by $\tau_\ell := (|D_\ell| + 1)\left(1 + \max_{t=0,\dots,|D_\ell|} |I_t|\right)$.

Proof (a) This statement follows immediately from the definition of D_ℓ in (16) and the fact that $d^k \neq 0$ for every $k = 0, \dots, \ell$.

(b) If $D_\ell = \emptyset$, then (18) trivially holds. Assume that $D_\ell \neq \emptyset$. Since $\varepsilon \le \|d^k\|$ for all $k = 0, 1, \dots, \ell$, it follows from Lemma 3 that

$$\varepsilon \le \|d^{k_{|D_\ell|}}\| \le \alpha^{|D_\ell|} \omega_0.$$

Thus, we have

$$0 \le \log(\alpha^{|D_\ell|} \omega_0 \varepsilon^{-1}) = |D_\ell| \log \alpha + \log(\omega_0 \varepsilon^{-1})$$

which, in view of the fact that $\alpha \in (0, 1)$, implies that

$$|D_\ell| |\log \alpha| \le \log(\omega_0 \varepsilon^{-1}) \le |\log(\omega_0 \varepsilon^{-1})|.$$

Hence, (18) follows.

(c) First, considering $D_\ell = \{k_i\}_{i=1}^{|D_\ell|}$, define $\tilde{D}_\ell := \{k_i + 1\}_{i=1}^{|D_\ell|}$, which corresponds to the set of flexible iterations of Algorithm 1 up to the ℓ-th iteration. Recall that I_t denotes the t-th cycle of conservative iterations, and let $\mathcal{N}_\ell$ denote the number of such cycles up to the ℓ-th iteration. It is easy to see that $|\tilde{D}_\ell| = |D_\ell|$ and that $\mathcal{N}_\ell \le |D_\ell| + 1$, given that Algorithm 1 starts with a cycle I_0 of conservative iterations and that a new cycle may occur after each flexible iteration $k_i + 1$. Hence, we have $\{0, 1, \dots, \ell\} = \tilde{D}_\ell \cup_{t=0}^{\mathcal{N}_\ell - 1} I_t$, and

$$\begin{aligned} \ell + 1 = |\tilde{D}_\ell| + \sum_{t=0}^{\mathcal{N}_\ell - 1} |I_t| &\le |D_\ell| + \mathcal{N}_\ell \max_{t=0,\dots,\mathcal{N}_\ell - 1} |I_t| \le |D_\ell| + (|D_\ell| + 1) \max_{t=0,\dots,|D_\ell|} |I_t| \\ &\le (|D_\ell| + 1)\left(1 + \max_{t=0,\dots,|D_\ell|} |I_t|\right), \end{aligned}$$

proving the statement in (c). □

The next result shows that the constant ηL acts as a threshold: if the stepsize parameter b_k exceeds this value, Algorithm 1 guarantees a decrease of $F_j(x^k)$. Conversely, if b_k is smaller than this threshold, the functional value $F_j(x^k)$ may not decrease; however, any possible increase is bounded in terms of this threshold and the size of the direction d^k.

Lemma 5 Let $\{(x^k, d^k, b_k)\}_{k\geq 0}$ be generated by Algorithm 1. Then, the following holds:

$$F_j(x^{k+1}) \leq F_j(x^k) - \frac{\eta}{b_{k+1}}\left(1 - \frac{\eta L}{2b_{k+1}}\right)\|d^k\|^2, \quad \forall j \in \mathcal{J}. \tag{19}$$

As a consequence, for every $j \in \mathcal{J}$, we have if $b_{k+1} \geq \eta L$, then $F_j(x^{k+1}) \leq F_j(x^k)$; otherwise,

$$F_j(x^{k+1}) \leq F_j(x^k) + \frac{\eta^2 L}{2b_{k+1}^2}\|d^k\|^2.$$

Proof From (2), the definition $L = \max_{j\in\mathcal{J}} L_j$, and the update rule (14), it follows that, for every $j \in \mathcal{J}$ and $k = 0, \ldots, \ell - 1$,

$$\begin{aligned}
F_j(x^{k+1}) &\leq F_j(x^k) + \langle \nabla F_j(x^k), x^{k+1} - x^k\rangle + \frac{L_j}{2}\|x^{k+1} - x^k\|^2 \\
&\leq F_j(x^k) + \max_{j\in\mathcal{J}}\langle \nabla F_j(x^k), x^{k+1} - x^k\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2 \\
&= F_j(x^k) + \frac{\eta}{b_{k+1}}\max_{j\in\mathcal{J}}\langle \nabla F_j(x^k), d^k\rangle + \frac{\eta^2 L}{2b_{k+1}^2}\|d^k\|^2 \\
&= F_j(x^k) - \frac{\eta}{b_{k+1}}\left(1 - \frac{\eta L}{2b_{k+1}}\right)\|d^k\|^2,
\end{aligned}$$

where the equality is due to the second inequality in (11). Therefore, the first statement of the lemma follows. The last statements for the theorem follow immediately from the first one. □

The next two results analyze the behavior of Algorithm 1 during a cycle of conservative steps. The first result shows that if $\|d^k\|$ is not sufficiently small, then the stepsize parameter b_k eventually becomes larger than a specific threshold determined by the Lipschitz constant L. The second result demonstrates, in essence, that if b_k exceeds the aforementioned threshold, the directions of minimum norm can be overestimated using the size of the corresponding conservative cycle, the functional value at the first iteration of the conservative cycle, the optimal functional value, and basic constants provided by the method.

Lemma 6 Let $\{(d^k, b_k)\}_{k\in I_t}$ be generated by Algorithm 1, where I_t is a cycle of convervative iterations. Given a tolerance $\varepsilon > 0$, assume that $\|d^k\| > \varepsilon$, for all $k \in I_t$, and that $|I_t| \geq N_\varepsilon := \lceil \eta^2 L^2/\varepsilon^2 \rceil$. Then, $b_{t_0+N_\varepsilon} \geq \eta L$, where $t_0 = 0$ if $t = 0$, and $t_0 = k_t + 1$ if $t \neq 0$.

Proof From the definition of b_{k+1} in (12) and the facts that t_0+1 is the first positive element in I_t and $\|d^k\| > \varepsilon$, for each $k \in I_t$, we have

$$b_{t_0+l+1}^2 = b_{t_0}^2 + \sum_{i=t_0}^{t_0+l} \|d^i\|^2 \geq (l+1)\varepsilon^2, \qquad \forall l \geq 0.$$

Hence, if $l := \lceil \eta^2 L^2/\varepsilon^2 \rceil - 1$, we obtain $b_{t_0+l+1} \geq \eta L$, and then the lemma follows by noting that $l+1 = N_\varepsilon$. □

Henceforth, we assume that there exists $j_* \in \mathcal{J}$ such that

$$F_{j_*}^* := \inf_x F_{j_*}(x) > -\infty. \tag{20}$$

Lemma 7 Let $\{(x^k, d^k, b_k)\}_{k \in I_t}$ be generated by Algorithm 1, where I_t is a cycle of conservative iterations. Assume that there exists an index $\tilde{k}_t \in I_t$ such that $\tilde{k}_t \geq 1$ and $b_{\tilde{k}_t} \geq \eta L$. Define $\tilde{I}_t := \{\tilde{k}_t - 1, \dots, k_{t+1} - 1\}$. Then,

$$\sum_{k \in \tilde{I}_t} \frac{\|d^k\|^2}{b_{k+1}} \leq C_t, \qquad \min_{k \in \tilde{I}_t} \|d^k\| \leq \left(\frac{C_t^2 + C_t b_{\tilde{k}_t - 1}}{|\tilde{I}_t|} \right)^{\frac{1}{2}}, \tag{21}$$

where

$$C_t := \frac{2}{\eta}(F_{j_*}(x^{\tilde{k}_t - 1}) - F_{j_*}^*), \tag{22}$$

and j_* is as in (20).

