New adaptive proximal gradient algorithms for solving multiobjective composite optimization problems

Pham Thi Hoai[a,*], Pham Thi Khanh[a], Jen-Chih Yao[b]

[a]Faculty of Mathematics and Informatics, Hanoi University of Science and Technology, 1 Dai Co Viet Road, Hanoi, Vietnam
[b]Research Center for Interneural Computing, China Medical University Hospital, China Medical University, Taichung, Taiwan

Abstract

In this paper, we propose new adaptive proximal gradient algorithms to solve multiobjective optimization problems, where each objective function is the sum of a differentiable function and a proper, closed, convex function. Utilizing the local behavior of the differentiable terms we propose new adaptive ways to select stepsizes used in proximal gradient scheme. In particular, the adaptive stepsizes are used in proximal subproblems whose optimal solution define the next iterates. We provide the convergence results for both nonconvex and convex cases of the considered problems. In particular, for a large class of nonconvex multiobjective composite optimization problems (where the differentiable terms are *Quacavex* including convex, concave, indefinite quadratic, or indefinite quadratic over positive linear functions), we show that any cluster point of the iterates is Pareto stationary. If all objectives are convex, we prove the convergence of the sequence of iterates to a weakly Pareto optimal solution of the considered problem. From a finite step, the standard theoretical convergence rates of the proposed algorithms are established at $\mathcal{O}(1/\sqrt{K})$; $\mathcal{O}(1/K)$ and Q-linear for the nonconvex; convex and strongly convex settings, respectively. Notably, multiobjective proximal gradient algorithms in the literature commonly use stepsizes chosen by line search backtracking procedures or/and fixed stepsizes that requires the knowledge of the global Lipschitz gradient constant. The empirical performance is evaluated through practical models including robust optimization and supervised feature selection for benchmark datasets. Experimental results demonstrate that our proposed algorithm significantly outperforms existing methods, especially in terms of processing time.

Keywords: nonconvex multiobjective programming, multiobjective composite optimization problems, proximal gradient method, nonsmooth multiobjective programming, convex multiobjective programming

1. Introduction

In real life, there are a lot of practical models simultaneously minimizing many objective functions which are commonly conflicting, see, e.g. [1] and the references therein. These problems belong to the class of multiobjective programming that has received a lot of attention from mathematicians in particular and scientists in general. In this paper, we consider one of the most important multiobjective optimization problems of the form

$$\min_{x\in\mathbb{R}^n} F(x) = (F_1(x), \dots, F_m(x)), \tag{P}$$

*Corresponding author
Email addresses: hoai.phamthi@hust.edu.vn (Pham Thi Hoai), khanh207111@gmail.com (Pham Thi Khanh), yaojc@mail.cmu.edu.tw (Jen-Chih Yao)

where all components $F_j : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$, for $j = 1, \dots, m$, have the following structure:

$$F_j(x) := G_j(x) + H_j(x), \quad j = 1, \dots, m, \tag{1.1}$$

and satisfy the following assumption.

Assumption 1. *(i)* $G_j : \mathbb{R}^n \to \mathbb{R}$ *is differentiable and has a globally Lipschitz gradient with constant* $L_j > 0$, *for all* $j = 1, \dots, m$;

(ii) $H_j : \mathbb{R}^n \to \bar{\mathbb{R}}$ *is proper, closed and convex, for all* $j = 1, \dots, m$.

Notably, Problem (P) has many important applications in various fields, specifically robust optimization, supervised feature selection, compressed sensing, machine learning, signal and image processing, approximation theory, and statistics; see, e.g., [2, 3, 4, 5, 6] and the references therein.

To address composite multiobjective optimization problems (P), one of the conventional approaches is based on multiobjective proximal gradient methods. In the pioneering studies on this research direction, Tanabe et al. [2] introduced two algorithms. The first one is *MPG-Normal* using fixed stepsizes within $(0, 2/L)$ (where $L = \max_{j=1,\dots,m} L_j$) for computing multiobjective proximal gradient operator that defines the next iterate at each step. The second one, *MPG-Armijo* using an arbitrary constant stepsize inside the proximal gradient operator and integrating Armijo-type line search backtracking for finding exogenous stepsizes to determine the next iterate. Under the boundedness condition of lower level sets, the authors established the global convergence of both algorithms by showing that every cluster point of the generated sequence $\{x^k\}$ is a Pareto stationary point for the nonconvex case of Problem (P). To complete the theoretical framework, Tanabe et al. [7] provided a detailed convergence rate analysis for these algorithms, specifically achieving $\mathcal{O}(1/\sqrt{k})$ for non-convex problems, $\mathcal{O}(1/k)$ for convex problems, and a linear convergence rate for the strongly convex case. Subsequently, to enhance computational performance, Tanabe et al. [8] proposed the accelerated version of MPG-Normal algorithm that is *MPG-Accelerated* by extending the FISTA framework from single-objective [9] to multiobjective optimization, which attains a theoretical global convergence rate of $\mathcal{O}(1/k^2)$.

Following this line of research, Bello-Cruz et al. [10] introduced the *MPG-Implicit* algorithm, in which a backtracking line search figures out the suitable stepsize used inside the proximal subproblem whose optimal solution determines the next iterate. Although *MPG-Implicit* generates the sequences that converges to weak Pareto solution but it suffers from a significant limitation: repeatedly solving the proximal subproblem if the descent condition is not satisfied. To overcome this drawback, Bello-Cruz et al. [11] proposed *MPGE* algorithm, replacing the implicit search procedure with an explicit one so that only a single subproblem needs to be solved at each iteration. The stepsize used in the subproblem is fixed and the exogenous stepsize is chosen by an explicit line search backtracking. Notably, both *MPG-Implicit* and *MPGE* are applied for convex case of Problem (P) and achieve a sublinear convergence rate of $\mathcal{O}(1/k)$ to a weak Pareto optimal solution under the assumption that every monotonically decreasing sequence in the image set of the objective function is bounded below. Considering the same class of convex problems (P), Zhao et al. [12] proposed *PGM* - a multiobjective proximal algorithm with a new line search backtracking condition to find stepsizes used in the proximal subproblems. This utilize the information of the gradient of the differentiable terms without recalculate the value of the objective functions along with line search process like traditional backtracking methods. Besides, they proposed an accelerated version of PGM that is accPGM which incorporates a Nesterov momentum strategy [13] to attain a theoretical rate of $\mathcal{O}(1/k^2)$ while PGM achieves a convergence rate of $\mathcal{O}(1/k)$. Their experiments indicate that although these two methods require higher computational time than MPG-Armijo and MPG-Accelerated, they deliver superior solution quality [12]. It is worth noting that, except for MPG-Normal and MPG-Armijo, a key theoretical limitation in MPG-Accelerated, MPG-Implicit, MPGE, PGM, and accPGM is the requirement regarding the convexity of the component functions G_j. However, another multiobjective proximal algorithm, termed BBPGMO proposed by Chen et al. [14], is capable of handling the nonconvex case of (P). BBPGMO incorporates Barzilai-Borwein stepsize inside the proximal subproblem and Armijo line search backtracking for finding exogenous stepsizes. The convergence results of BBPGMO are $O(1/\sqrt{k})$, $O(1/k)$ and linear rates for nonconvex, convex, and strongly convex cases of (P), respectively.

Nevertheless, all the algorithms mentioned above use a fixed stepsize or line search backtracking to select stepsize inside or/and outside proximal subproblems, and therefore suffering the expensive cost to process along with the algorithms.

Main contributions: In this paper, we propose new adaptive proximal gradient algorithms corresponding to new adaptive strategies for selecting stepsizes used in proximal subproblems whose optimal solutions define the next iterates. No line search or knowledge of Lipschitz constants is required during the processing of our new method. We just utilize the local curvature of the differentiable terms to figure out the stepsizes. Under Assumption 1 and there exists an objective being lower bounded, we obtain the standard convergence results for a large class of nonconvex (P). In particular, if all $G_j,\ j = 1, ..., m$ are Quacavex[1] functions (which include convex, concave, indefinite quadratic and indefinite quadratic over positive affine functions), we prove that any cluster point of the generated sequence is a stationary point of problem (P) and from a fixed iteration, the computational complexity is $O(1/\sqrt{K})$. Under Assumption 1 together with the additional assumption that every monotonically decreasing sequence in the image set of the objective function is bounded from below, we show that the iterates converges to a weakly Pareto optimal solution of Problem (P) with convergence rate $O(1/K)$ from a finite step if and if $G_j,\ j = 1, ..., m$ are convex. Next, if $G_j,\ j = 1, ..., m$ are strongly convex we provide the linear convergence of the generated sequence to a Pareto optimal solution of Problem (P). The efficiency of our new method is demonstrated by numerical experiments for various test instances including robust optimization multiobjective problems and biobjective supervised feature selection problems with benchmark datasets.

Organization of the paper: Section 2 provides essential definitions and preliminary results used throughout the paper. In Section 3, we propose two new adaptive multiobjective proximal gradient algorithms. The convergence properties and rates of the proposed algorithms are rigorously analyzed for different cases of the component G_j. Specifically, the non-convex, convex, and strongly convex cases of G_j are addressed in Sections 4, 5, and 6, respectively. Numerical experiments in Section 7 compare the proposed algorithms with three state-of-the-art proximal-type algorithms recently introduced in the literature. Finally, Section 8 concludes the paper.

2. Preliminaries

In this paper, we use $\mathbb{R}_+$ and $\mathbb{R}_{++}$ to denote the sets of nonnegative and positive real numbers, respectively. The notations $\mathbb{R}^n_+ := \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$ and $\mathbb{R}^n_{++} := \mathbb{R}_{++} \times \cdots \times \mathbb{R}_{++}$ correspond to the nonnegative orthant and the positive orthant of $\mathbb{R}^n$. We consider the partial order induced by $\mathbb{R}^n_+$: for $u, v \in \mathbb{R}^n$, $u \leq v$ if $v - u \in \mathbb{R}^n_+$, and a stronger relation: $u < v$ if $v - u \in \mathbb{R}^n_{++}$. In other words, $u \leq v$ means that $u_i \leq v_i$ for all $i = 1, \ldots, n$, and $u < v$ also represents $u_i < v_i$ for all $i = 1, \ldots, n$. Next, we recall some standard definitions from convex analysis that will be used extensively in the sequel.

Definition 2.1. *[16, Theorem 25.1]* Let $h : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a function, and let $\mathrm{dom}(h) := \{x \in \mathbb{R}^n : h(x) < +\infty\}$ denote its effective domain. The function h is said to be proper if $\mathrm{dom}(h) \neq \emptyset$.

[1]The definition of Quacavex function can be found in [15].

(i) For a given $x \in \mathrm{dom}(h)$ and a direction $d \in \mathbb{R}^n$, the directional derivative of h at x in the direction d is defined as

$$h'(x; d) := \lim_{\alpha \searrow 0} \frac{h(x + \alpha d) - h(x)}{\alpha}.$$

When h is differentiable at x, it is easy to see that $h'(x; d) = \nabla h(x)^\top d$, where $\nabla h(x)$ denotes the gradient of h at x.

(ii) Let $h : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper convex function, and let $\bar{x} \in \mathrm{dom}(h)$. The subdifferential of h at $\bar{x}$ is defined as the set

$$\partial h(\bar{x}) := \{v \in \mathbb{R}^n : h(\bar{x}) + \langle v, x - \bar{x} \rangle \leq h(x), \quad \forall x \in \mathbb{R}^n\}. \tag{2.1}$$

If h is differentiable at $\bar{x}$, then $\partial h(\bar{x}) = \{\nabla h(\bar{x})\}$.

Now, we recall the concepts of optimality for the multiobjective optimization problem (P).

Definition 2.2. *[17, 18, 2]*

(i) A vector $x^* \in \mathrm{dom}(F) = \bigcap_{j=1,\dots,m} \mathrm{dom} F_j$ is called a *Pareto optimal* solution of Problem (P) if there does not exist $x \in \mathrm{dom}(F)$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$.

(ii) A vector $x^* \in \mathrm{dom}(F)$ is called a *weakly Pareto optimal* solution of Problem (P) if there does not exist $x \in \mathrm{dom}(F)$ such that $F(x) < F(x^*)$.

(iii) A vector $x^* \in \mathrm{dom}(F)$ is called a *Pareto stationary* (or *critical*) solution of Problem (P) if, for all $d \in \mathbb{R}^n$, $\max_{j=1,\dots,m} F_j'(x^*; d) \geq 0$.

The following lemma establishes the relationships between Pareto stationarity and Pareto optimality under varying convexity assumptions.

Lemma 2.3. *[2, Lemma 2.2]*

(i) *If $x \in \mathrm{dom}(F)$ is a weakly Pareto optimal point of F, then x is Pareto stationary.*

(ii) *If all components F_j of F are convex and $x \in \mathrm{dom}(F)$ is a Pareto stationary point of F, then x is weakly Pareto optimal.*

(iii) *If all components F_j of F are strictly convex and $x \in \mathrm{dom}(F)$ is a Pareto stationary point of F, then x is Pareto optimal.*

One of the most typical methods to address Problem (P) is based on multiobjective proximal gradient scheme. Specifically, $\psi_x(\cdot) : \mathbb{R}^n \to \bar{\mathbb{R}}$ is constructed to approximate the simultaneous variation of the objective functions around the current point x:

$$\psi_x(u) := \max_{j=1,\dots,m} \{\langle \nabla G_j(x), u - x \rangle + H_j(u) - H_j(x)\} \quad \text{for all } u \in \mathbb{R}^n. \tag{2.2}$$

Then, the next iterate is defined as the unique optimal solution of the following subproblem:

$$\min_{u \in \mathbb{R}^n} \quad \varphi_{x,t}(u) = \psi_x(u) + \frac{1}{2t}\|u - x\|^2, \quad t > 0. \tag{2.3}$$

In particular, we can take the next iterate as $\rho(x, t)$, i.e.,

$$\rho(x, t) := \arg\min_{u \in \mathbb{R}^n} \varphi_{x,t}(u). \tag{2.4}$$

Setting

$$\theta(x,t) := \varphi_{x,t}(\rho(x,t)). \tag{2.5}$$

The relationship between $\rho(x,t), \theta(x,t)$ and the Pareto stationarity of x is stated in the following lemma.

Lemma 2.4. *[2, Lemma 3.2] The following statements hold:*

(i) $\theta(x,t) \leq 0$ *for all* $x \in \mathrm{dom}(F)$.

(ii) *The following statements are equivalent:*

a. x *is a Pareto stationary point (i.e., weakly Pareto point when* F *is convex) of Problem* (P)*;*

b. $\theta(x,t) = 0$*;*

c. $\rho(x,t) = x$.

(iii) $\rho(\cdot,t)$ *and* $\theta(\cdot,t)$ *are continuous.*

The monotone of the residual $\|x-\rho(x,t)\|$ with respect to t is presented in the following lemma.

Lemma 2.5. *[10, Lemma 3.2] For any fixed point* $x \in \mathrm{dom}(F)$ *and* $t_2 \geq t_1 > 0$*, we have*

$$\frac{t_2}{t_1}\|x-\rho(x,t_1)\| \geq \|x-\rho(x,t_2)\| \geq \|x-\rho(x,t_1)\|.$$

We also need the concept of a quasi-Fejér sequence, which will be useful for deriving the convergence results of our new algorithms in the upcoming sections.

Definition 2.6. *[19]* A sequence $\{y^k\}$ is quasi-Fejér convergent to a set $U \subseteq \mathbb{R}^n$ if for every $u \in U$ there exists a sequence $\{\epsilon_k\} \subseteq \mathbb{R}_+$ such that $\sum_{k=0}^{+\infty} \epsilon_k < +\infty$ and

$$\|y^{k+1}-u\|^2 \leq \|y^k-u\|^2 + \epsilon_k \quad \text{for all } k \geq 0.$$

Lemma 2.7. *[19] If* $\{y^k\}$ *is quasi-Fejér convergent to a nonempty set* $U \subseteq \mathbb{R}^n$*, then* $\{y^k\}$ *is bounded. Furthermore, if a cluster point* y *of* $\{y^k\}$ *belongs to* U*, then* $\lim_{k\to+\infty} y^k = y$.

3. New adaptive proximal gradient algorithms based on proximal gradient scheme for multiobjective optimization problems

Definition 3.1. For any $x, y \in \mathbb{R}^n$, we construct a function

$$B_\ell(x,y) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \quad \ell = 1,2,$$

defined by

$$B_1(x,y) = \max_{j=1,\ldots,m} |\langle \nabla G_j(x) - \nabla G_j(y), x-y \rangle|$$

and

$$B_2(x,y) = \left(\max_{j=1,\ldots,m} \|\nabla G_j(x) - \nabla G_j(y)\| \right) \|x-y\|.$$

We directly establish a property of $B_\ell(x,y)$ in the following lemma.

Lemma 3.2. *Suppose that Problem* (P) *satisfies Assumption 1. Then, for $\ell = 1, 2$, the following inequalities hold.*

$$B_1(x, y) \le B_2(x, y) \le L\|x - y\|^2, \tag{3.1}$$

where $L = \max_{j=1,\ldots,m} L_j$.

Proof. For the left-hand side inequality, one can use some simple evaluations as follow.

$$\begin{aligned} B_1(x, y) &= \max_{j=1,\ldots,m} |\langle \nabla G_j(x) - \nabla G_j(y), x - y\rangle| \le \max_{j=1,\ldots,m} \|\nabla G_j(x) - \nabla G_j(y)\|\|x - y\| \\ &= \left(\max_{j=1,\ldots,m} \|\nabla G_j(x) - \nabla G_j(y)\|\right) \|x - y\| \\ &= B_2(x, y). \end{aligned}$$

For the right-hand side inequality, the proof is based on the global Lipschitz continuity of ∇G_j with constant L_j from Assumption 1 and $L = \max_{j=1,\ldots,m} L_j$. □

In what follows, we propose new proximal gradient algorithms to solve Problem (P). In this framework, the function $B_\ell(x, y)$ plays an important role in determining our stepsizes.

Algorithm 3.1 AMPGℓ (Adaptive Multiobjective Proximal Gradient algorithm)

Preparation: Choose $\ell \in \{1, 2\}$ and use $B_\ell(x, y)$ as defined in Definition 3.1. The corresponding algorithm is then denoted by AMPGℓ.

Step 0. Choose $x^0 \in \mathrm{dom}(F)$, set $k = 0$, select $t_0 > 0$, $0 < c_1 < c_0 < \frac{1}{2}$, $\varepsilon > 0$, and a convergent positive series $\sum_{k=0}^{+\infty} \varepsilon_k$.

Step 1.

$$\begin{aligned} &\textbf{Compute} \quad x^{k+1} := \rho(x^k, t_k) \text{ as in (2.4).} \\ &\textbf{If} \quad \|x^{k+1} - x^k\| < \varepsilon \quad \textbf{then} \quad \text{STOP} \\ &\textbf{else} \quad \text{go to } \textbf{Step 2.} \end{aligned} \tag{3.2}$$

Step 2.

$$\begin{aligned} &\textbf{If} \quad t_k > c_0 \frac{\|x^{k+1} - x^k\|^2}{B_\ell(x^{k+1}, x^k)} \\ &\textbf{then} \quad t_{k+1} = c_1 \frac{\|x^{k+1} - x^k\|^2}{B_\ell(x^{k+1}, x^k)} \\ &\textbf{else} \quad t_{k+1} = (1 + \varepsilon_k) t_k \\ &\qquad k := k + 1 \text{ and return to } \textbf{Step1}. \end{aligned} \tag{3.3}$$

To investigate the convergence of our new methods, it is necessary to first analyze several fundamental properties of the new stepsize sequence.

Lemma 3.3. *For Algorithm 3.1 - AMPGℓ, $\ell = 1, 2$, we have $\inf_{k\geq 0} t_k > 0$ and $\{t_k\}$ converges to a positive limit t^*.*

Proof. We divide the proof into two parts:

(i) **Part 1:** For $k = 0$, following Algorithm 3.1 - AMPGℓ, if (3.3) is true, i.e.,

$$B_\ell(x^1, x^0) > c_0 \frac{\|x^1 - x^0\|^2}{t_0}$$

then

$$t_1 = c_1 \frac{\|x^1 - x^0\|^2}{B_\ell(x^1, x^0)} \underset{\text{by (3.1)}}{\geq} c_1 \frac{\|x^1 - x^0\|^2}{L\|x^1 - x^0\|^2} = \frac{c_1}{L}.$$

Otherwise, we have $t_1 = (1 + \varepsilon_0)t_0 > t_0$. Therefore $t_1 \geq \min\left\{\frac{c_1}{L}, t_0\right\} > 0$. By an analogous argument we derive that $t_{k+1} \geq \min\left\{\frac{c_1}{L}, t_k\right\} \geq \min\left\{\frac{c_1}{L}, t_0\right\}$ for all $k \geq 0$. It follows that $\inf_{k\geq 0} t_k \geq \min\left\{\frac{c_1}{L}, t_0\right\} = t_{\min} > 0$.

(ii) **Part 2:** Let $u_k = \ln t_{k+1} - \ln t_k$ for $k \geq 0$. Then, we have $u_k = u_k^+ - u_k^-$, where

$$u_k^+ = \max\{0, u_k\}, \quad u_k^- = -\min\{0, u_k\}.$$

Then $u_k^+ \geq 0$ and $u_k^- \geq 0$ for all $k \geq 0$. From the definition of t_k in Algorithm 3.1 - AMPGℓ, we derive that

$$u_k = \ln \frac{t_{k+1}}{t_k} \leq \ln(1 + \varepsilon_k) \leq \varepsilon_k, \quad \forall k \geq 0,$$

which implies $u_k^+ \leq \varepsilon_k$. Since $\sum_{k=0}^{+\infty} \varepsilon_k$ is convergent, we obtain $\sum_{k=0}^{+\infty} u_k^+ < +\infty$. Observing that $\sum_{k=0}^{+\infty} u_k^-$ is a nonnegative series and using the following relation:

$$\ln t_{k+1} - \ln t_0 = \sum_{i=0}^{k} u_i = \sum_{i=0}^{k}(u_i^+ - u_i^-) = \sum_{i=0}^{k} u_i^+ - \sum_{i=0}^{k} u_i^-, \tag{3.4}$$

we assert that if $\lim_{k\to+\infty} \sum_{i=0}^{k} u_i^- = +\infty$ then $\lim_{k\to+\infty} (\ln t_{k+1}) = -\infty \iff \lim_{k\to+\infty} t_k = 0$.

From **Part 1**, we have $\inf_{k\geq 0} t_k > 0$. This contradiction proves the convergence of $\sum_{k=0}^{+\infty} u_k^-$. Finally, from (3.4) we get the desired conclusion that $\lim_{k\to+\infty} t_k = t^* < +\infty$.

$\square$

Lemma 3.4. *For each $\ell = 1, 2$, let $\{x^k\}$ be sequence generated by Algorithm 3.1 - AMPGℓ. Then there exists an index $\hat{k}_\ell$ such that*

$$B_\ell(x^{k+1}, x^k) \leq c_0 \frac{\|x^{k+1} - x^k\|^2}{t_k}, \ \textit{ for all } k \geq \hat{k}_\ell.$$

Proof. Suppose by contradiction that there exists a subsequence of indices $\{k_j\}$ with $k_j \to +\infty$ such that

$$B_\ell(x^{k_j+1}, x^{k_j}) > c_0 \frac{\|x^{k_j+1} - x^{k_j}\|^2}{t_{k_j}}.$$

For this case,

$$t_{k_j+1} = c_1 \frac{\|x^{k_j+1} - x^{k_j}\|^2}{B_\ell(x^{k_j+1}, x^{k_j})}.$$

Consequently,

$$c_1 \frac{\|x^{k_j+1} - x^{k_j}\|^2}{t_{k_j+1}} = B_\ell(x^{k_j+1}, x^{k_j}) > c_0 \frac{\|x^{k_j+1} - x^{k_j}\|^2}{t_{k_j}},$$

i.e., $\dfrac{t_{k_j+1}}{t_{k_j}} < \dfrac{c_1}{c_0}$ for all $j \in \{1, \ldots, m\}$. On the other hand, from Lemma 3.3 we have

$$\lim_{k_j \to +\infty} t_{k_j+1} = \lim_{k_j \to +\infty} t_{k_j} = \lim_{k \to +\infty} t_k = t^*$$

which implies that $\dfrac{t^*}{t^*} \leq \dfrac{c_1}{c_0} < 1$. This is a contradiction, thereby completing the proof. $\square$

Remark 3.5. From Lemma 3.3 and Lemma 3.4 we obtain the following relations:

(i) $t_k \geq t_{\min}$ for all $k \geq 0$;

(ii) $t_{\min} \leq t_{\hat{k}_\ell} \leq t_k \leq t_{k+1} \leq t^*$ for all $k \geq \hat{k}_\ell$.

4. The convergence of Algorithm 3.1-AMPGℓ for nonconvex case of Problem (P)

Assumption 2. Suppose that for every $j \in \{1, \ldots, m\}$, the function G_j is *Quacavex*, i.e., for $u, v \in \operatorname{dom}(F)$, the function $D_j^{uv} : [0,1] \to \mathbb{R}$ defined by

$$D_j^{uv}(t) = \langle \nabla G_j(u + t(v - u)), v - u \rangle$$

is quasiconvex on $[0, 1]$.

Remark 4.1. As presented in [20, 21, 15], the class of Quacavex functions includes convex, concave, indefinite quadratic and indefinite quadratic over positive affine functions.

Below are some examples of Quacavex functions (see e.g., [20]).

Example 4.2. For every $j \in \{1, ..., m\}$ we can use G_j as the following to satisfy Assumption 2.

(i) If G_j is convex (or concave), then, the convexity (concavity, resp.) of G_j implies the convexity (concavity, resp.) of $G_j(u + t(v - u))$ on the set $\{t \in \mathbb{R} \mid u + t(v - u) \in \operatorname{dom}(G_j)\} \supset [0, 1]$ (since $\operatorname{dom}(G_j)$ is a convex set). Consequently, $(G_j(u + t(v - u)))'_t = D_j^{uv}(t)$ is increasing (decreasing, resp.) over $[0, 1]$, and thus $D_j^{uv}(t)$ is quasiconvex on $t \in [0, 1]$.