Proof Note that, for all $k \in \tilde{I}_t$, we have $k+1 \in I_t$, which corresponds to a conservative cycle. Hence, for every $k \in \tilde{I}_t$, we have $b_{k+1} \geq b_{\tilde{k}_t} \geq \eta L$. Thus, (19) implies that

$$F_j(x^{k+1}) \leq F_j(x^k) - \frac{\eta}{2b_{k+1}} \|d^k\|^2, \qquad \forall k \in \tilde{I}_t,\ \forall j \in \mathcal{J}.$$

Thus, rewriting the above inequality with $j = j_*$ and summing over $k \in \tilde{I}_t$, we obtain

$$\sum_{k \in \tilde{I}_t} \frac{\|d^k\|^2}{b_{k+1}} \leq \frac{2(F_{j_*}(x^{\tilde{k}_t - 1}) - F_{j_*}(x^{k_t+1}))}{\eta} \leq \frac{2(F_{j_*}(x^{\tilde{k}_t - 1}) - F_{j_*}^*)}{\eta} = C_t, \tag{23}$$

which proves the first inequality in (21). Now, let $Z_t := \sum_{k \in \tilde{I}_t} \|d^k\|^2$. Since, for all $k \in \tilde{I}_t$, we have $k+1 \in I_t$, which corresponds to a conservative cycle whose last element is k_{t+1}, the update rule for b_{k+1} in (12) yields $b_{k+1} \leq b_{k_{t+1}} = \sqrt{Z_t + b_{\tilde{k}_t - 1}^2}$, for all $k \in \tilde{I}_t$. Thus, (23) implies that

$$\frac{1}{b_{k_t+1}} \sum_{k \in \tilde{I}_t} \|d^k\|^2 = \frac{Z_t}{\sqrt{Z_t + b_{\tilde{k}_t - 1}^2}} \leq C_t.$$

The last inequality above is equivalent to

$$Z_t^2 - C_t^2 Z_t - C_t^2 b_{\tilde{k}_t - 1}^2 \leq 0.$$

It follows from the above quadratic inequality on Z_t and simple calculus that

$$|\tilde{I}_t| \min_{k \in \tilde{I}_t} \|d^k\|^2 \leq \sum_{k \in \tilde{I}_t} \|d^k\|^2 = Z_t \leq \frac{C_t^2 + \sqrt{C_t^4 + 4C_t^2 b_{\tilde{k}_t - 1}^2}}{2} \leq C_t^2 + C_t b_{\tilde{k}_t - 1},$$

which implies the second inequality in (21). □

In the following, we show, in particular, that although Algorithm 1 may not produce a sequence of decreasing functional values $F_j(x^k)$, $j \in \mathcal{J}$, it is still possible to uniformly bound the terms of this sequence in terms of the initial functional value $F_j(x^0)$, some initial parameters, and a prescribed tolerance $\varepsilon > 0$. Moreover, it is important to note that the dependence of this upper bound on the tolerance ε is of order $\log(\varepsilon^{-1})$. This result will be used to establish a uniform upper bound for the scalar C_t defined in (22).

Lemma 8 Let $\{(x^k, d^k)\}_{k=0}^{\ell}$ be generated by Algorithm 1. Given a tolerance $\varepsilon > 0$, assume that $\|d^k\| > \varepsilon$, for every $k = 0, 1, \ldots, \ell$. Then, for any $j \in \mathcal{J}$ and $k = 0, \ldots, \ell$, we have

$$F_j(x^k) \leq F_j(x^0) + R_\varepsilon, \quad \text{and} \quad \sum_{k=0}^{\ell} \frac{\|d^k\|^2}{b_{k+1}^2} \leq 2\left(\frac{R_\varepsilon}{\eta^2 L} + \frac{(F_{j_*}(x^0) - F_{j_*}^* + R_\varepsilon)K_\varepsilon}{\eta b_{\min}}\right), \tag{24}$$

where

$$R_\varepsilon := \frac{\eta^2 L}{2}\left[\frac{\alpha^2\omega_0^2}{(1-\alpha^2)b_{\min}^2} + 2K_\varepsilon \log\left(\frac{\eta L}{b_{\min}}\right)\right], \tag{25}$$

with

$$K_\varepsilon := 1 \ \text{ if } \ \alpha = 0, \ \text{ and } \ K_\varepsilon := 1 + \left|\frac{\log\left(\omega_0\varepsilon^{-1}\right)}{\log(\alpha)}\right|, \ \text{ if } \ \alpha \in (0,1). \tag{26}$$

Proof It follows from Lemma 5 that if $b_{k+1} \geq \eta L$, then $F_j(x^{k+1}) \leq F_j(x^k)$, and if $b_{k+1} < \eta L$, then

$$F_j(x^{k+1}) \leq F_j(x^k) + \frac{\eta^2 L}{2b_{k+1}^2}\|d^k\|^2,$$

for every $j \in \mathcal{J}$. Hence, denoting $B_k := \{l \leq k : b_{l+1} < \eta L\}$, we have that if $B_k = \emptyset$, then the first inequality in (24) trivially holds. Now, assume that $B_k \neq \emptyset$. Thus, we have

$$F_j(x^{k+1}) \leq F_j(x^0) + \frac{\eta^2 L}{2}\sum_{l \in B_k}\frac{\|d^l\|^2}{b_{l+1}^2}, \qquad \forall k = 0, \ldots, \ell - 1. \tag{27}$$

Now, note that

$$\sum_{l \in B_k}\frac{\|d^l\|^2}{b_{l+1}^2} = \sum_{l \in B_k \cap D_k}\frac{\|d^l\|^2}{b_{l+1}^2} + \sum_{l \in B_k \cap D_k^C}\frac{\|d^l\|^2}{b_{l+1}^2}, \tag{28}$$

where $D_k = \{k_1, \ldots, k_{|D_k|}\}$ is defined in (16). If $D_k \neq \emptyset$, then it follows from Lemma 3 and by using $b_{k_i+1} = \hat{b}_{k_i} \geq b_{\min}$, that

$$\sum_{l \in B_k \cap D_k}\frac{\|d^l\|^2}{b_{l+1}^2} \leq \sum_{l \in D_k}\frac{\|d^l\|^2}{b_{l+1}^2} = \sum_{i=1}^{|D_k|}\frac{\|d^{k_i}\|^2}{b_{k_i+1}^2} \leq \sum_{i=1}^{|D_k|}\alpha^{2i}\frac{\omega_0^2}{b_{\min}^2} \leq \left(\frac{\alpha^2}{1-\alpha^2}\right)\frac{\omega_0^2}{b_{\min}^2}. \tag{29}$$

On the other hand, in view of the update rule of b_{k+1} in (12) and the definition of B_k, we see that $B_k \cap D_k^C = \cup_{i=0}^{|D_k|} J_i$, where, for each $i = 0, \ldots, |D_k|$, we have $J_i = \emptyset$ or $J_i = \{k_i + 1, \ldots, \hat{k}_i\}$, with $k_i < \hat{k}_i < k_{i+1}$, $k_0 := -1$, $\hat{k}_{|D_k|} \leq k$, and $k_{|D_k|+1} := k + 1$. Define $\hat{J} := \{i = 0, \ldots, |D_k| : J_i \neq \emptyset\}$. Thus, the last term in (28) can be rewritten as

$$\sum_{l \in B_k \cap D_k^C}\frac{\|d^l\|^2}{b_{l+1}^2} = \sum_{i \in \hat{J}}\sum_{l \in J_i}\frac{\|d^l\|^2}{b_{l+1}^2}. \tag{30}$$

Now, observe that $l+1$ is a conservative iteration for any $l \in J_i$ with $i \in \hat{J}$. Therefore, by the update rule (12), we have

$$b_{l+1}^2 = b_l^2 + \|d^l\|^2.$$

Combining this relation with Lemma 2, and defining $a_l := \|d^l\|^2/b_{k_i+1}^2$, we obtain

$$\begin{aligned}
\sum_{l\in J_i} \frac{\|d^l\|^2}{b_{l+1}^2} &= \sum_{l=k_i+1}^{\hat{k}_i} \frac{\|d^l\|^2}{b_{k_i+1}^2 + \sum_{t=k_i+1}^{l}\|d^t\|^2} = \sum_{l=k_i+1}^{\hat{k}_i} \frac{a_l}{1+\sum_{t=k_i+1}^{l} a_t} = \sum_{t=1}^{\hat{k}_i-k_i} \frac{a_{k_i+t}}{1+\sum_{r=1}^{t} a_{k_i+r}} \\
&\le \log\left(1+\sum_{l=1}^{\hat{k}_i-k_i} a_{k_i+l}\right) = \log\left(1+\sum_{l=k_{i+1}}^{\hat{k}_i} a_l\right) = \log\left(1+\sum_{l=k_{i+1}}^{\hat{k}_i} \frac{\|d_l\|^2}{b_{k_i+1}^2}\right) \\
&= \log\left(\frac{b_{\hat{k}_i+1}^2}{b_{k_i+1}^2}\right) \le \log\left(\frac{\eta^2 L^2}{b_{\min}^2}\right),
\end{aligned}$$

where the last inequality is due to $b_{k_i+1} = \hat{b}_{k_i} \ge b_{\min}$ and $b_{\hat{k}_i+1} < \eta L$, because $\hat{k}_i \in B_k$. Hence, combining the latter inequality with (30) and the fact that $|\hat{J}| \le |D_k|+1$, we get

$$\sum_{l\in B_k\cap D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2} \le \sum_{i\in\hat{J}} \log\left(\frac{\eta^2L^2}{b_{\min}^2}\right) \le 2(|D_k|+1)\log\left(\frac{\eta L}{b_{\min}}\right). \tag{31}$$

Note that if $D_k = \emptyset$ (which is the case if the initial parameter $\alpha = 0$), then $\hat{J} = \emptyset$ or $\hat{J} = \{0\}$, i.e, only $J_0 \neq \emptyset$, and hence the bound in (31) still holds with $|D_k| = 0$. Therefore, since (18) implies that $|D_k| \le |\log\left(\omega_0\varepsilon^{-1}\right)/\log(\alpha)|$, the first inequality in (24) follows by combining (27), (28), (29), (31), and the definitions of R_ε and K_ε.

We now proceed to prove the second inequality in (24). It follows from (29) and (31) that

$$\begin{aligned}
\sum_{l=0}^{k} \frac{\|d^l\|^2}{b_{l+1}^2} &= \sum_{l\in D_k} \frac{\|d^l\|^2}{b_{l+1}^2} + \sum_{l\in D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2} = \sum_{l\in D_k} \frac{\|d^l\|^2}{b_{l+1}^2} + \sum_{l\in B_k\cap D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2} + \sum_{l\in B_k^C\cap D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2} \\
&\le \left(\frac{\alpha^2}{1-\alpha^2}\right)\frac{\omega_0^2}{b_{\min}^2} + 2\left(|D_k|+1\right)\log\left(\frac{\eta L}{b_{\min}}\right) + \sum_{l\in B_k^C\cap D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2} \\
&\le \frac{2R_\varepsilon}{\eta^2 L} + \sum_{l\in B_k^C\cap D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2},
\end{aligned} \tag{32}$$

where the last inequality is due to (18) and the definition of R_ε in (25). In view of the definitions of B_k and D_k, we observe that $B_k^C \cap D_k^C = \bigcup_{t=0}^{|D_k|} \tilde{I}_t$, where, for each $t = 0, \dots, |D_k|$, either $\tilde{I}_t = \emptyset$ or $\tilde{I}_t$ is defined as in Lemma 7, namely, $\tilde{I}_t = \{\tilde{k}_t - 1, \dots, k_{t+1}-1\}$, with $\tilde{k}_t \ge 1$. Let us consider $\tilde{I} := \{t = 0, \dots, |D_k| : \tilde{I}_t \neq \emptyset\}$. Thus, by noting that $b_{l+1} \ge b_{\min}$ and applying the first inequality in (21), we obtain

$$\begin{aligned}
\sum_{l\in B_k^C\cap D_k^C} \frac{\|d^l\|^2}{b_{l+1}^2} &= \sum_{t\in\tilde{I}}\sum_{l\in\tilde{I}_t} \frac{\|d^l\|^2}{b_{l+1}^2} \le \frac{1}{b_{\min}}\sum_{t\in\tilde{I}}\sum_{l\in\tilde{I}_t} \frac{\|d^l\|^2}{b_{l+1}} \le \frac{1}{b_{\min}}\sum_{t\in\tilde{I}} C_t \\
&\le \frac{|\tilde{I}|\max_{t\in\tilde{I}} C_t}{b_{\min}} \le \frac{2|D_k|(F_{j_*}(x^0) - F_{j_*}^* + R_\varepsilon)}{\eta b_{\min}},
\end{aligned} \tag{33}$$

where the last inequality is due to the definition of C_t in (22), the first inequality in (24) and the fact that $|\tilde{I}| \le |D_k|$. Therefore, the second inequality in (25) follows from (32), (33), and the definition of K_ε in (26). □

In the following, we establish the main result of this section, which is the worst-case iteration-complexity bound on the total number of iterations of Algorithm 1 for computing an approximate Pareto critical to problem (1). This approximation is measured in terms of the sequence $\{d^k\}$ generated by the method. Indeed, given a tolerance $\varepsilon > 0$, we are interested in estimating the number of iterations K required to obtain d^K such that $\|d^K\| \leq \varepsilon$. Moreover, note that in the scalar case, for which $\mathcal{J} = \{1\}$, we have $d^k = -\nabla f(x^k)$, and hence the above concept of approximate solution consists of obtaining a point x^K satisfying $\|\nabla f(x^K)\| \leq \varepsilon$.

Theorem 9 *Let $\{(x^k, d^k)\}_{k\geq 0}$ be generated by Algorithm 1. Given a tolerance $\varepsilon > 0$, in at most K iterations, we have $\|d^K\| \leq \varepsilon$, where*

$$K \leq K_\varepsilon(1 + N_\varepsilon + M_\varepsilon), \tag{34}$$

with

$$N_\varepsilon = \left\lceil \frac{\eta^2 L^2}{\varepsilon^2} \right\rceil, \quad M_\varepsilon := \left\lceil \frac{\frac{4}{\eta^2}(F_{j_*}(x^0) - F^*_{j_*} + R_\varepsilon)^2 + \frac{2}{\eta}(F_{j_*}(x^0) - F^*_{j_*} + R_\varepsilon)\max\{b_{\max}, \eta L\}}{\varepsilon^2} \right\rceil,$$

and R_ε and K_ε are as in (25) and (26), respectively.

Proof Assume that $\|d^k\| > \varepsilon$ for every $k = 0, \ldots \ell$. In view of Lemma 4 (a)-(b), the number $|D_\ell|$ of flexible iterations of Algorithm 1 up to the ℓ-th iteration is bounded by $|\log\left(\omega_0 \varepsilon^{-1}\right)|/|\log(\alpha)|$ if $\alpha \neq 0$ or $|D_\ell| = 0$ if $\alpha = 0$. Hence, in view of the definition of K_ε in (26), we have $|D_\ell| \leq K_\varepsilon - 1$. Thus, it follows from Lemma 4(c) that the number ℓ of iterations is bounded by $\tau_\ell := K_\varepsilon\left(1 + \max_{t=0,\ldots,|D_\ell|}|I_t|\right)$, where I_t corresponds to the t-th cycle of conservative iterations. Therefore, the desired bound (34) follows once we show that $|I_t| \leq N_\varepsilon + M_\varepsilon$ for all $t = 0, \ldots, |D_\ell|$. If $|I_t| < N_\varepsilon$ for all $t = 0, \ldots, |D_\ell|$, the result follows trivially. Thus, suppose that there exists t such that $|I_t| \geq N_\varepsilon$. Fix such a t, and let $\tilde{k}_t$ be the first element of I_t satisfying $\tilde{k}_t \geq 1$ and $b_{\tilde{k}_t} \geq \eta L$. The existence of such an index is guaranteed by Lemma 6, which ensures that $b_{t_0+N_\varepsilon} \geq \eta L$, where $t_0 = 0$ if $t = 0$, and $t_0 = k_t + 1$ if $t \neq 0$. Hence, we have the following two cases to consider:

- the iteration $\tilde{k}_t$ is not the first element in the cycle I_t, in which case $b_{\tilde{k}_t - 1} < \eta L$.
- the iteration $\tilde{k}_t$ corresponds to the first positive element in the cycle I_t, which implies that either $\tilde{k}_t = 1$, or $\tilde{k}_t - 1$ is the last flexible iteration preceding the cycle I_t. Hence, we have that either $b_{\tilde{k}_t-1} = b_0$ or $b_{\tilde{k}_t-1} = \hat{b}_{\tilde{k}_t-2}$. In particular, $b_{\tilde{k}_t-1} \leq b_{\max}$, in view of Step 1 of Algorithm 1 and (13).