(ii) For the indefinite quadratic case of G_j, i.e., $G_j(x) = \frac{1}{2}x^\top Ax + b^\top x$ (where A is a symmetric matrix in $\mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$). Thus, $D_j^{uv}(t) = \langle \nabla A(u + t(v - u)) + b, v - u \rangle$ is a linear function, and therefore it is quasiconvex on $[0, 1]$ for any $u, v \in \mathbb{R}^n$.

(iii) Problem (P) with:

1. $G_j(x) = \dfrac{x^\top Ax + b^\top x + c}{p^\top x + q}$, where A is a real symmetric matrix in $\mathbb{R}^{n\times n}$, $b, p \in \mathbb{R}^n$, and $c, q \in \mathbb{R}$;
2. $H_j = i_{\mathcal{C}}$ (the indicator function of $\mathcal{C}$, where $\mathcal{C} \subset \mathbb{R}^n$ is a closed and convex set such that $p^\top x + q > 0$ for all $x \in \mathcal{C}$).

Lemma 4.3. *For all $k \geq \hat{k}_\ell$ we have*

$$F_j(x^{k+1}) - F_j(x^k) \leq \varphi_{x^k,t_k}(x^{k+1}) + \left(c_0 - \frac{1}{2}\right) \frac{\|x^{k+1} - x^k\|^2}{t_k}, \text{ for all } j = 1, \ldots, m. \tag{4.1}$$

As a consequence, $\{F(x^k)\}_{k\geq\hat{k}_\ell}$ is decreasing.

Proof. By the fundamental theorem of calculus, for each $j \in \{1, \ldots, m\}$, we have

$$\begin{aligned} G_j(x^{k+1}) - G_j(x^k) &= \int_0^1 \left\langle \nabla G_j(x^k + t(x^{k+1} - x^k)), x^{k+1} - x^k \right\rangle dt \\ &= \left\langle \nabla G_j(x^k), x^{k+1} - x^k \right\rangle + \int_0^1 u_j^k(t)\, dt, \end{aligned} \tag{4.2}$$

where, for $t \in [0, 1]$,

$$\begin{aligned} u_j^k(t) &= \left\langle \nabla G_j(x^k + t(x^{k+1} - x^k)) - \nabla G_j(x^k), x^{k+1} - x^k \right\rangle \\ &= \left\langle \nabla G_j(x^k + t(x^{k+1} - x^k)), x^{k+1} - x^k \right\rangle - \left\langle \nabla G_j(x^k), x^{k+1} - x^k \right\rangle \\ &= D_j^{x^k x^{k+1}}(t) - \left\langle \nabla G_j(x^k), x^{k+1} - x^k \right\rangle, \quad \text{for all } j = 1, \ldots, m. \end{aligned}$$

Because of the quasiconvexity of $D_j^{x^k x^{k+1}}$ defined in Assumption 2, for $t \in [0, 1]$, we have

$$\begin{aligned} u_j^k(t) &\leq \max\left\{u_j^k(0), u_j^k(1)\right\} = \max\left\{0, u_j^k(1)\right\} \\ &\leq |u_j^k(1)| = \left|\left\langle \nabla G_j(x^{k+1}) - \nabla G_j(x^k), x^{k+1} - x^k \right\rangle\right| \leq B_\ell(x^k, x^{k+1}) \\ &\overset{\text{Lemma 3.4}}{\leq} \frac{c_0\|x^{k+1} - x^k\|^2}{t_k}, \quad \text{for all } k \geq \hat{k}_\ell. \end{aligned} \tag{4.3}$$

From (4.3) we obtain that

$$\int_0^1 u_j^k(t)\, dt \leq \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2, \quad \text{for all } k \geq \hat{k}_\ell. \tag{4.4}$$

From (4.2) & (4.4), for all $k \geq \hat{k}_\ell$, we have

$$\begin{aligned} F_j(x^{k+1}) - F_j(x^k) &= G_j(x^{k+1}) - G_j(x^k) + H_j(x^{k+1}) - H_j(x^k) \\ &\leq \left\langle \nabla G_j(x^k), x^{k+1} - x^k \right\rangle + H_j(x^{k+1}) - H_j(x^k) + \frac{1}{2t_k}\|x^{k+1} - x^k\|^2 + \frac{c_0 - \frac{1}{2}}{t_k} \cdot \|x^{k+1} - x^k\|^2 \\ &\overset{(2.3)}{\leq} \varphi_{x^k,t_k}(x^{k+1}) + \left(c_0 - \frac{1}{2}\right) \cdot \frac{\|x^{k+1} - x^k\|^2}{t_k}, \quad \text{for all } j = 1, \ldots, m. \end{aligned}$$

which leads to (4.1).

From (2.5), (3.2) and Lemma 2.4 (i), we have

$$\varphi_{x^k,t_k}(x^{k+1}) = \varphi_{x^k,t_k}(\rho(x^k,t_k)) = \theta(x^k,t_k) \leq 0.$$

On the other hand, since $c_0 \in \left(0, \frac{1}{2}\right)$, it follows from (4.1) for any $j \in \{1,\dots,m\}$ we have that

$$F_j(x^k) - F_j(x^{k+1}) \geq \left|\theta(x^k,t_k)\right| + \left(\frac{1}{2} - c_0\right) \frac{\|x^{k+1}-x^k\|^2}{t_k} \geq 0, \quad \text{for all } k \geq \hat{k}_\ell. \tag{4.5}$$

We thus deduce that $\{F_j(x^k)\}_{k\geq \hat{k}_\ell}$ is monotonically decreasing for each $j \in \{1,\dots,m\}$. □

Theorem 4.4. *Suppose that Problem* (P) *satisfies Assumption 1 and Assumption 2 and there exists $j_0 \in \{1,\dots,m\}$ such that F_{j_0} is lower bounded on $\mathbb{R}^n$. Then the sequence $\{x^k\}$ generated by Algorithm 3.1 - AMPGℓ with $\ell = 1,2$ satisfies:*

- ***(i)*** *$F(x^{k+1}) < F(x^k)$ unless x^k is a Pareto stationary point of Problem* (P).
- ***(ii)*** *The series $\sum_{k=0}^{+\infty} \|x^{k+1}-x^k\|^2$ and $\sum_{k=0}^{+\infty} \left|\theta(x^k,t_k)\right|$ are convergent. Consequently, $\|x^{k+1}-x^k\| \to 0$ and $\left|\theta(x^k,t_k)\right| \to 0$ as $k \to +\infty$ and*

$$\min_{\hat{k}_\ell \leq k \leq K-1} \|x^{k+1}-x^k\| \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \quad K > \hat{k}_\ell.$$

- ***(iii)*** *Any accumulation point of $\{x^k\}$ is Pareto stationary to Problem* (P).

Proof. ***(i)*** Suppose that there exists $j_k \in \{1,\dots,m\}$ such that $F_{j_k}(x^k) = F_{j_k}(x^{k+1})$. Then, using (4.5) with $j = j_k$ we have $x^k = x^{k+1} = \rho(x^k,t_k)$. By Lemma 2.4(ii)-(c), x^k is a Pareto stationary point of Problem (P).

(ii) Now, utilizing the condition that F_{j_0} is bounded from below, we derive that the sequence $\{F_{j_0}(x^k)\}_{k\geq\hat{k}_\ell}$ is decreasing and bounded below, hence converging to a finite limit $\hat{F}_{j_0}$ as $k \to +\infty$. This implies that $\lim_{k\to+\infty}(F_{j_0}(x^k) - F_{j_0}(x^{k+1})) = 0$. Moreover, by Remark 3.5, we have $t_k \leq t^*$ for all $k \geq \hat{k}_\ell$, it follows from (4.5) that

$$F_{j_0}(x^k) - F_{j_0}(x^{k+1}) \geq \left|\theta(x^k,t_k)\right| + \frac{(1/2-c_0)}{t^*}\|x^{k+1}-x^k\|^2, \text{ for all } k \geq \hat{k}_\ell. \tag{4.6}$$

Taking the limit as $k \to +\infty$ in formula (4.6) we get that $\lim_{k\to+\infty}\left|\theta(x^k,t_k)\right| = 0$ and $\lim_{k\to+\infty}\|x^{k+1} - x^k\| = 0$. Next, summing up inequality (4.6) from $k = \hat{k}_\ell$ to $K-1 \geq \hat{k}_\ell$, yielding that

$$\begin{aligned}
\sum_{k=\hat{k}_\ell}^{K-1}\left|\theta(x^k,t_k)\right| + \frac{(1/2-c_0)}{t^*}\cdot\sum_{k=\hat{k}_\ell}^{K-1}\|x^{k+1}-x^k\|^2 &\leq \sum_{k=\hat{k}_\ell}^{K-1}\Big(F_{j_0}(x^k) - F_{j_0}(x^{k+1})\Big)\\
&= F_{j_0}(x^{\hat{k}_\ell}) - F_{j_0}(x^K)\\
&\leq F_{j_0}(x^{\hat{k}_\ell}) - \hat{F}_{j_0}.
\end{aligned} \tag{4.7}$$

Taking the limit as $K \to +\infty$ we obtain the convergence of $\sum_{k=\hat{k}_\ell}^{+\infty}\left|\theta(x^k,t_k)\right|$ and $\sum_{k=\hat{k}_\ell}^{+\infty}\|x^{k+1}-x^k\|^2$. Therefore, $\left|\theta(x^k,t_k)\right| \to 0$ as $k \to +\infty$. From inequality (4.7), we obtain

$$\sum_{k=\hat{k}_\ell}^{K-1}\|x^{k+1}-x^k\|^2 \leq \frac{t^*}{(1/2-c_0)}\cdot\left(F_{j_0}(x^{\hat{k}_\ell}) - \hat{F}_{j_0}\right),$$

thus,

$$\min_{\hat{k}_\ell \le k \le K-1} \|x^{k+1} - x^k\| \le \sqrt{\frac{t^*\left(F_{j_0}(x^{\hat{k}_\ell}) - \hat{F}_{j_0}\right)}{(1/2 - c_0)(K - \hat{k}_\ell)}} = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \ \forall K \ge \hat{k}_\ell + 1.$$

Apply (4.7) once again to derive that

$$\min_{\hat{k}_\ell \le k \le K-1} \left|\theta(x^k, t_k)\right| \le \frac{F_{j_0}(x^{\hat{k}_\ell}) - \hat{F}_{j_0}}{K - \hat{k}_\ell} = O\left(\frac{1}{K}\right), \ \forall K \ge \hat{k}_\ell + 1.$$

(iii) Suppose that $\bar{x}$ is a cluster point of $\{x^k\}$. Then there exists a subsequence $\{x^{k_j}\}$ such that $x^{k_j} \to \bar{x}$. Remember that from Remark 3.5 we have $t_k \ge t_{\min}$ for all $k \ge 0$; hence, from Lemma 2.5 and (3.2), we obtain that

$$\|x^k - \rho(x^k, t_{\min})\| \le \|x^k - \rho(x^k, t_k)\| = \|x^k - x^{k+1}\| \underset{k\to\infty}{\to} 0 \ \text{(by } (ii)).$$

We then derive that

$$\lim_{k\to+\infty} \|x^k - \rho(x^k, t_{\min})\| = 0.$$

Next, for all $j = 1, \dots, m$ we evaluate

$$\|\bar{x} - \rho(\bar{x}, t_{\min})\| \le \|\bar{x} - x^{k_j}\| + \|x^{k_j} - \rho(x^{k_j}, t_{\min})\| + \|\rho(x^{k_j}, t_{\min}) - \rho(\bar{x}, t_{\min})\|. \tag{4.8}$$

Taking the limit as $k_j \to +\infty$ we see that the right-hand side of (4.8) tends zero because $x^{k_j} \to \bar{x}$ and the continuity of $\rho(\cdot, t_{\min})$ (Lemma 2.4(iii)). Thus, $\|\bar{x} - \rho(\bar{x}, t_{\min})\| = 0$ which implies that $\bar{x}$ is a Pareto stationary point of Problem (P) by Lemma 2.4 (ii)-(c).

□

5. The convergence of Algorithm 3.1-AMPGℓ for the convex case of Problem (P)

In this section, we provide stronger convergence results for AMPGℓ when Problem (P) satisfies Assumption 1 and the following one.

Assumption 3. ***(i)*** The function G_j is convex for each $j = 1, \dots, m$.

(ii) All monotonically non-increasing sequences in the set $\mathrm{Im}(F)$ are bounded below by a point in $\mathrm{Im}(F)$, i.e., for every sequence $\{y^k\} \subset \mathrm{dom}(F)$ such that $F(y^{k+1}) \le F(y^k)$ for all $k \in \mathbb{N}$, there exists $\bar{y} \in \mathrm{dom}(F)$ such that $F(\bar{y}) \le F(y^k)$ for all $k \in \mathbb{N}$.

Remark 5.1. ***(i)*** Obviously, if Problem (P) satisfies Assumption 3(i), it also satisfies Assumption 2; hence we can inherit the results of Section 4 for this section.

(ii) Assumption 3(ii) is widely used in the related literature (e.g., [11, 22, 23, 24, 25, 26, 27, 28]). It is commonly employed to ensure the existence of efficient solutions for vector-valued optimization problems, see [29, Section 2.3]. In the scalar-valued unconstrained case, it is equivalent to ensuring the existence of an optimal point.