Thus, in both cases, we must have $b_{\tilde{k}_t-1} \leq \max\{b_{\max}, \eta L\}$. On the other hand, the definition of C_t in (22) and the first inequality in (24) with $k = \tilde{k}_t - 1$ imply that

$$C_t = \frac{2}{\eta}(F_{j_*}(x^{\tilde{k}_t-1}) - F^*_{j_*}) \leq \frac{2}{\eta}(F_{j_*}(x^0) - F^*_{j_*} + R_\varepsilon).$$

Hence, in view of the second inequality in (21), the fact that $b_{\tilde{k}_t-1} \leq \max\{b_{\max}, \eta L\}$, and the definition of M_ε, we have

$$\min_{k\in\tilde{I}_t}\|d^k\| \leq \left(\frac{\frac{4}{\eta^2}\left(F_{j_*}(x^0) - F^*_{j_*} + R_\varepsilon\right)^2 + \frac{2}{\eta}\left(F_{j_*}(x^0) - F^*_{j_*} + R_\varepsilon\right)\max\{b_{\max}, \eta L\}}{|\tilde{I}_t|}\right)^{\frac{1}{2}}$$

$$\leq \varepsilon \frac{M_\varepsilon^{\frac{1}{2}}}{|\tilde{I}_t|^{\frac{1}{2}}},$$

where $\tilde{I}_t = \{\tilde{k}_t, \dots, k_{t+1}\}$ is as in Lemma 7. Since $\|d^k\| > \varepsilon$ for every $k = 0, \dots \ell$, the above inequalities imply that $|\tilde{I}_t| < M_\varepsilon$. Now, note that

$$I_0 = \{0, \dots, \tilde{k}_t - 1\} \cup \tilde{I}_0, \qquad I_t = \{k_t + 2, \dots, \tilde{k}_t - 1\} \cup \tilde{I}_t, \text{ if } t \neq 0,$$

where $\tilde{k}_t \leq N_\varepsilon$ if $t = 0$, and $\tilde{k}_t \leq k_t + 1 + N_\varepsilon$ if $t \neq 0$, in view of Lemma 6. Therefore, $|I_t| \leq N_\varepsilon + |\tilde{I}_t| < N_\varepsilon + M_\varepsilon$, for any $t = 0, \dots, |D_\ell|$, which concludes the proof of the theorem. □

Remark 2 If $\alpha = 0$ (conservative variant), Theorem 9 implies that Algorithm 1 requires at most $\mathcal{O}(\varepsilon^{-2})$ iterations (or evaluations of the steepest descent direction) to reach an ε-approximate Pareto critical point. In contrast, the flexible variant ($\alpha \in (0, 1)$) achieves a slightly weaker bound of $\mathcal{O}(\varepsilon^{-2} \log^3(\varepsilon^{-1}))$. We also note that, in the scalar case ($\mathcal{J} = \{1\}$), the iteration complexity established in Theorem 9 for the conservative variant is consistent with the result in [27, Theorem 2.2].

We conclude this section by presenting a liminf-type global convergence result for Algorithm 1.

Corollary 10 Let $\{x_k\}_{k\geq 0}$ be an infinite sequence generated by Algorithm 1. Then

$$\liminf_{k\to+\infty} \|d^k\| = 0. \tag{35}$$

Proof For any natural number $j \geq 1$, by applying Theorem 9 successively with tolerance $\varepsilon_j = 1/j$, we obtain an element $d^{K(j)}$ of the sequence $\{d^k\}$ generated by Algorithm 1 such that $\|d^{K(j)}\| \leq 1/j$. As a consequence, $\liminf_{k\to+\infty} \|d^k\| = 0$. □

4.2 Convex Case

In this subsection, we analyze the conservative variant of Algorithm 1 (case $\alpha = 0$) under the assumption that all components of F are convex functions. Let us consider the merit function $u_0 : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ defined by

$$u_0(x) = \sup_{z\in\mathbb{R}^n} \min_{j\in\mathcal{J}} \{F_j(x) - F_j(z)\}. \tag{36}$$

By [17, Theorem 3.1], we have $u_0(x) \geq 0$ for all $x \in \mathbb{R}^n$ and $u_0(x) = 0$ if and only if x is a weak Pareto optimal point of F.

Lemma 11 Given $x \in \mathbb{R}^n$, let

$$\mathcal{L}(F(x)) := \left\{y \in \mathbb{R}^n \,:\, F(y) \leq F(x)\right\}.$$

Then

$$u_0(x) = \sup_{z\in\mathcal{L}(F(x))} \min_{j\in\mathcal{J}} \left\{F_j(x) - F_j(z)\right\}.$$

Proof Consider the function $\phi : \mathbb{R}^n \to \mathbb{R}$ given by

$$\phi(z) = \min_{j\in\mathcal{J}} \left\{F_j(x) - F_j(z)\right\}. \tag{37}$$

If $z \in \mathcal{L}(F(x))$ then

$$F_j(x) - F_j(z) \geq 0, \quad j \in \mathcal{J},$$

and so $\phi(z) \geq 0$. Since this is true for any $z \in \mathcal{L}(F(x))$, it follows that

$$\sup_{z\in\mathcal{L}(F(x))} \phi(z) \geq 0. \tag{38}$$

On the other hand, if $z \in \mathcal{L}(F(x))^c$ then there exists $j_z \in \mathcal{J}$ such that $F_{j_z}(z) > F_{j_z}(x)$, and so $\phi(z) < 0$. Since this is true for any $z \in \mathcal{L}(F(x))^c$, we have

$$\sup_{z\in\mathcal{L}(F(x))^c} \phi(z) \leq 0. \tag{39}$$

Finally, combining (36)–(39), we obtain

$$\begin{aligned} u_0(x) &= \sup_{z\in\mathbb{R}^n} \phi(z) = \max\left\{\sup_{z\in\mathcal{L}(F(x))} \phi(z), \sup_{z\in\mathcal{L}(F(x))^c} \phi(z)\right\} = \sup_{z\in\mathcal{L}(F(x))} \phi(z) \\ &= \sup_{z\in\mathcal{L}(F(x))} \min_{j\in\mathcal{J}} \left\{F_j(x) - F_j(z)\right\}. \end{aligned}$$

□

Let us consider explicitly the following assumption:

A1. $F_j(\,\cdot\,)$ is convex for each $j \in \mathcal{J}$.

Under A1, the next lemma establishes the connection between $u_0(x)$ and $d_{SD}(x)$.

Lemma 12 Suppose that A1 holds. Then, for any $x \in \mathbb{R}^n$, we have

$$u_0(x) \leq \left(\sup_{z\in\mathcal{L}(F(x))} \|z - x\|\right) \|d_{SD}(x)\|.$$

Proof Given $z \in \mathbb{R}^n$ and $i \in \mathcal{J}$, it follows from A1 that

$$F_i(z) \geq F_i(x) + \langle \nabla F_i(x), z - x\rangle,$$

which implies

$$\langle \nabla F_i(x), x - z\rangle \geq F_i(x) - F_i(z) \geq \min_{j\in\mathcal{J}} \left\{F_j(x) - F_j(z)\right\}. \tag{40}$$

Recall that

$$d_{SD}(x) = -\sum_{i=1}^{m} \lambda_i^{SD} \nabla F_i(x), \quad \sum_{i=1}^{m} \lambda_i^{SD}(x) = 1, \quad \text{and} \quad \lambda_i^{SD}(x) \geq 0 \ \ \forall i \in \mathcal{J}.$$

Then, it follows from (40) and the Cauchy-Schwarz inequality that

$$\begin{aligned} \min_{j\in\mathcal{J}} \left\{F_j(x) - F_j(z)\right\} &\leq \sum_{i=1}^{m} \lambda_i^{SD}(x) \langle \nabla F_i(x), x - z\rangle \\ &= \left\langle \sum_{i=1}^{m} \lambda_i^{SD}(x) \nabla F_i(x), x - z\right\rangle \end{aligned}$$

$$= \left\langle -d^{SD}(x), x - z \right\rangle$$
$$\leq \|z - x\| \|d^{SD}(x)\|.$$

The proof of the lemma follows by taking the supremum over $z \in \mathcal{L}(F(x))$ on both sides of the above inequalities and by using Lemma 11. □

Consider the set

$$\Omega_R(x^0) = \left\{ x \in \mathbb{R}^n \,:\, F_j(x) \leq F_j(x^0) + R \right\},$$

where

$$R = \eta^2 L \log\left(\frac{\eta L}{b_{\min}}\right). \tag{41}$$

In what follows we will consider the additional assumption:

A2. $\tilde{D}_0 := \sup_{z \in \Omega_R(x^0)} \|z - x^0\| < +\infty$.