Lemma 5.2. *If Problem* (P) *satisfies Assumption 1 and Assumption 3, then for all $x \in \mathrm{dom}(F)$ and any $j = 1, \dots, m$, we have*

$$\begin{aligned} \|x^{k+1} - x\|^2 \le \|x^k - x\|^2 + 2t_k\left(F_j(x^k) - F_j(x^{k+1})\right) \\ + 2t_k \max_{i=1,\dots,m}\left(F_i(x) - F_i(x^k)\right) - (1 - 2c_0)\|x^k - x^{k+1}\|^2, \quad \textit{for all } k \ge \hat{k}_\ell. \end{aligned} \tag{5.1}$$

Proof. Since $x^{k+1} = \rho(x^k, t_k)$ is defined as the unique optimal solution determined by (2.4) and (3.2), it follows from the first-order optimality condition of (2.3) that we have

$$\frac{x^k - x^{k+1}}{t_k} \in \partial \psi_{x^k}(x^{k+1}). \tag{5.2}$$

Given $x \in \mathrm{dom}(F)$. By the definition of the subdifferential in (2.1), it follows that

$$\frac{1}{t_k}\langle x^k - x^{k+1}, x - x^{k+1}\rangle + \psi_{x^k}(x^{k+1}) \le \psi_{x^k}(x). \tag{5.3}$$

From the definition of $\psi_{x^k}(\cdot)$ in (2.2) and the convexity of G_j, we have

$$\begin{aligned}\psi_{x^k}(x) &= \max_{i=1,2,\ldots,m}\left\{H_i(x) - H_i(x^k) + \left\langle \nabla G_i(x^k), x - x^k\right\rangle\right\}\\ &\le \max_{i=1,2,\ldots,m}\left\{H_i(x) - H_i(x^k) + G_i(x) - G_i(x^k)\right\}\\ &= \max_{i=1,2,\ldots,m}\left\{F_i(x) - F_i(x^k)\right\}.\end{aligned} \tag{5.4}$$

In addition, by Lemma 4.3 we have, for all $k \ge \hat{k}_\ell$

$$\begin{aligned}F_j(x^{k+1}) - F_j(x^k) &\le \varphi_{x^k,t_k}(x^{k+1}) + \left(c_0 - \frac{1}{2}\right)\frac{\|x^{k+1} - x^k\|^2}{t_k}\\ &= \psi_{x^k}(x^{k+1}) + c_0 \cdot \frac{\|x^{k+1} - x^k\|^2}{t_k}, \quad \text{for all } j = 1,\ldots,m.\end{aligned}$$

Hence

$$F_j(x^{k+1}) - F_j(x^k) - \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 \le \psi_{x^k}(x^{k+1}). \tag{5.5}$$

From (5.3), (5.4) and (5.5), we obtain that for all $j = 1,\ldots,m$,

$$\frac{1}{t_k}\langle x^k - x^{k+1}, x - x^{k+1}\rangle + F_j(x^{k+1}) - F_j(x^k) - \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 \le \max_{i=1,\ldots,m}\{F_i(x) - F_i(x^k)\}$$

which is equivalent to

$$2\langle x^k - x^{k+1}, x - x^{k+1}\rangle + 2t_k\left(F_j(x^{k+1}) - F_j(x^k)\right) - 2c_0\|x^{k+1} - x^k\|^2 \le 2t_k \max_{i=1,\ldots,m}\{F_i(x) - F_i(x^k)\}. \tag{5.6}$$

Since

$$2\langle x^k - x^{k+1}, x - x^{k+1}\rangle = -\|x^k - x\|^2 + \|x^k - x^{k+1}\|^2 + \|x - x^{k+1}\|^2,$$

it follows from (5.6) that

$$\begin{aligned}\|x - x^{k+1}\|^2 + (1 - 2c_0)\|x^k - x^{k+1}\|^2 + 2t_k\left(F_j(x^{k+1}) - F_j(x^k)\right)\\ \le \|x^k - x\|^2 + 2t_k \max_{i=1,\ldots,m}\left\{F_i(x) - F_i(x^k)\right\}.\end{aligned}$$

Thus,

$$\begin{aligned}\|x - x^{k+1}\|^2 \le \|x^k - x\|^2 &+ 2t_k\left(F_j(x^k) - F_j(x^{k+1})\right)\\ &+ 2t_k \max_{i=1,\ldots,m}\left\{F_i(x) - F_i(x^k)\right\} - (1 - 2c_0)\|x^k - x^{k+1}\|^2.\end{aligned}$$

This completes the proof. □

To continue analyzing the convergence of Algorithm 3.1-AMPGℓ, we use an additional assumption below.

Assumption 4. *All monotonically non-increasing sequences in the set* $\mathrm{Im}(F)$ *are bounded from below by a point in* $\mathrm{Im}(F)$*, i.e., for every sequence* $\{y^k\} \subset \mathrm{dom}(F)$ *such that* $F(y^{k+1}) \leq F(y^k)$ *for all* $k \in \mathbb{N}$*, there exists* $\bar{y} \in \mathrm{dom}(F)$ *such that* $F(\bar{y}) \leq F(y^k)$ *for all* $k \in \mathbb{N}$*.*

Assumption 4 is widely used in the literature (see e.g., [11, 22, 23, 24, 25, 26, 27, 28]). It is commonly employed to ensure the existence of efficient solutions for vector-valued optimization problems, see [29, Section 2.3]. In the scalar-valued unconstrained case, it is equivalent to ensuring the existence of an optimal point.

Remark 5.3. Clearly, if (P) satisfies Assumption 3 then it satisfies Assumption 2 and if (P) satisfies Assumption 4 then the requirement of Theorem 4.4 regarding the existence of $j_0 \in \{1, \ldots, m\}$ such that F_{j_0} is lower bounded on $\mathbb{R}^n$, is also satisfied. Hence, we can inherit all the results of section 4 in the next theorem (Theorem 5.4) where the convergence of (P) is established under Assumption 1, 3 and 4.

Theorem 5.4. *Suppose that Problem* (P) *satisfies Assumption 1, 3 and 4. Then, we have:*

(i) *The sequence* $\{x^k\}$ *generated by Algorithm 3.1 - AMPGℓ converges to a weak Pareto optimal solution of Problem* (P)*;*

(ii) *For all* $j = 1, \ldots, m$*, and for all* $K \geq \hat{k}_\ell$*,*

$$\min_{i=1,\ldots,m} \left(F_i(x^K) - F_i(x^*)\right) \leq \frac{\|x^{\hat{k}_\ell} - x^*\|^2 + 2t^* \left(F_j(x^{\hat{k}_\ell}) - F_j(x^{K+1})\right)}{2t_{\min}\left(K - \hat{k}_\ell + 1\right)} = \mathcal{O}\left(\frac{1}{K}\right).$$

Proof. ***(i)*** Setting $T = \{x \in \mathrm{dom}(F) \mid F(x) \leq F(x^k),\ \forall k \geq \hat{k}_\ell\}$.

Following Lemma 4.3, $\{F(x^k)\}_{k \geq \hat{k}_\ell}$ is decreasing; then, by Assumption 4, $T \neq \emptyset$. Let $x^* \in T$ and substitute $x = x^*$ in (5.1). For any $j = 1, \ldots, m$, we have that

$$\begin{aligned} \|x^* - x^{k+1}\|^2 &\leq \|x^k - x^*\|^2 + 2t_k \left(F_j(x^k) - F_j(x^{k+1})\right) \\ &\quad + 2t_k \max_{i=1,\ldots,m} \underbrace{\left(F_i(x^*) - F_i(x^k)\right)}_{\leq 0} - \underbrace{(1 - 2c_0)}_{>0} \|x^k - x^{k+1}\|^2 \qquad (5.7) \\ &\leq \|x^k - x^*\|^2 + 2t_k \left(F_j(x^k) - F_j(x^{k+1})\right) \quad \text{for all } k \geq \hat{k}_\ell, \qquad (5.8) \end{aligned}$$

where the last inequality follows from $x^* \in T$ and $0 < c_0 < \frac{1}{2}$. Thus, if we set

$$\varepsilon_k = 2t_k \left(F_j(x^k) - F_j(x^{k+1})\right)$$

then from (5.8) we have

$$\|x^* - x^{k+1}\|^2 \leq \|x^k - x^*\|^2 + \epsilon_k, \quad \text{for all } k \geq \hat{k}_\ell. \qquad (5.9)$$

Note that $F(x^k) \geq F(x^{k+1})$ (by Lemma 4.3) and $t_k \leq t^*$ (by Remark 3.5) for all $k \geq \hat{k}_\ell$, then

$$\begin{aligned} \sum_{k=\hat{k}_\ell}^{K-1} \epsilon_k &\leq \sum_{k=\hat{k}_\ell}^{K-1} 2t^* \left(F_j(x^k) - F_j(x^{k+1})\right) = 2t^*(F_j(x^{\hat{k}_\ell}) - F_j(x^K)) \\ &\leq 2t^* \left(F_j(x^{\hat{k}_\ell}) - F_j(x^*)\right), \quad \text{for all } K - 1 \geq \hat{k}_\ell. \end{aligned}$$

Therefore,

$$\sum_{k=\hat{k}_\ell}^{+\infty} \epsilon_k \leq 2t^* \left(F_j(x^{\hat{k}_\ell}) - F_j(x^*)\right) < +\infty. \tag{5.10}$$

Inequalities (5.9) and (5.10) imply that the sequence $\{x^k\}_{k\geq\hat{k}_\ell}$ is quasi-Fejér convergent to T (by Definition 2.6). From Lemma 2.7, we obtain that $\{x^k\}_{k\geq\hat{k}_\ell}$ is bounded and hence it has cluster points. Suppose that $\bar{x}$ is a cluster point of $\{x^k\}_{k\geq\hat{k}_\ell}$ then there exists a subsequence $\{x^{k_\nu}\}$ converging to $\bar{x}$.

Note that, $F(x^*) \leq F(x^k)$ for all $k \geq \hat{k}_\ell$ and by Remark 5.3, we inherit some following results from section 4 including Lemma 4.3 and Theorem 4.4:

(a) For each $j \in \{1,\ldots,m\}$, the sequence $\{F_j(x^k)\}_{k\geq\hat{k}_\ell}$ converges to $\hat{F}_j$ satisfies $\hat{F}_j \leq F_j(x^k)$ for all $k \geq \hat{k}_\ell$. From the continuity of F_j we derive that

$$\hat{F}_j = \lim_{k\to\infty} F_j(x^k) = \lim_{k_\nu\to\infty} F_j(x^{k_\nu}) = F_j(\bar{x}).$$

Thus, $F_j(\bar{x}) \leq F(x^k)$ for all $k \geq \hat{k}_\ell$ and hence $\bar{x} \in T$. Following Lemma 2.7, the sequence $\{x^k\}_{k\geq\hat{k}_\ell}$ converges to $\bar{x}$.

(b) Theorem 4.4 shows that $\bar{x}$ is a stationary point of Problem (P). By Lemma 2.3 (ii), $\bar{x}$ is a weak Pareto solution of Problem (P).

This completes the proof of Part (i).

(ii) From inequality (5.7) and noting that for each $j \in \{1,...,m\}$, $F_j(x^k) - F_j(x^{k+1}) \geq 0$ for all $k \geq \hat{k}_\ell$ (by Lemma 4.3), $t_k \leq t^*$ (by Remark 3.5), then we have for any $j \in \{1,\ldots,m\}$,

$$\begin{aligned} &2t_k \min_{i=1,\ldots,m} \left(F_i(x^k) - F_i(x^*)\right) \\ &\leq \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 - (1-2c_0)\|x^k - x^{k+1}\|^2 + 2t_k \left(F_j(x^k) - F_j(x^{k+1})\right) \\ &\leq \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 + 2t^* \left(F_j(x^k) - F_j(x^{k+1})\right) \quad \text{for all } k \geq \hat{k}_\ell. \end{aligned}$$

Moreover, since $t_k \geq t_{\min}$ (by Remark 3.5) and $\min_{i=1,\ldots,m} \left(F_i(x^k) - F_i(x^*)\right) \geq 0$ for all $k \geq \hat{k}_\ell$ then

$$2t_{\min} \min_{i=1,\ldots,m} \left(F_i(x^k) - F_i(x^*)\right) \leq \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 + 2t^* \left(F_j(x^k) - F_j(x^{k+1})\right), \forall j = 1,...,m. \tag{5.11}$$

Summing up (5.11) from $k = \hat{k}_\ell$ to $K \geq \hat{k}_\ell$ yields

$$2t_{\min} \sum_{k=\hat{k}_\ell}^{K} \min_{i=1,\ldots,m} \left(F_i(x^k) - F_i(x^*)\right) \leq \|x^{\hat{k}_\ell} - x^*\|^2 - \|x^{K+1} - x^*\|^2 + 2t^* \left(F_j(x^{\hat{k}_\ell}) - F_j(x^{K+1})\right). \tag{5.12}$$

On the other hand, $F_j(x^k) \geq F_j(x^K)$ for all $\hat{k}_\ell \leq k \leq K$ then

$$\left(K - \hat{k}_\ell + 1\right) \min_{i=1,\ldots,m} \left(F_i(x^K) - F_i(x^*)\right) \leq \sum_{k=\hat{k}_\ell}^{K} \min_{i=1,\ldots,m} \left(F_i(x^k) - F_i(x^*)\right). \tag{5.13}$$

From (5.12), (5.13) and $\|x^{K+1} - x^*\|^2 \geq 0$, we get

$$2t_{\min}\left(K - \hat{k}_\ell + 1\right) \min_{i=1,\dots,m} \left(F_i(x^K) - F_i(x^*)\right) \leq \|x^{\hat{k}_\ell} - x^*\|^2 + 2t^*\left(F_j(x^{\hat{k}_\ell}) - F_j(x^{K+1})\right).$$

Dividing both sides by $2t_{\min}(K - \hat{k}_\ell + 1)$, for all $K \geq \hat{k}_\ell$ we obtain

$$\min_{i=1,\dots,m} \left(F_i(x^K) - F_i(x^*)\right) \leq \frac{\|x^{\hat{k}_\ell} - x^*\|^2 + 2t^*\left(F_j(x^{\hat{k}_\ell}) - F_j(x^{K+1})\right)}{2t_{\min}\left(K - \hat{k}_\ell + 1\right)} = \mathcal{O}\left(\frac{1}{K}\right).$$

□

6. Linear convergence rate for strongly convex case of Problem (P)

In this section, we establish the linear convergence rate of the sequence $\{x^k\}_{k\geq\hat{k}_\ell}$ to a Pareto optimal solution of Problem (P) when every component G_j is strongly convex. First, we prove the key inequality for the convergence analysis.

Lemma 6.1. *Let $x \in \mathrm{dom}(F)$ and let G_j be strongly convex with modulus $\mu_j > 0$ ($j = 1, 2, \dots, m$). Then, for any $k \geq \hat{k}_\ell$, there exist $\{\lambda_j^k\}_{j=1}^m \subseteq [0, 1]$ satisfying $\sum\limits_{j=1}^m \lambda_j^k = 1$ such that*

$$\sum_{j=1}^m \lambda_j^k \left(F_j(x^{k+1}) - F_j(x)\right) \leq \frac{1}{2t_k}\left(\|x^k - x\|^2 - \|x^{k+1} - x\|^2\right) - \frac{\mu}{2}\|x^k - x\|^2,$$

where $\mu := \min\limits_{j=1,2,\dots,m} \mu_j$.

Proof. From (4.2) and (4.4), for all $k \geq \hat{k}_\ell$, we have

$$G_j(x^{k+1}) \leq G_j(x^k) + \left\langle \nabla G_j(x^k), x^{k+1} - x^k \right\rangle + \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2, \quad \text{for all } j \in \{1, 2, \dots, m\}. \tag{6.1}$$

Since G_j is strongly convex with modulus $\mu_j > 0$, fixing any $x \in \mathrm{dom}(F)$, we have

$$\begin{aligned} G_j(x^k) &\leq G_j(x) - \frac{\mu_j}{2}\|x^k - x\|^2 + \left\langle \nabla G_j(x^k), x^k - x \right\rangle \\ &\leq G_j(x) - \frac{\mu}{2}\|x^k - x\|^2 + \left\langle \nabla G_j(x^k), x^k - x \right\rangle, \end{aligned} \tag{6.2}$$

where $\mu := \min\limits_{j=1,2,\dots,m} \mu_j > 0$.
Combining (6.1) with (6.2), for all $j \in \{1, 2, \dots, m\}$,

$$G_j(x^{k+1}) \leq G_j(x) + \left\langle \nabla G_j(x^k), x^{k+1} - x \right\rangle + \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 - \frac{\mu}{2}\|x^k - x\|^2. \tag{6.3}$$

On the other hand, by (5.2), we have

$$\frac{x^k - x^{k+1}}{t_k} \in \partial\psi_{x^k}(x^{k+1}).$$

Since $\psi_{x^k}(u) = \max_{j=1,\ldots,m} \{\langle \nabla G_j(x^k), u - x\rangle + H_j(u) - H_j(x^k)\}$, apply Theorem 3.50 in [30] for finding the subdifferential of $\psi_{x^k}(\cdot)$ at x^{k+1} (the maximum of convex functions) then there exists $\lambda_j^k \geq 0$, $\sum_{j=1}^m \lambda_j^k = 1$, $p_j^k \in \partial H_j(x^{k+1})$ such that

$$\frac{x^k - x^{k+1}}{t_k} = \sum_{j=1}^m \lambda_j^k \left(\nabla G_j(x^k) + p_j^k\right). \tag{6.4}$$

Since $p_j^k \in \partial H_j(x^{k+1})$ then

$$H_j(x^{k+1}) \leq \langle p_j^k, x^{k+1} - x\rangle + H_j(x).$$

This together with (6.3) yields that for each $j \in \{1, 2, \ldots, m\}$ and $k \geq \hat{k}_\ell$,

$$F_j(x^{k+1}) - F_j(x) \leq \left\langle \nabla G_j(x^k) + p_j^k, x^{k+1} - x\right\rangle + \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 - \frac{\mu}{2}\|x^k - x\|^2,$$

which yields that

$$\begin{aligned}
&\sum_{j=1}^m \lambda_j^k \left(F_j(x^{k+1}) - F_j(x)\right) \\
&\qquad \leq \sum_{j=1}^m \lambda_j^k \left\langle \nabla G_j(x^k) + p_j^k, x^{k+1} - x\right\rangle + \sum_{j=1}^m \lambda_j^k \left(\frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 - \frac{\mu}{2}\|x^k - x\|^2\right).
\end{aligned}$$

Noting (6.4), $\sum_{j=1}^m \lambda_j^k = 1$, and $c_0 \in (0, \frac{1}{2})$, we get that

$$\begin{aligned}
\sum_{j=1}^m \lambda_j^k \left(F_j(x^{k+1}) - F_j(x)\right) &\leq \frac{1}{t_k}\langle x^k - x^{k+1}, x^{k+1} - x\rangle + \frac{c_0}{t_k}\|x^{k+1} - x^k\|^2 - \frac{\mu}{2}\|x^k - x\|^2 \\
&\leq \frac{1}{2t_k}\left(2\langle x^k - x^{k+1}, x^{k+1} - x\rangle + \|x^{k+1} - x^k\|^2\right) - \frac{\mu}{2}\|x^k - x\|^2 \\
&= \frac{1}{2t_k}\left(\|x^k - x\|^2 - \|x^{k+1} - x\|^2\right) - \frac{\mu}{2}\|x^k - x\|^2.
\end{aligned}$$

Thus, we obtain the desired conclusion. □

The linear convergence rate of $\{x^k\}$ generated by Algorithm 3.1-AMPGℓ for the case where G_j ($j = 1, \ldots, m$) are strongly convex is presented in the following theorem.

Theorem 6.2. *Suppose that Problem* (P) *satisfies Assumption 1, 3, and each $G_j (j = 1, 2, \ldots, m)$ is strongly convex with modulus $\mu_j > 0$. Then, $\{x^k\}$ converges linearly to a Pareto optimal point x^* of* (P)*. Specifically, we have*

$$\|x^k - x^*\| \leq \left(\sqrt{1 - t_{\min}\mu}\right)^k \|x^0 - x^*\| = \mathcal{O}(r^k) \quad \textit{for all } k \geq \hat{k}_\ell,$$

where $\mu := \min_{j=1,\ldots,m} \mu_j$ and $r := \sqrt{1 - t_{\min}\mu} \in (0, 1)$.

Proof. From Theorem 5.4, $\{x^k\}$ converges to a weakly Pareto optimal solution of Problem (P) that is $x^* \in T$. Since F_j is strongly convex for all $j = 1, ..., m$, by Lemma 2.3, x^* is a Pareto optimal solution of Problem (P). Using Lemma 6.1 (iii) with $x = x^*$ we have

$$0 \leq \sum_{j=1}^m \lambda_j^k \left(F_j(x^{k+1}) - F_j(x^*)\right) \leq \frac{1}{2t_k}\left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2\right) - \frac{\mu}{2}\|x^k - x^*\|^2,$$

which implies that

$$\|x^{k+1} - x^*\|^2 \leq (1 - t_k\mu)\|x^k - x^*\|^2 \underset{t_k \geq t_{\min}}{\leq} (1 - t_{\min}\mu)\|x^k - x^*\|^2, \quad \text{for all } k \geq \hat{k}_\ell.$$

Taking the square root on both sides and applying the inequality recursively, we obtain

$$\begin{aligned}\|x^{k+1} - x^*\| &\leq \sqrt{1 - t_{\min}\mu}\,\|x^k - x^*\| \leq \left(\sqrt{1 - t_{\min}\mu}\right)^2 \|x^{k-1} - x^*\| \\ &\leq \cdots \leq \left(\sqrt{1 - t_{\min}\mu}\right)^{k+1} \|x^0 - x^*\| \\ &= \mathcal{O}(r^{k+1}), \quad \text{where } r = \sqrt{1 - t_{\min}\mu} \in (0,1), \quad \text{for all } k \geq \hat{k}_\ell.\end{aligned}$$

This completes the proof. □

7. Numerical results

In this section, the performance of the proposed AMPGℓ ($\ell = 1, 2$) algorithms will be evaluated and compared against three state-of-the-art proximal gradient algorithms including: **MPGE** [11], **PGM**[12], and **accPGM** [12].

Parameters setting: For AMPGℓ ($\ell = 1, 2$), we set the parameters $\eta_0 = 0.2, \eta_1 = 0.19$. The convergent positive series $\sum_{k=0}^{+\infty} \varepsilon_k$ is created by $\varepsilon_k = \dfrac{1.5(\ln(k+1))^6}{(k+1)^{1.1}}$, $k \geq 0$. The parameter configurations for the MPGE, PGM, and accPGM are retained exactly as described in their respective original papers. Specifically:

- MPGE: The parameters used are $t = 1, \gamma = 1.9999, \tau_1 = 0.1, \tau_2 = 0.9$.
- PGM and accPGM: The configured parameters include $\delta = 0.3, \eta = 0.5, \sigma = 1$.

For all algorithms, we set the tolerance $\varepsilon = 10^{-6}$ and the maximum number of iterations to 1000. During the implementation, the algorithms are terminated upon satisfying either of the two criteria below:

$$\|x^{k+1} - x^k\| \leq \varepsilon, \quad \text{or} \quad \frac{\|x^{k+1} - x^k\|_\infty}{\max\{1, \|x^k\|_\infty\}} \leq \varepsilon,$$

where the second stopping criterion is redesigned following the approach in [11].

For each test problem, we generated 100 uniformly distributed starting points x^0 within the corresponding domain $\mathrm{dom}(F)$. All numerical experiments were conducted using Python 3.11.5 on a personal laptop Lenovo Thinkpad running Windows 11, equipped with Processor 13th Gen Intel(R) Core(TM) i7-1360P (2.20 GHz); Installed RAM 16.0 GB; System type 64-bit operating system, x64-based processor. Our codes are avaiable at the link https://github.com/hoaiphamthi/AMPG.

Performance metrics: The computational efficiency of the algorithms is evaluated based on the speed of computation and the quality of the obtained Pareto fronts. In particular, we use *execution time*, *the number of iterations* required to solve the problem from each starting point, and *average step-size* to evaluate the processing rate of algorithms. Meanwhile, the solution quality is assessed using widely adopted multiobjective metrics: *Global_nondominated*, *Purity*, *Spread* Γ, and *Hypervolume*.

Let $Algs$ and $Probs$ be the sets of algorithms and test problems used in experiments. For each test problem p in problem set $Probs$, let $F_{a,p}$ denote the set of nondominated points obtained by algorithm a, where $a \in Algs$. Let F_p be the set of nondominated points of $\left(\bigcup_{a \in Algs} F_{a,p}\right)$.

- *Global_nondominated* measures the number of solutions obtained by algorithm a that belong to the global nondominated set F_p, defined as $|F_{a,p} \cap F_p|$.

- *The purity* defined by $\dfrac{|F_{a,p} \cap F_p|}{|F_{a,p}|}$, measures the proportion of nondominated points obtained by algorithm a that also belong to F_p, see e.g., [31, 8] for details.

- *Hypervolume* measures the coverage of the set of obtained nondominated point. Each nondominated point $y \in F_{a,p}$, together with the reference point R_p, forms a hyper-rectangle $[y, R_p]$. Hypervolume of algorithm a for solving problem p is the volume of the union of all these hyper-rectangles $\bigcup_{y \in F_{a,p}} [y, R_p]$. We choose the reference point R_p for each problem p by $(R_p)_j = \max \left\{ y_j \mid y \in \bigcup_{a \in Algs} F_{a,p} \right\} \times 1.1, \quad j = 1, \ldots, m$. One can see [32, 8] for more details.