Theorem 13 *Suppose that A1-A2 holds, and let $\{x^k\}_{k\geq 0}$ be generated by Algorithm 1 with $\alpha = 0$. Then*

$$\liminf_{k \to +\infty} u_0(x^k) = 0. \tag{42}$$

Proof Let $k \in \mathbb{N}$. Note that the scalar R, defined in (41), coincides with the quantity R_ε defined in (25) when $\alpha = 0$. Hence, it follows from the first inequality in (24) that $x^k \in \Omega_R(x^0)$. Thus, $\mathcal{L}(F(x^k)) \subset \Omega_R(x^0)$ and so, by A2,

$$\sup_{z \in \mathcal{L}(F(x^k))} \|z - x^k\| \leq \sup_{z \in \Omega_R(x^0)} \left(\|z - x^0\| + \|x^0 - x^k\| \right) \leq 2\tilde{D}_0.$$

Consequently, it follows from Lemma 12 that

$$u_0(x^k) \leq 2\tilde{D}_0 \|d_{SD}(x^k)\| = 2\tilde{D}_0 \|d^k\|. \tag{43}$$

Since (43) holds for every k, it follows from Corollary 10 that (42) is true. □

By Theorem 13, given $\varepsilon > 0$ there exists k such that $u_0(x^k) \leq \varepsilon$, i.e., Algorithm 1 with $\alpha = 0$ is guaranteed to find an ε-approximate weak Pareto optimal point. Let us define

$$T(\varepsilon) = \inf \left\{ k \in \mathbb{N} \,:\, u_0(x^k) \leq \varepsilon \right\}. \tag{44}$$

In what follows we will establish an upper bound of $\mathcal{O}(\varepsilon^{-1})$ on $T(\varepsilon)$ when b_1 is sufficiently large. For that, let us consider a weaker version of assumption A2:

A2'. $D_0 := \sup_{z \in \mathcal{L}(F(x^0))} \|z - x^0\| < +\infty$.

Note that A2' means that $\mathcal{L}(F(x^0)$ is bounded. Since this set is closed (due to the continuity of F), we conclude that it is compact. Thus, by the Weierstrass Theorem, it follows that the function $\psi_F : \mathbb{R}^n \to \mathbb{R}$, given by

$$\psi_F(x) = \max_{j \in \mathcal{J}} \left\{ F_j(x) \right\},$$

has a global minimizer x^*. Let us denote $\psi_F^* = \psi_F(x^*)$. Our next lemma provides all the auxiliary inequalities we need to obtain the referred complexity bound.

Lemma 14 Suppose that A2' holds and let $\{x^k\}_{k\geq 0}$ be generated by Algorithm 1 with $\alpha = 0$. If $b_1 = \sqrt{b_0^2 + \|d^0\|^2} \geq \eta L$, then we have:

(a) $F(x^{k+1}) \preceq F(x^k)$ for all $k \geq 0$.
(b) $\sum_{k=0}^{T} \left(\frac{\eta}{b_{k+1}}\right)^2 \|d^k\|^2 \leq \frac{2}{L}\left[\psi_F(x^0) - \psi_F^*\right]$, for all $T \geq 0$.
(c) $b_{k+1} \leq b_0 + \frac{2}{\eta}\left[\psi_F(x^0) - \psi_F^*\right]$ for all $k \geq 0$.

Proof Since $\alpha = 0$, the sequence $\{b_k\}_{k\geq 1}$ generated by Algorithm 1 is nondecreasing. Therefore, the assumption $b_1 \geq \eta L$ implies that

$$b_{k+1} \geq \eta L, \quad \forall k \geq 0. \tag{45}$$

Consequently, by Lemma 5, we have

$$F_j(x^{k+1}) \leq F_j(x^k) - \frac{\eta}{2b_{k+1}}\|d^k\|^2, \quad j \in \mathcal{J} \tag{46}$$

for all $k \geq 0$. This establishes statement (a). Moreover, (46) implies that

$$\psi_F(x^{k+1}) \leq \psi_F(x^k) - \frac{\eta}{2b_{k+1}}\|d^k\|^2,$$

that is

$$\frac{\eta}{b_{k+1}}\|d^k\|^2 \leq 2\left[\psi_F(x^k) - \psi_F(x^{k+1})\right]. \tag{47}$$

Combining (45) and (47), for any $T \geq 1$, we get

$$\sum_{k=0}^{T-1}\left(\frac{\eta}{b_{k+1}}\right)^2\|d^k\|^2 \leq \sum_{k=0}^{T-1}\frac{1}{L}\left(\frac{\eta}{b_{k+1}}\right)\|d^k\|^2 \leq \frac{2}{L}\left[\psi_F(x^0) - \psi_F(x^T)\right]. \tag{48}$$

Thus, statement (b) follows from inequality (48) together with the bound $\psi_F(x^T) \geq \psi_F^*$. Now, regarding item (c), first note that from the definition of b_{k+1} it follows that

$$\frac{\eta}{b_{k+1}}\|d^k\|^2 = \frac{\eta\left(b_{k+1}^2 - b_k^2\right)}{b_{k+1}} = \frac{\eta(b_{k+1}-b_k)(b_{k+1}+b_k)}{b_{k+1}} \geq \frac{\eta(b_{k+1}-b_k)\,b_{k+1}}{b_{k+1}} = \eta(b_{k+1}-b_k),$$

where the inequality follows from the fact that $b_{k+1} \geq b_k$. Thus, for any $T \geq 1$, summing the above inequality for $k = 0, \ldots, T-1$ and using (47), we obtain

$$\eta(b_T - b_0) = \sum_{k=0}^{T-1}\eta(b_{k+1}-b_k) \leq \sum_{k=0}^{T-1}\frac{\eta}{b_{k+1}}\|d^k\|^2 \leq 2\left[\psi_F(x^0) - \psi_F^*\right].$$

Consequently,

$$b_T \leq b_0 + \frac{2}{\eta}\left[\psi_F(x^0) - \psi_F^*\right].$$

Since T is an arbitrary integer greater than or equal to 1, we conclude that statement (c) holds. □

Now we are ready to establish our complexity bound for Algorithm 1 ($\alpha = 0$) when all the components of F are convex and b_1 is sufficiently large.

Theorem 15 *Suppose that A1-A2' hold, and let* $\{x^k\}_{k\geq 0}$ *be generated by Algorithm 1 with* $\alpha = 0$. *If* $b_1 = \sqrt{b_0^2 + \|d^0\|^2} \geq \eta L$, *then, given* $\varepsilon > 0$, *we have*

$$T(\varepsilon) \leq \frac{\left(D_0^2 + \frac{2}{L}\left[\psi_F(x^0) - \psi_F^*\right]\right)\left(b_0 + \frac{2}{\eta}\left[\psi_F(x^0) - \psi_F^*\right]\right)}{2\eta}\varepsilon^{-1} \tag{49}$$

where $T(\varepsilon)$ *is defined in (44).*

Proof If $T(\varepsilon) = 0$, then (49) is clearly true. Therefore, from now one we assume that $T(\varepsilon) \geq 1$. Let us denote $\alpha_k = \eta/b_{k+1}$ and recall that $d^k = d_{SD}(x^k) = -\sum_{i=1}^m \lambda_i^{SD}(x^k)\nabla F_i(x^k)$. Then, given $z \in \mathbb{R}^n$ we have

$$\begin{aligned}\|x^{k+1} - z\|^2 &= \|x^k - z\|^2 + 2\alpha_k\langle x^k - z, d^k\rangle + \alpha_k^2\|d^k\|^2 \\ &= \|x^k - z\|^2 - 2\alpha_k\left\langle x^k - z, \sum_{i=1}^m \lambda_i^{SD}(x^k)\nabla F_i(x^k)\right\rangle + \alpha_k^2\|d^k\|^2. \end{aligned}\tag{50}$$