- *Spread* Γ evaluates the distribution and uniformity of points along the Pareto front. Suppose that $F_{a,p} \cap F_p$ contain N points, indexed by $1, \ldots, N$. We add two points Q_p^{min} and Q_p^{max}, indexed by 0 and $N+1$, respectively. Here, $Q^{p,min} = (Q_j^{p,min})_{j=1,\ldots,m}$ and $Q^{p,max} = (Q_j^{p,max})_{j=1,\ldots,m}$, where $Q_j^{p,min} = \min\{y_j \mid y \in F_p\}$ and $Q_j^{p,max} = \max\{y_j \mid y \in F_p\}$. For $i = 1, ..., N$, let $z_i = (z_{i,1}, \ldots, z_{i,m}) \in F_{a,p} \cap F_p$, where $z_{i,j}$ is the value of the j-th objective function; and let $z_0 = Q^{p,min}$, $z_{N+1} = Q^{p,max}$. For each objective j, the points are sorted in ascending order according to the values of the j-th objective. Then, for $N \geq 2$, the Spread Γ metric $\Gamma_{a,p}$ for $p \in Probs$ and $a \in Algs$ is defined by $\Gamma_{a,p} = \max\limits_{j \in \{1,\ldots,m\}} \max\limits_{i \in \{0,\ldots,N\}} \delta_{i,j}$, where $\delta_{i,j} = z_{i+1,j} - z_{i,j}$ for $i = 0, ..., N$ and $j = 1, ..., m$. If $N < 2$, we set $\Gamma_{a,p} = \infty$. One can see [33, 8] for more details.

Obviously, higher values of Global_nondominated, Purity, and Hypervolume, alongside lower values of Spread Γ, indicate better algorithmic performance. To synthesize and compare the performance of the algorithms across all test problems, we use the **performance profiles** proposed by Dolan and Moré [34]. Let $t_{a,p}$ be the value of a metric obtained by evaluating algorithm a on test problem p, assuming that a smaller value is better. For each problem set $Probs$, the performance profile of algorithm a is defined as

$$\rho_a(\tau) = \frac{1}{|Probs|} \left| \{ p \in Probs : r_{a,p} \leq \tau \} \right|, \qquad \text{where } \tau \geq 1 \text{ and } r_{a,p} = \frac{t_{a,p}}{\min\{t_{a',p} \mid a' \in Algs\}}.$$

The value $\rho_a(\tau)$ represents the proportion of problems for which the performance of algorithm a is at most τ times that of the best algorithm on the same problem. Therefore, $\rho_a(1)$ represents the proportion of problems for which algorithm a achieves the best performance. An algorithm with a higher performance profile curve is considered to have better overall performance.

Tested instances: We employ two sets of test problems to conduct our experiments, detailed as follows:

- Set 1: The robust multiobjective optimization problem considered in [11, 2].

- Set 2: The supervised feature selection problem, as presented in [3].

We describe these problems and the corresponding numerical results in the following subsections.

7.1. **The robust multiobjective optimization problem**

The robust multiobjective optimization considered in [2, 11] has the form of Problem (P) with

$$\min_{x\in\mathbb{R}^n} (F_1(x), ..., F_m(x)) \qquad \text{(rMOP)}$$
$$\text{with } F_j(x) = G_j(x) + H_j(x), \quad j = 1, \dots, m,$$

where $\mathrm{dom}(F_j) = \{x \in \mathbb{R}^n | \ x_L \leq x \leq x_U\}$ for all $j \in \{1, ..., m\}$; G_j is selected from works in the literature as. Detailed information and formulations of G_j can be found in Table 1 and Appendix A, respectively. For each $j \in \{1, ..., m\}$, the convex function $H_j : \mathrm{dom}(F_j) \to \mathbb{R}$, acting as the robust function, defined as follows:

$$H_j(u) = \max_{z\in Z_j} u^\top z, \tag{7.1}$$

where $Z_j = \{z \in \mathbb{R}^n | \ A_j z \leq b_j\}$ ($A_j \in \mathbb{R}^{d\times n}, b_j \in \mathbb{R}^n$) is a nonempty, bounded convex polyhedron standing for the uncertainty set. For the experimental configuration, Z_j is generated by the similar method in [11]. Specifically, $B_j \in \mathbb{R}^{n\times n}$ is a randomly generated nonsingular matrix. We set $\lambda = \hat{\lambda}\|\hat{x}\|$, where $\hat{\lambda}$ is randomly chosen from the interval $[0.02, 0.10]$, and $\hat{x}$ is an arbitrary point in $\mathrm{dom}(F)$. Then $Z_j := \{z \in \mathbb{R}^n | -\lambda e \leq B_j z \leq \lambda e\}, \quad j = 1, \dots, m$, where $e = (1, \dots, 1)^\top \in \mathbb{R}^n$. Thus, we have $d = 2n$, i.e., $A_j \in \mathbb{R}^{2n\times n}$. Clearly, computing the value of $H_j(\cdot)$ is equivalent to solving a linear programming problem with high cost. To implement Algorithm 3.1-AMPGℓ, we need to solve the subproblem (2.3) which is now reformulated as follows:

$$\begin{aligned} &\min_{u\in\mathbb{R}^n} \left\{ \max_{j=1,\dots,m} \left(\nabla G_j(x)^\top (u - x) + H_j(u) - H_j(x) \right) + \frac{1}{2t}\|u - x\|^2 \right\} \\ &\text{s.t. } \ x_L \leq u \leq x_U. \end{aligned}$$

By introducing an auxiliary variable $\tau \in \mathbb{R}$ into the model, we obtain the equivalent problem:

$$\begin{aligned} \min_{\tau,u} \quad & \tau + \frac{1}{2t}\|u - x\|^2 \\ \text{s.t.} \quad & \nabla G_j(x)^\top (u - x) + H_j(u) - H_j(x) \leq \tau, \quad j = 1, \dots, m, \\ & x_L \leq u \leq x_U. \end{aligned} \tag{7.2}$$

Note that for each $j = 1, \dots, m$, the dual problem of (7.1) takes the form:

$$\begin{aligned} \min_{w_j} \quad & b_j^\top w_j \\ \text{s.t.} \quad & A_j^\top w_j = u, \quad w_j \geq 0. \end{aligned} \tag{7.3}$$

Using the strong duality theorem alongside the dual problem of H_j in (7.3), we can reformulate subproblem (7.2) as a quadratic programming:

$$\begin{aligned} \min_{\tau,u,w_j} \quad & \tau + \frac{1}{2t}\|u - x\|^2 \\ \text{s.t.} \quad & \nabla G_j(x)^\top (u - x) + b_j^\top w_j - H_j(x) \leq \tau, \quad j = 1, \dots, m, \\ & A_j^\top w_j = u, \quad \forall j = 1, \dots, m, \\ & w_j \geq 0, \quad j = 1, \dots, m, \\ & x_L \leq u \leq x_U. \end{aligned} \tag{7.4}$$

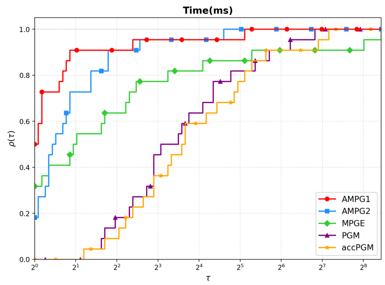
Regarding computational implementation, we employ the `linprog` solver to compute the value of the function H_j at $x \in \mathrm{dom}(F)$, and the `quadprog` solver to solve subproblem (7.4).

Table 1: The reference corresponding to the formulation of G_j used for experiments, number of variables (n) and objectives (m), and variable bounds.

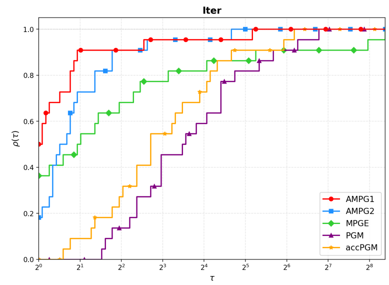
Problem	Refs.	n	m	x_L	x_U
AP1	[35]	2	3	$(-10, -10)$	$(10, 10)$
AP4	[35]	3	3	$(-10, -10, -10)$	$(10, 10, 10)$
DTLZ1	[36]	7	3	$(0, \dots, 0)$	$(1, \dots, 1)$
DTLZ2	[36]	7	3	$(0, \dots, 0)$	$(1, \dots, 1)$
DTLZ3	[36]	7	3	$(0, \dots, 0)$	$(1, \dots, 1)$
DTLZ4	[36]	7	3	$(0, \dots, 0)$	$(1, \dots, 1)$
FA1	[37]	3	3	$(0.01, 0.01, 0.01)$	$(1, 1, 1)$
Far1	[37]	2	2	$(-1, -1)$	$(1, 1)$
FDS	[38]	10	3	$(-2, \dots, -2)$	$(2, \dots, 2)$
IKK1	[39]	2	3	$(-50, -50)$	$(50, 50)$
LE1	[40]	2	2	$(1, 1)$	$(10, 10)$
LTDZ	[41]	3	3	$(0, 0, 0)$	$(1, 1, 1)$
MGH33	[42]	3	3	$(-1, -1, -1)$	$(1, 1, 1)$
MLF1	[43]	1	2	0	20
MMR4	[44]	3	2	$(0, 0, 0)$	$(4, 4, 4)$
MOP2	[45]	2	2	$(-1, -1)$	$(1, 1)$
MOP3	[45]	2	2	$(-\pi, -\pi)$	(π, π)
MOP5	[45]	2	3	$(-1, -1)$	$(1, 1)$
SK1	[46]	1	2	-100	100
TKLY1	[47]	4	2	$(0.1, 0, 0, 0)$	$(1, 1, 1, 1)$
Toi9	[48]	4	4	$(-1, -1, -1, -1)$	$(1, 1, 1, 1)$
ZDT6	[49]	10	2	$(0.01, \dots, 0.01)$	$(1, \dots, 1)$

Obtained numerical results for Problem (rMOP): Figure 1 illustrates the overall computational performance of the evaluated algorithms on the selected test problems. Since evaluating the function H_j at each iterate x^k requires solving a linear programming problem which incurs a high computational cost, we specifically track *the number of H_j evaluations (H-Ev)* for each starting point x^0. As observed in the results, regarding execution time, the AMPG1 and AMPG2 algorithms demonstrate the most superior performance among the investigated methods, followed by MPGE, PGM, and accPGM. This observation is highly consistent with the other evaluation metrics, including the number of iterations, stepsize, and H-Ev. Specifically, AMPG1 and AMPG2 require the fewest iterations while maintaining a larger stepsize compared to the alternative approaches. Furthermore, these two algorithms necessitate the lowest number of H_j evaluations, which significantly mitigates the computational overhead and shortens the overall execution time.

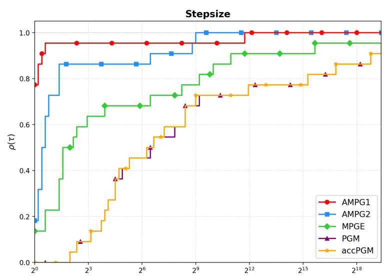
Figure 2 illustrates the evaluation results regarding the quality of the Pareto solutions generated by the considered algorithms. It shows the highest performances of AMPG1 and AMPG2 in terms of Global_nondominated and Purity metrics. For Spread Γ and Hypervolume metrics, our algorithms are competitive with the remaining ones. Regarding AMPG1 and AMPG2: AMPG2 demonstrates better computational speed with a faster execution time and fewer calls for evaluating the H_j function. Meanwhile, AMPG1 exhibits better solution quality, clearly outperforming in terms of the Global_nondominated, Purity, and Spread Γ metrics.



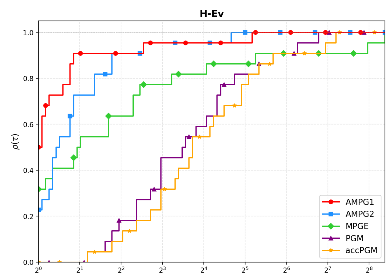
(a) Time (ms)



(b) Number of iterations

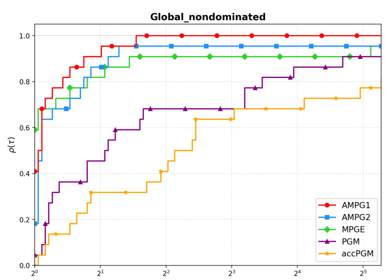


(c) Average stepsize

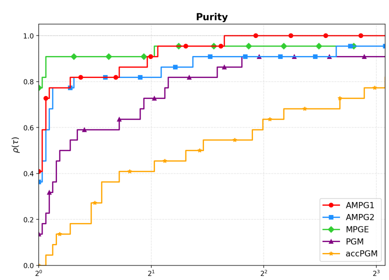


(d) H-Ev

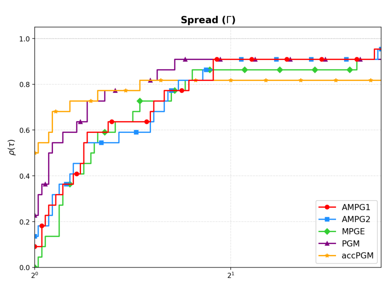
Figure 1: Performance profiles of the compared algorithms for the robust multiobjective optimization (rMOP)



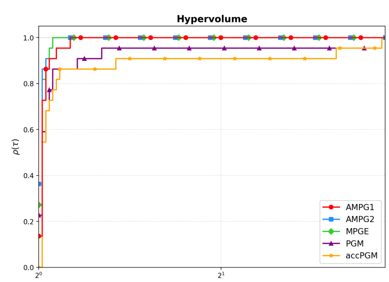
(a) Global_nondominated



(b) Purity



(c) Spread Γ



(d) Hypervolume

Figure 2: Performance profiles of the compared algorithms for the robust multiobjective optimization (rMOP)

Figure 3 visually illustrates the initial starting points and the final obtained solutions of each run for the evaluated algorithms, alongside the approximated Pareto fronts for several representative benchmark problems.

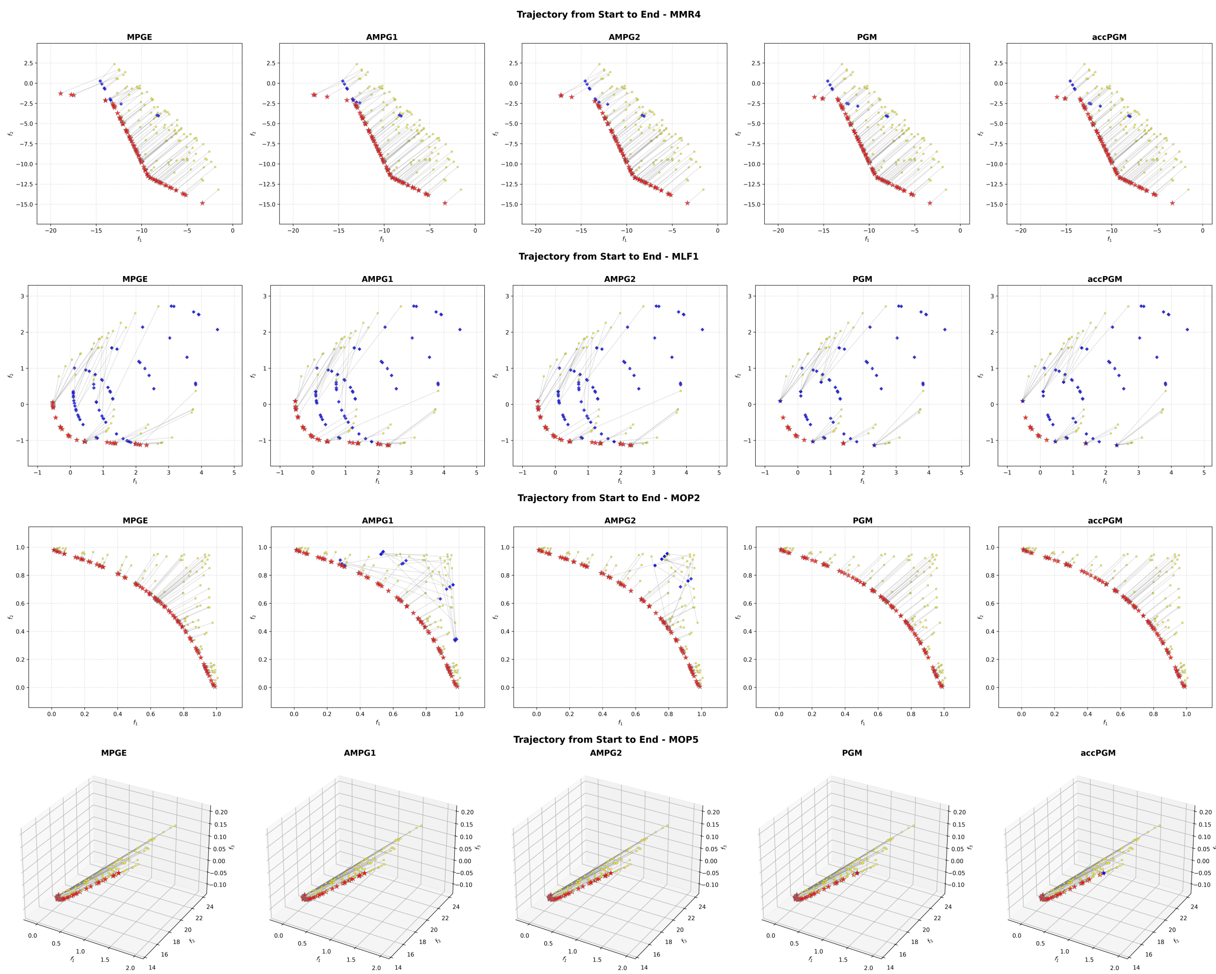


Figure 3: Visualization of the initial starting points and the final solutions obtained by all algorithms for Problem (rMOP)

7.2. **Biobjective supervised feature selection (B-SFS) problems**

In the study by Wang et al. [50], the authors address the supervised feature selection (SFS) problem within the context of multi-class classification problem. Given an input dataset comprising N samples, n features, and K classes, the objective is to select an optimal subset of $\tilde{n}$ features ($\tilde{n} < n$) that best represents the original feature space. These selected features must simultaneously satisfy two criteria: maximizing the relevance to the target labels and minimizing the redundancy among themselves. Accordingly, this problem is formulated as a fractional programming:

$$\begin{aligned} &\min_x \frac{x^\top Q x}{\rho^\top x} \\ &\text{s.t.} \quad x \in \mathcal{C} = \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^{p} x_i = 1, x \geq 0 \right\}. \end{aligned} \tag{7.5}$$

Here, $Q \in \mathbb{R}^{n \times n}$ is a positive semi-definite matrix representing the redundancy, and $\rho \in \mathbb{R}^n$ is a vector representing the relevance levels of all n features with respect to the class labels. Both are computed from the input data, while $x = (x_1, x_2, \ldots, x_n)^\top$ serves as the feature score vector.

This approach is further improved in the research by Khanh and Hoai [3], where problem (7.5) is reformulated as a convex bi-objective optimization problem aimed at simultaneously optimizing two objective functions:

$$\min_x \begin{cases} G_1(x) = x^\top Qx \\ G_2(x) = -\rho^\top x \end{cases}, \quad \text{s.t.} \quad x \in \mathcal{C} = \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 1, x \geq 0 \right\}. \tag{7.6}$$

Model (7.6) can then be rewritten as the following unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^n} \begin{cases} F_1(x) = G_1(x) + H_1(x) \\ F_2(x) = G_2(x) + H_2(x) \end{cases}, \text{where } H_1(x) = H_2(x) = i_{\mathcal{C}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{C} \\ +\infty, & \text{otherwise} \end{cases}. \tag{B-SFS}$$

Consequently, the subproblem of (B-SFS) corresponding to the problem (2.3) is

$$\min_{u\in\mathbb{R}^n} \left\{ \max_{j=1,2} \left(\nabla G_j(x)^\top (u-x) + i_C(u) - i_C(x) \right) + \frac{1}{2t}\|u-x\|^2 \right\}. \tag{7.7}$$

By introducing an auxiliary variable $\tau = \max_{j=1,2} \nabla G_j(x)^T(u-x)$, the problem (7.7) can be rewritten as a quadratic programming problem

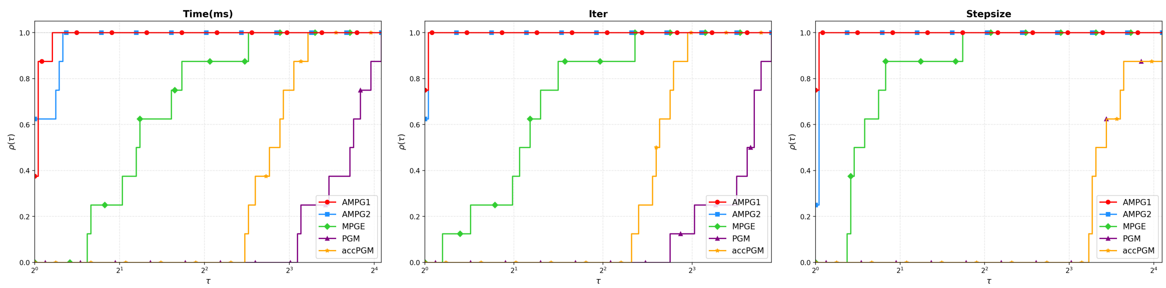
$$\begin{aligned} \min_{\tau,u} \quad & \tau + \frac{1}{2t}\|u-x\|^2 \\ \text{s.t.} \quad & \nabla G_j(x)^\top (u-x) \leq \tau, \quad j = 1,2, \\ & \sum_{i=1}^n u_i = 1, \ u \geq 0. \end{aligned} \tag{7.8}$$

In our experiments, subproblem (7.8) is solved using the `quadprog` solver. The details of the utilized datasets are summarized in Table 2.

Table 2: Summary of the benchmark datasets used for Problem (B-SFS)

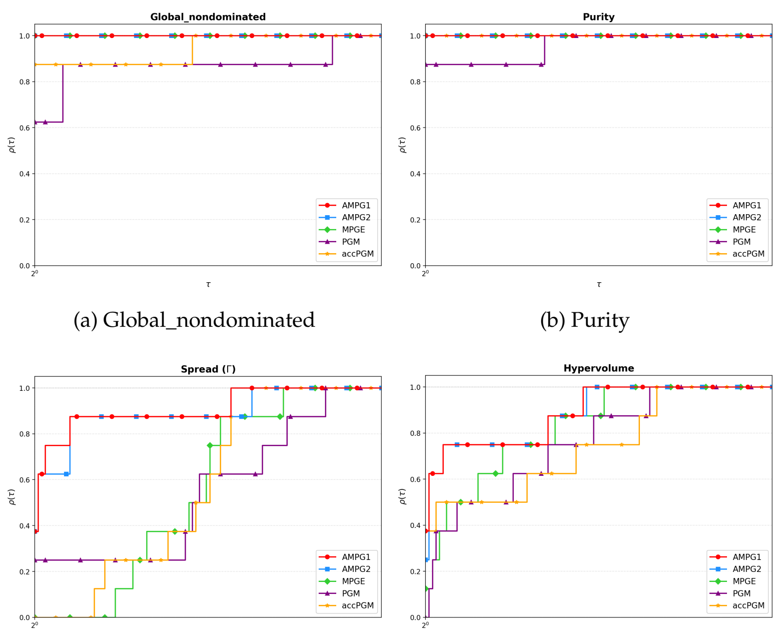
Dataset	Samples (N)	Features (n)	Classes (K)
CMC `https://archive.ics.uci.edu/dataset/30/contraceptive+method+choice`	1473	9	3
Heart `https://archive.ics.uci.edu/dataset/45/heart+disease`	270	12	2
Parkinsons `https://archive.ics.uci.edu/dataset/174/parkinsons`	195	22	2
Ionosphere `https://archive.ics.uci.edu/dataset/52/ionosphere`	351	34	2
Gallstone `https://archive.ics.uci.edu/dataset/1150/gallstone-1`	319	38	2
Musk-v1 `https://archive.ics.uci.edu/dataset/74/musk+version+1`	476	167	2
Arrhythmia `https://archive.ics.uci.edu/dataset/5/arrhythmia`	452	278	2
Darwin `https://archive.ics.uci.edu/dataset/732/darwin`	174	450	2

Obtained numerical results for Problem (B-SFS): Figure 4 and Figure 5 respectively present the performance profiles regarding the computational efficiency and the solution quality of the evaluated algorithms. The obtained results are entirely consistent with those of the first experiment. Specifically, AMPG1 and AMPG2 continue to demonstrate overwhelming superiority, leading in all three computational efficiency metrics (Time (ms), Number of iterations, and Average stepsize), while notably leading in all four solution quality evaluation metrics (Global_nondominated, Purity, Spread Γ, Hypervolume).



(a) Time (ms) (b) Number of iterations (c) Average stepsize

Figure 4: Performance profiles of the compared algorithms for Problem (B-SFS)



(a) Global_nondominated (b) Purity

(c) Spread Γ (d) Hypervolume

Figure 5: Performance profiles of the compared algorithms for Problem (B-SFS)

8. Conclusions

In this paper, we propose new proximal gradient algorithms equipped with two adaptive strategies for selecting the stepsize to solve multiobjective composite optimization problems. The resulting algorithms are evaluated comprehensively from both theoretical and computational perspectives. Under the nonconvex, convex, and strongly convex conditions of the differentiable terms, we established the corresponding theoretical convergence rates of $\mathcal{O}(1/\sqrt{K})$, $\mathcal{O}(1/K)$ and Q-linear, starting from a fixed iteration. Numerical results for various practical multiobjective composite optimization problems demonstrate that when multiobjective proximal gradient method combines with our new stepsize selection strategies, the obtained algorithms achieve better performance compared with the state-of-the-art.

References

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

[2] H. Tanabe, E. H. Fukuda, N. Yamashita, Proximal gradient methods for multiobjective optimization and their applications, Computational Optimization and Applications 72 (2) (2019) 339–361. doi:10.1007/s10589-018-0043-x.

[3] P. T. Khanh, P. T. Hoai, Supervised feature selection via multiobjective programming and its application in the medical field, Optimization Eruditorum 3 (3) (2026). doi:10.69829/oper-026-0303-ta02.

[4] J. Mairal, Incremental majorization-minimization optimization with application to large-scale machine learning, SIAM J. Optim. 25 (2) (2015) 829–855. doi:10.1137/140957639.

[5] M. El Moudden, A. El Mouatasim, Accelerated diagonal steepest descent method for unconstrained multiobjective optimization, Journal of Optimization Theory and Applications 188 (1) (2021) 220–242. doi:10.1007/s10957-020-01785-9.

[6] E. H. Fukuda, L. M. Graña Drummond, A survey on multiobjective descent methods, Pesquisa Operacional 34 (3) (2014) 585–620. doi:10.1590/0101-7438.2014.034.03.0585.

[7] H. Tanabe, E. H. Fukuda, N. Yamashita, Convergence rates analysis of a multiobjective proximal gradient method, Optimization Letters (2022). doi:10.1007/s11590-022-01877-7.