It follows from assumption A1 that

$$\langle x^k - z, \nabla F_i(x^k)\rangle \geq F_i(x^k) - F_i(z), \quad i \in \mathcal{J},$$

and so, as $\sum_{i=1}^m \lambda_i^{SD}(x^k) = 1$ and $\lambda_i^{SD}(x^k) \geq 0$ for all i, we obtain

$$\begin{aligned}\left\langle x^k - z, \sum_{i=1}^m \lambda_i^{SD}(x^k)\nabla F_i(x^k)\right\rangle &= \sum_{i=1}^m \lambda_i^{SD}(x^k)\langle x^k - z, \nabla F_i(x^k)\rangle \\ &\geq \sum_{i=1}^m \lambda_i^{SD}(x^k)(F_i(x^k) - F_i(z)) \geq \min_{j\in\mathcal{J}}(F_j(x^k) - F_j(z)).\end{aligned}\tag{51}$$

Combining (50) and (51), we get

$$2\alpha_k \min_{j\in\mathcal{J}}(F_j(x^k) - F_j(z)) \leq \|x^k - z\|^2 - \|x^{k+1} - z\|^2 + \alpha_k^2\|d^k\|^2.$$

Summing up this inequality for $k = 0, \ldots, T$, and using item (b) of Lemma 14, it follows that

$$2\sum_{k=0}^T \alpha_k \min_{j\in\mathcal{J}}(F_j(x^k) - F_j(z)) \leq \|x^0 - z\|^2 + \frac{2}{L}\left[\psi_F(x^0) - \psi_F^*\right].$$

Since, by item (a) of Lemma 14, we have $F_j(x^T) \leq F_j(x^k)$, for all $j \in \mathcal{J}$ and $k = 0, \ldots, T$, the above inequality implies that

$$2\min_{j\in\mathcal{J}}(F_j(x^T) - F_j(z))\sum_{k=0}^T \alpha_k \leq \|x^0 - z\|^2 + \frac{2}{L}\left[\psi_F(x^0) - \psi_F^*\right].$$

By taking the supremum over $z \in \mathcal{L}(F(x^T))$ on both sides of the above inequality, $A2'$ together with Lemma 11, yields that

$$2u_0(x^T)\sum_{k=0}^T \alpha_k \leq D_0^2 + \frac{2}{L}\left[\psi_F(x^0) - \psi_F^*\right]. \tag{52}$$

From the definition of α_k and statement (c) in Lemma 14, we have

$$\sum_{k=0}^T \alpha_k = \sum_{k=0}^T \frac{\eta}{b_{k+1}} \geq \sum_{k=0}^T \frac{\eta}{b_0 + \frac{2}{\eta}\left[\psi_F(x^0) - \psi_F^*\right]} = \frac{\eta(T+1)}{b_0 + \frac{2}{\eta}\left[\psi_F(x^0) - \psi_F^*\right]},$$

which combined with (52) yields

$$u_0(x^T) \le \frac{\left(D_0^2 + \frac{2}{L}\left[\psi_F(x^0) - \psi_F^*\right]\right)\left(b_0 + \frac{2}{\eta}\left[\psi_F(x^0) - \psi_F^*\right]\right)}{2\eta(T+1)}.$$

Finally, by considering this inequality with $T = T(\varepsilon) - 1$, we conclude that (49) holds, since, by the definition of $T(\varepsilon)$ in (44), $u_0(x^{T(\varepsilon)-1}) > \varepsilon$. □

5 Numerical Experiments

In this section, we evaluate the performance of the proposed algorithm on two classes of problems: (i) a standard collection of multiobjective test functions, and (ii) a PDE multiobjective optimization problem.

We consider two variants of Algorithm 1, using parameters $\omega_0 = d^0$, $\eta := 1$, $b_0 := 10^{-3}$ and $b_{\min} := 10^{-4}$:

- C-AMG: Algorithm 1 with $\alpha := 0$ (conservative version);
- F-AMG: Algorithm 1 with $\alpha := 0.95$ and $\hat{b}_k$ as in (15) (flexible version).

These variants are compared with two baseline methods from the literature [12]:

- Fixed-SD: the steepest descent method with constant stepsize equal to 1;
- Armijo-SD: the steepest descent method with Armijo line search.

The experiments were conducted using the Python programming language, which was installed on a machine equipped with a 3.5 GHz Dual-Core Intel Core i5 processor and 16 GB of 2400 MHz DDR4 memory. For all algorithms, the steepest descent subproblem (8) was solved using the `cvxpy` package, except in the case $m = 2$, where the subproblem admits a closed-form solution.

5.1 Benchmark Test Problems

We first consider a collection of 44 unconstrained multiobjective test problems, covering both convex and nonconvex scenarios, with varying dimensions n and numbers of objectives m. A summary of these problems is given in Table 1; further details can be found in [20]. The initial points were generated within the box $\ell \le x \le u$, where ℓ and u denote the lower and upper bounds of each problem. More precisely, each initial point was computed as

$$x_0 = (1-\beta)\ell + \beta u, \tag{53}$$

with $\beta \sim U(0,1)$. We consider the stopping criterion $|\theta^k| \le 10^{-4}$, and set a maximum number of iterations to 5,000.

We adopted the total computational cost, denoted by $\text{Cost}(\cdot)$, as the performance measure. The total cost up to iteration K is defined as

$$\text{Cost}(K) := \frac{1}{m}\left(\sum_{i=1}^{m} f_i e(K) + 3 \times g_i e(K)\right), \tag{54}$$

where $f_i e(K)$ denotes the total number of evaluations of the component function $f_i(\cdot)$, and $g_i e(K)$ represents the total number of evaluations of the corresponding

Problem	n	m	Convex	ℓ	u
AP1	2	3	Y	(-10, -10)	(10, 10)
AP2	1	2	Y	-100	100
AP3	2	2	N	(-100, -100)	(100, 100)
AP4	3	3	Y	(-10, -10, -10)	(10, 10, 10)
BK1	2	2	Y	(-5, -5)	(10, 10)
DD1a	5	2	N	(-20, . . . , -20)	(20, . . . , 20)
DGO1	1	2	N	-10	13
Far1	2	2	N	(-1, -1)	(1, 1)
FDS	5	3	Y	(-2, . . . , -2)	(2, . . . , 2)
FF1	2	2	N	(-1, -1)	(1, 1)
Hil1	2	2	N	(0, 0)	(1, 1)
IKK1	2	3	Y	(-50, -50)	(50, 50)
JOS1	100	2	Y	(-100, . . . , -100)	(100, . . . , 100)
KW2	2	2	N	(-3, -3)	(3, 3)
LE1	2	2	N	(-5, -5)	(10, 10)
Lov1	2	2	Y	(-10, -10)	(10, 10)
Lov3	2	2	N	(-20, -20)	(20, 20)
Lov4	2	2	N	(-20, -20)	(20, 20)
Lov5	3	2	N	(-2, -2, -2)	(2, 2, 2)
MGH16b	4	5	N	(-25, -5, -5, -1)	(25, 5, 5, 1)
MGH26b	4	4	N	(-1, -1, -1 - 1)	(1, 1, 1, 1)
MGH33b	10	10	Y	(-1, . . . , -1)	(1, . . . , 1)
MHHM2	2	3	Y	(0, 0)	(1, 1)
MLF2	2	2	N	(-100, -100)	(100, 100)
MMR1c	2	2	N	(0.1, 0)	(1, 1)
MMR3	2	2	N	(-1, -1)	(1, 1)
MOP2	2	2	N	(-1, -1)	(1, 1)
MOP3	2	2	N	($-\pi$, $-\pi$)	(π, π)
MOP5	2	3	N	(-1, -1)	(1, 1)
MOP7	2	3	Y	(-400, -400)	(400, 400)
PNR	2	2	Y	(-2, -2)	(2, 2)
QV1	10	2	N	(-5, . . . , -5)	(5, . . . , 5)
SK1	1	2	N	-100	100
SK2	4	2	N	(-10, -10, -10, -10)	(10, 10, 10, 10)
SLCDT1	2	2	N	(-1.5, -1.5)	(1.5, 1.5)
SLCDT2	10	3	Y	(-1, . . . , -1)	(1, . . . , 1)
SP1	2	2	Y	(-100, -100)	(100, 100)
SSFYY2	1	2	N	-100	100
Toi4b	4	2	Y	(-2, -2, -2, -2)	(5, 5, 5, 5)
Toi8b	3	3	Y	(-1, -1, -1, -1)	(1, 1, 1, 1)
Toi9b	4	4	N	(-1, -1, -1, -1)	(1, 1, 1, 1)
Toi10b	4	3	N	(-2, -2, -2, -2)	(2, 2, 2, 2)
VU1	2	2	N	(-3, -3)	(3, 3)
ZLT1	10	5	Y	(-1000, . . . , -1000)	(1000, . . . , 1000)

Table 1 Test problems

gradient $\nabla f_i(\cdot)$, both accumulated up to iteration K. This cost function reflects the relative computational expense of evaluating gradients compared to function values by assigning a weight of 3 to each gradient evaluation. This choice is motivated by the fact that, when using reverse-mode automatic differentiation, computing $\nabla f_i(\cdot)$ typically requires about three times the computational effort of a single evaluation of $f_i(\cdot)$; see, e.g., [1, 3].

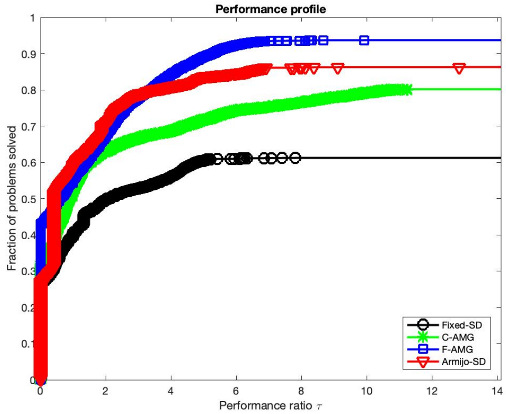

Fig. 1 Performance profiles ($\log_2$ scale) comparing Fixed-SD, C-AMG, F-AMG and Armijo-SD in terms of Cost(K).

We evaluated the algorithms on the 44 test problems listed in Table 1, using 100 different initial points for each problem, resulting in a total of 4,400 instances. The performance profile based on the cost measure Cost(K) is shown in Figure 1. As observed in the figure, F-AMG was the most efficient method in terms of Cost(K), achieving the lowest cost in (42.98%) of the instances. In comparison, Fixed-SD, C-AMG, and Armijo-SD were the best performers in (26.68%), (31.73%) and (27.05%) of the cases, respectively. Moreover, F-AMG also demonstrated the highest robustness among all methods. This superior performance, in terms of efficiency, of the Algorithm 1 variants (C-AMG and F-AMG) over Armijo-SD is largely due to the fact that they do not require evaluations of the objective function. In contrast, Armijo-SD often incurs a large number of function evaluations because of its line search procedure. Moreover, the superior robustness of AMG variants compared with Fixed-SD further highlights another appealing feature of the adaptive methodits global convergence, achieved independently of the stepsize choice.

Naturally, the algorithms may generate different iterates and, as a result, produce distinct approximate Pareto points. Therefore, it is worthwhile to examine whether they recover the Pareto front in a similar manner. Figure 2 illustrates the recovered Pareto fronts for four representative problems: BK1, DGO1, QV1, and Toi4. The plots on the left correspond to F-AMG, while those on the right correspond to Armijo-SD. Visually, the Pareto fronts recovered by F-AMG appear slightly better.

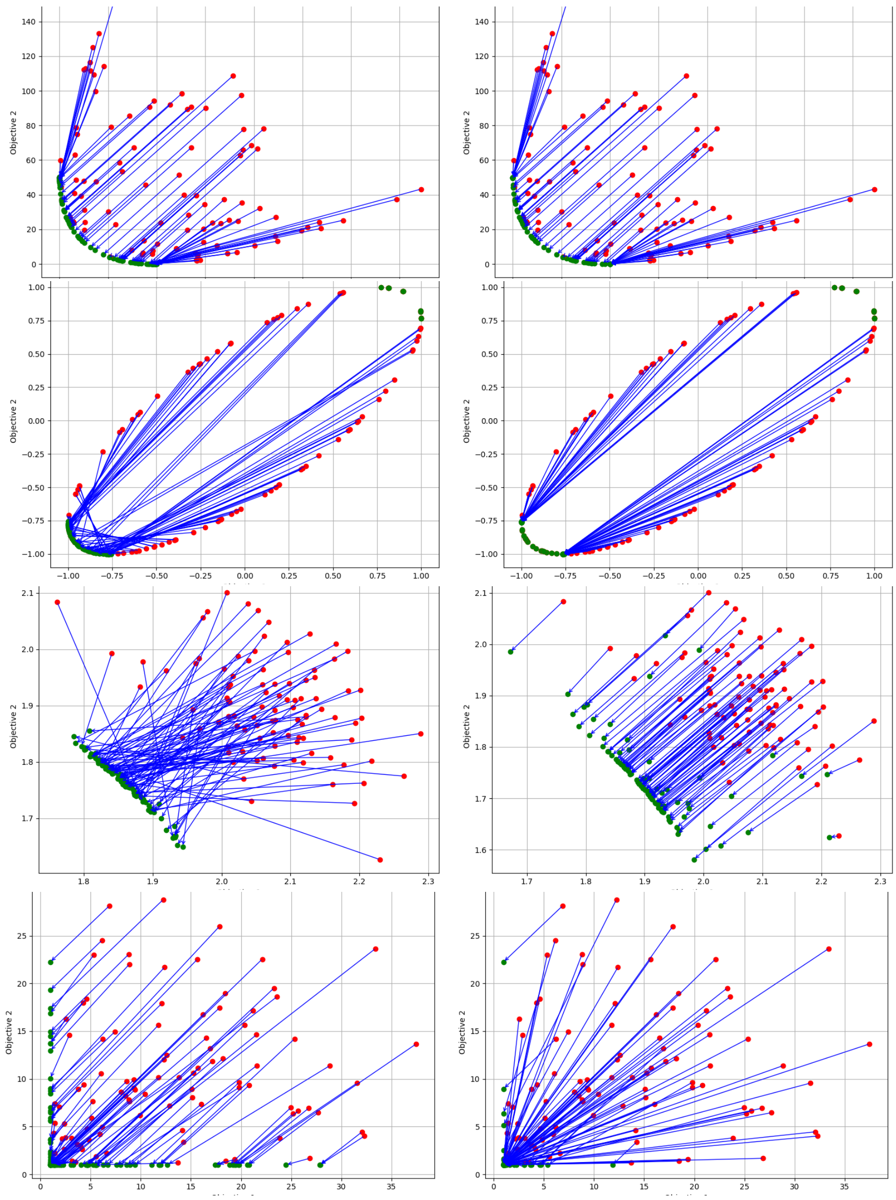

Fig. 2 Recovered Pareto fronts from 100 random starting points. From top to bottom: BK1, DGO1, QV1, Toi4b. Left: F-AMG, Right: Armijo-SD.

5.2 PDE Multiobjective Problem

We also evaluated the algorithms on a PDE-constrained multiobjective optimization problem [7], which is particularly challenging due to the high computational cost of function and gradient evaluations.

Let $\Omega = (0,1)^2 \subset \mathbb{R}^2$ denote the unit square and $\partial\Omega$ its boundary. For a given control function $y : \Omega \to \mathbb{R}$, we define the state $u : \Omega \to \mathbb{R}$ as the weak solution of the boundary value problem

$$-\Delta u = y \quad \text{in } \Omega, \; u \quad = 0 \quad \text{on } \partial\Omega, \tag{55}$$

where $\Delta = \partial^2/\partial x_1^2 + \partial^2/\partial x_2^2$ denotes the Laplace operator. Problem (55) is discretized using the finite element method (FEM) with piecewise linear Lagrange basis functions on a uniform triangular mesh of Ω.

The aim is to minimize two competing objective functions. The first one measures the mismatch between the state u and a desired state

$$u_d(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2), \quad (x_1, x_2) \in \Omega,$$

while the second one penalizes the energy of the control. More precisely, for a finite-dimensional control vector $y \in \mathbb{R}^n$ (the FEM coefficients of the control function), we define the objectives

$$\begin{aligned} F_1(y) &= \tfrac{1}{2}\int_\Omega \big(u(y)(x) - u_d(x)\big)^2 \, dx, \\ F_2(y) &= \tfrac{1}{2}\int_\Omega y(x)^2 \, dx, \end{aligned} \tag{56}$$

where $u(y)$ denotes the FEM solution of (55) corresponding to control y.

The resulting optimization problem is the unconstrained multiobjective problem

$$\min_{y \in \mathbb{R}^n} \; F(y) = \big(f_1(y), f_2(y)\big). \tag{57}$$

The PDE was discretized by the finite element method using piecewise linear Lagrange basis functions on a uniform 16×16 triangular mesh, resulting in (n = 289) control variables. All finite element formulations and numerical solutions (used to evaluate F and its Jacobian) were implemented in Python using the DOLFIN interface of the FEniCS library [23]. We generated 70 random Gaussian initial points $y_0 \in \mathbb{R}^{289}$. We consider the stopping criterion $|\theta^k| \leq 10^{-2}$, and set a maximum number of iterations to 500.

For each algorithm, we recorded the average number of iterations, function evaluations, and gradient evaluations. A representative summary is reported in Table 5.2. All algorithms successfully solved the problem instances. Notably, F-AMG achieved solutions of comparable quality to those obtained by the other methods while requiring the fewest iterations and gradient evaluationsand without any function evaluations. This highlights the efficiency and practical appeal of the proposed method, especially for computationally expensive problems such as PDE-constrained optimization. Figure 3 displays the recovered Pareto fronts of problem (57) obtained from 70 random starting points. Visually, the Pareto fronts recovered by all algorithms appear very similar, indicating that each method approximates the true Pareto front with comparable accuracy.

Algorithm	Iterations	Func. evals	Grad. evals
Fixed-SD	71.0	-	72.0
C-AMG	155.7	-	156.7
F-AMG	33.9	-	34.9
Armijo-SD	71.0	72.0	72.0

Table 2 Average performance over 70 runs (dimension = 289)

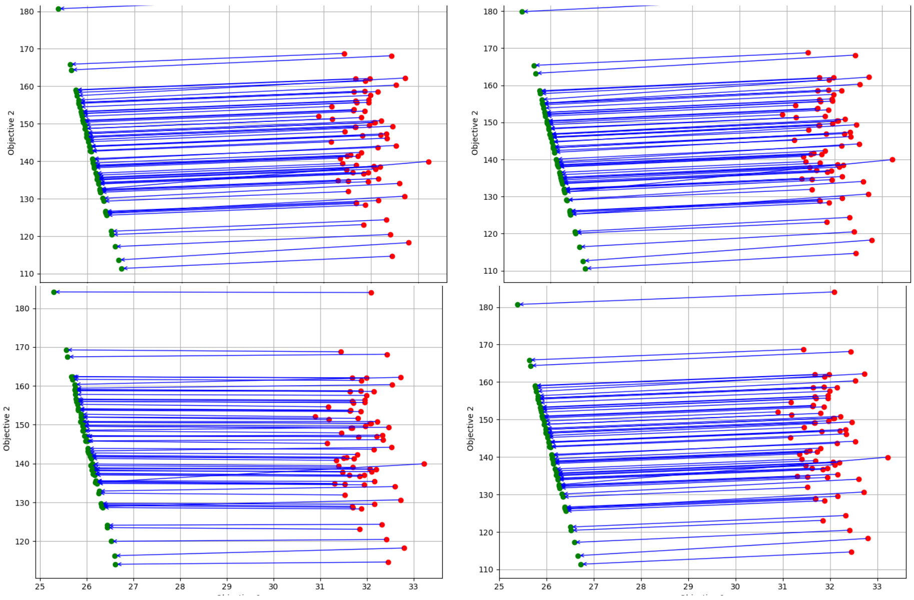

Fig. 3 Recovered Pareto fronts of problem (57) from 70 random starting points. Top row: Fixed-SD and C-AMG. Bottom row: F-AMG and Armijo-SD.

6 Conclusion

This work introduced AMG, an adaptive line-search-free multiobjective gradient method for solving multiobjective optimization problems. The AMG method preserves key advantages of adaptive gradient schemesrobustness with respect to stepsize selection, independence from function evaluations, and low computational costwhile being tailored to the multiobjective setting. The algorithm admits two variants: a conservative variant with monotonically decreasing stepsizes and a flexible variant that allows occasional stepsize increases. Under standard Lipschitz continuity assumptions, we established iteration-complexity guarantees for both variants in the nonconvex setting and improved bounds for the conservative variant in the convex case. Numerical experiments show that the flexible AMG performs favorably compared to multiobjective steepest descent methods with fixed or Armijo stepsizes.

Acknowledgments

The works of M. L. N. Gonçalves and J. G. Melo were supported in part by CNPq (Grant No. 312223/2022-6), FAPEG (Grant No. 202510267001610), and FAPESC (Grant No. 2024TR002238). The work of G. N. Grapiglia was partially supported by the Fonds d'appui l'internationalisation (FAI), UCLouvain.

References

[1] A. W. A. Griewank. Evaluationg derivatives: Principles and techniques of algorithmic differentiation. SIAM, Philadelphia, 2008.

[2] P. B. Assunção, O. P. Ferreira, and L. F. Prudente. Conditional gradient method for multiobjective optimization. *Comput. Optim. Appl.*, 78:741–768, 2021.

[3] M. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon. Automatic differentiation of algorithms. *J. Comput. Appl. Math.*, 124(1):171–190, 2000.

[4] Y. Bello-Cruz, L. R. Lucambio Pérez, and J. G. Melo. Convergence of the projected gradient method for quasiconvex multiobjective optimization. *Nonlinear Anal. Theory Methods Appl.*, 74(16):5268 – 5273, 2011.

[5] Y. Bello-Cruz, J. G. Melo, and R. V. G. Serra. A proximal gradient splitting method for solving convex vector optimization problems. *Optimization*, 71(1):33–53, 2022.

[6] G. C. Bento, J. X. Cruz-Neto, and A. Soubeyran. A proximal point-type method for multicriteria optimization. *Set-Valued Var Anal.*, 22(3):557–573, 2014.

[7] M. Bernreuther, M. Dellnitz, B. Gebken, M. G., S. Peitz, K. Sonntag, and S. Volkwein. Multiobjective optimization of non-smooth pde-constrained problems. *arXiv preprint arXiv:2308.01113v1*, 2023.

[8] H. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM J. Optim.*, 15(4):953–970, 2005.

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *J. Mach. Learn. Res.*, volume 12, pages 2121–2159, 2011.

[10] H. A. Eschenauer, J. Koski, and A. Osyczka. *Multicriteria Optimization — Fundamentals and Motivation*, pages 1–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990.

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

[12] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Method. Oper. Res.*, 51(3):479–494, 2000.

[13] E. H. Fukuda and L. M. G. Drummond. Inexact projected gradient method for vector optimization. *Comput. Optim. Appl.*, 54(3):473–493, 2013.

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

[15] E. H. Fukuda, H. Tanabe, and N. Yamashita. Proximal gradient methods for multiobjective optimization and their applications. *Comput. Optim. Appl.*, 72(2):339–361, 2019.

[16] E. H. Fukuda, H. Tanabe, and N. Yamashita. An accelerated proximal gradient method for multiobjective optimization. *Comput. Optim. Appl.*, 86(2):421–455, 2023.

[17] E. H. Fukuda, H. Tanabe, and N. Yamashita. New merit functions for multiobjective optimization and their properties. *Optimization*, 73(13):3821–3858, 2024.

[18] D. S. Gonçalves, M. L. N. Gonçalves, and J. G. Melo. An away-step Frank-Wolfe algorithm for constrained multiobjective optimization. *Comput. Optim. Appl.*, 88(2):759–781, 2024.

[19] D. S. Gonçalves, M. L. N. Gonçalves, and J. G. Melo. Improved convergence rates for the multiobjective Frank-Wolfe method. *J. Optim. Theory Appl.*, 205(20):1–25, 2025.

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

[21] L. M. Graña Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *J. Comput. Appl. Math.*, 175(2):395 – 414, 2005.

[22] G. Grapiglia and G. Stella. An adaptive trust-region method without function evaluations. *Comput. Optim. Appl.*, 82(1):31–60, 2022.

[23] A. Logg, K.-A. Mardal, G. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, volume 84 of *Lecture Notes in Computational Science and Engineering.* Springer, 2012.

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

[25] H. B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. In *Conference on Learning Theory (COLT)*, pages 244–256, 2010.

[26] E. Schreibmann, M. Lahanas, L. Xing, and D. Baltas. Multiobjective evolutionary optimization of the number of beams, their orientations and weights for intensity-modulated radiation therapy. *Physics in medicine and biology*, 49:747–70, 04 2004.

[27] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscape. *J. Mach. Learn. Res.*, 21(1):1–30, 2020.