[8] H. Tanabe, E. H. Fukuda, N. Yamashita, An accelerated proximal gradient method for multiobjective optimization, Computational Optimization and Applications 86 (2) (2023) 421–455. doi:10.1007/s10589-023-00497-w.

[9] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM journal on imaging sciences 2 (2009) 183–202. doi:10.1137/080716542.

[10] J. Y. Bello-Cruz, J. G. Melo, R. V. G. Serra, A proximal gradient splitting method for solving convex vector optimization problems, Optimization 71 (1) (2022) 33–53. doi:10.1080/02331934.2020.1800699.

[11] Y. Bello-Cruz, J. G. Melo, L. F. Prudente, R. V. G. Serra, A proximal gradient method with an explicit line search for multiobjective optimization, Computational Optimization and Applications 92 (2025) 437—469. doi:10.1007/s10589-025-00711-x.

[12] X. Zhao, R. Raushan, D. Ghosh, J.-C. Yao, M. Qi, Proximal gradient method for convex multiobjective optimization problems without Lipschitz continuous gradients, Computational Optimization and Applications 91 (1) (2025) 27–66. doi:10.1007/s10589-025-00663-2.

[13] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, Soviet Mathematics Doklady 27 (1983) 372–376.

[14] J. Chen, L. Tang, X. Yang, Barzilai-Borwein proximal gradient methods for multiobjective composite optimization problems with improved linear convergence, arXiv preprint arXiv:2306.09797 (2023).

[15] P. T. Hoai, H. T. H. Yen, Steepest descent method using novel adaptive stepsizes for unconstrained nonlinear multiobjective programming, Optimization Online (2025). URL: `https://optimization-online.org/?p=30359`, published: 2025/05/11, Updated: 2025/10/03.

[16] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, 1970.

[17] J. Fliege, B. F. Svaiter, Steepest descent methods for multicriteria optimization, Mathematical Methods of Operations Research 51 (4) (2000) 479–494. doi:`10.1007/s001860000077`.

[18] J. Fliege, A. I. F. Vaz, L. N. Vicente, Complexity of gradient descent for multiobjective optimization, Optimization Methods and Software 34 (5) (2019) 949–959. doi:`10.1080/10556788.2018.1510928`.

[19] R. Burachik, L. M. G. Drummond, A. N. Iusem, B. F. Svaiter, Full convergence of the steepest descent method with inexact line searches, Optimization 59 (1) (2010) 137–146. doi:`10.1080/02331939508844042`.

[20] P. T. Hoai, N. P. D. Thai, Composite optimization problems via novel proximal gradient algorithms and applications, Optimization Online (2024). URL: `https://optimization-online.org/?p=26741`.

[21] P. Hoai, N. Vinh, N. Chung, A novel stepsize for gradient descent method, Operations Research Letters 52 (1) (2024) 107072. doi:`10.1016/j.orl.2024.107072`.

[22] J. Y. Bello Cruz, L. R. Lucambio Pérez, J. G. Melo, Convergence of the projected gradient method for quasiconvex multiobjective optimization, Nonlinear Analysis: Theory, Methods & Applications 74 (16) (2011) 5268–5273. doi:`10.1016/j.na.2011.04.067`.

[23] G. C. Bento, J. X. C. Neto, A. Soubeyran, A proximal point-type method for multicriteria optimization, Set-Valued and Variational Analysis 22 (3) (2014) 557–573. doi:`10.1007/s11228-014-0279-2`.

[24] H. Bonnel, A. N. Iusem, B. F. Svaiter, Proximal methods in vector optimization, SIAM Journal of Optimization 15 (4) (2005) 953–970. doi:`10.1137/S1052623403429093`.

[25] R. I. Boţ, S. M. Grad, Inertial forward-backward methods for solving vector optimization problems, Optimization 67 (7) (2018) 959–974. doi:`10.1080/02331934.2018.1440553`.

[26] T. D. Chuong, J. C. Yao, Steepest descent methods for critical points in vector optimization problems, Applicable Analysis 91 (10) (2012) 1811–1829. doi:`10.1080/00036811.2011.640629`.

[27] T. D. Chuong, Generalized proximal method for efficient solutions in vector optimization, Numerical Functional Analysis and Optimizition 32 (8) (2011) 843–857. doi:`10.1080/01630563.2011.587072`.

[28] E. H. Fukuda, L. M. G. Drummond, Inexact projected gradient method for vector optimization, Computational Optimization and Applications 54 (2013) 473–493. doi:`10.1007/s10589-012-9501-z`.

[29] D. T. Luc, Theory of Vector Optimization, volume 319 of *Lecture Notes in Economics and Mathematical Systems*, Springer, Berlin, 1989.

[30] A. Beck, First-order methods in optimization, SIAM, 2017.

[31] V. Morovati, L. Pourkarimi, H. Basirzadeh, Barzilai and borwein's method for multiobjective optimization problems, Numerical Algorithms 72 (2016) 539–604. doi:10.1007/s11075-015-0058-7.

[32] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms — a comparative case study, in: Parallel Problem Solving from Nature — PPSN V, 1998, p. 292–301. doi:10.1007/BFb0056872.

[33] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, L. N. Vicente, Direct multisearch for multiobjective optimization, SIAM Journal on Optimization 21 (3) (2011) 1109–1140. doi:10.1137/10079731X.

[34] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2002) 201–213. doi:10.1007/s101070100263.

[35] M. Ansary, G. Panda, A modified quasi-newton method for vector optimization problem, Optimization 64 (11) (2015) 2289–2306. doi:10.1080/02331934.2014.947500.

[36] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), volume 1, 2002, pp. 825–830. doi:10.1109/CEC.2002.1007032.

[37] A. Farhang-Mehr, S. Azarm, Diversity assessment of Pareto optimal solution sets: An entropy approach, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), volume 1, 2002, pp. 723–728. doi:10.1109/CEC.2002.1007015.

[38] J. Fliege, L. M. G. n. Drummond, B. F. Svaiter, Newton's method for multiobjective optimization, SIAM Journal on Optimization 20 (2) (2009) 602–626. doi:10.1137/08071692X.

[39] K. Ikeda, H. Kita, S. Kobayashi, Failure of Pareto-based moeas: Does non-dominated really mean near to optimal?, in: IEEE Congress on Evolutionary Computation (CEC), volume 2, IEEE, 2001, pp. 957–962. doi:10.1109/CEC.2001.934293.

[40] J. Lis, A. E. Eiben, Multi-sexual genetic algorithm for multiobjective optimization, in: Proc. IEEE Int. Conf. Evol. Comput., 1997, pp. 59–64. doi:10.1109/ICEC.1997.592269.

[41] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary Computation 10 (3) (2002) 263–282. doi:10.1162/106365602760234108.

[42] K. Mita, E. H. Fukuda, N. Yamashita, Nonmonotone line searches for unconstrained multiobjective optimization problems, Journal of Global Optimization 75 (1) (2019) 63–90. doi:10.1007/s10898-019-00802-0.

[43] A. Molyneaux, D. Favrat, G. B. Leyland, A new clustering evolutionary multi-objective optimisation technique, in: Third International Symposium on Adaptative Systems, Institute of Cybernetics, Mathematics and Physics, 2001, pp. 41–47.

[44] E. Miglierina, E. Molho, M. C. Recchioni, Box-constrained multi-objective optimization: A gradient-like method without "a priori" scalarization, European Journal of Operational Research 188 (2008) 662–682. doi:10.1016/j.ejor.2007.05.015.

[45] D. A. Van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, Ph.D. thesis, 1999.

[46] K. Socha, M. Kisiel-Dorohinicki, Agent-based evolutionary multiobjective optimization, in: D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, M. Shackleton (Eds.), Proc. Congr. Evol. Comput., volume 1, 2002, pp. 109–114.

[47] K. C. Tan, E. F. Khor, T. H. Lee, Y. J. Yang, A tabu-based exploratory evolutionary algorithm for multiobjective optimization, Artificial Intelligence Review 19 (2003) 231–260. doi:10.1023/A:1022863019997.

[48] P. L. Toint, Test problems for partially separable optimization and results for the routine pspmin, The University of Namur, Department of Mathematics, Belgium, Tech. Rep (1983).

[49] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput. 8 (2) (2000) 173–195. doi:10.1162/106365600568202.

[50] Y. Wang, X. Li, J. Wang, A neurodynamic optimization approach to supervised feature selection via fractional programming, Neural Networks 136 (2021). doi:10.1016/j.neunet.2021.01.004.

Appendix

A. Detailed formulation of G_j used in the robust multiobjective optimization problem (rMOP)

1. **AP1** [35]

$$\min_x \begin{cases} f_1(x) = 0.25\left((x_1-1)^4 + 2(x_2-2)^4\right), \\ f_2(x) = \exp\left(\dfrac{x_1+x_2}{2}\right) + x_1^2 + x_2^2, \\ f_3(x) = \dfrac{1}{6}\left(\exp(-x_1) + 2\exp(-x_2)\right). \end{cases}$$

2. **AP4** [35]

$$\min_x \begin{cases} f_1(x) = \dfrac{1}{9}\left((x_1-1)^4 + 2(x_2-2)^4 + 3(x_3-3)^4\right), \\ f_2(x) = \exp\left(\dfrac{x_1+x_2+x_3}{3}\right) + x_1^2 + x_2^2 + x_3^2, \\ f_3(x) = \dfrac{1}{12}\left(3\exp(-x_1) + 4\exp(-x_2) + 3\exp(-x_3)\right). \end{cases}$$

3. **DTLZ1** [36]

$$\min_x \begin{cases} f_1(x) = 0.5(1+g(x))x_1x_2, \\ f_2(x) = 0.5(1+g(x))x_1(1-x_2), \\ f_3(x) = 0.5(1+g(x))(1-x_1), \end{cases}$$

$$\text{where } g(x) = 100\left(5 + \sum_{i=3}^{n}\left[(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]\right).$$

4. **DTLZ2** [36]

$$\min_x \begin{cases} f_1(x) = (1+g(x))\cos\left(\dfrac{x_1\pi}{2}\right)\cos\left(\dfrac{x_2\pi}{2}\right), \\ f_2(x) = (1+g(x))\cos\left(\dfrac{x_1\pi}{2}\right)\sin\left(\dfrac{x_2\pi}{2}\right), \\ f_3(x) = (1+g(x))\sin\left(\dfrac{x_1\pi}{2}\right), \end{cases}$$

$$\text{where } g(x) = \sum_{i=3}^{n}(x_i-0.5)^2.$$

5. **DTLZ3** [36]

$$\min_x \begin{cases} f_1(x) = (1+g(x))\cos\left(\dfrac{x_1\pi}{2}\right)\cos\left(\dfrac{x_2\pi}{2}\right), \\ f_2(x) = (1+g(x))\cos\left(\dfrac{x_1\pi}{2}\right)\sin\left(\dfrac{x_2\pi}{2}\right), \\ f_3(x) = (1+g(x))\sin\left(\dfrac{x_1\pi}{2}\right), \end{cases}$$

$$\text{where } g(x) = 100\left(5 + \sum_{i=3}^{n}\left[(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]\right).$$

6. **DTLZ4** [36]

$$\min_x \begin{cases} f_1(x) = (1+g(x))\cos\left(\dfrac{x_1^2\pi}{2}\right)\cos\left(\dfrac{x_2^2\pi}{2}\right), \\ f_2(x) = (1+g(x))\cos\left(\dfrac{x_1^2\pi}{2}\right)\sin\left(\dfrac{x_2^2\pi}{2}\right), \\ f_3(x) = (1+g(x))\sin\left(\dfrac{x_1^2\pi}{2}\right), \end{cases}$$

$$\text{where } g(x) = \sum_{i=3}^{n}(x_i - 0.5)^2.$$

7. **FA1** [37]

$$\min_x \begin{cases} f_1(x) = \dfrac{1-\exp(-4x_1)}{1-\exp(-4)}, \\ f_2(x) = (x_2+1)\left(1 - \sqrt{\dfrac{f_1(x)}{x_2+1}}\right), \\ f_3(x) = (x_3+1)\left(1 - \left(\dfrac{f_1(x)}{x_3+1}\right)^{0.1}\right). \end{cases}$$

8. **Far1** [37]

$$\min_x \begin{cases} f_1(x) = -2\exp\left(15\left(-(x_1-0.1)^2 - x_2^2\right)\right) - \exp\left(20\left(-(x_1-0.6)^2 - (x_2-0.6)^2\right)\right) \\ \qquad + \exp\left(20\left(-(x_1+0.6)^2 - (x_2-0.6)^2\right)\right) + \exp\left(20\left(-(x_1-0.6)^2 - (x_2+0.6)^2\right)\right) \\ \qquad + \exp\left(20\left(-(x_1+0.6)^2 - (x_2+0.6)^2\right)\right), \\ f_2(x) = 2\exp\left(20\left(-x_1^2 - x_2^2\right)\right) + \exp\left(20\left(-(x_1-0.4)^2 - (x_2-0.6)^2\right)\right) \\ \qquad - \exp\left(20\left(-(x_1+0.5)^2 - (x_2-0.7)^2\right)\right) - \exp\left(20\left(-(x_1-0.5)^2 - (x_2+0.7)^2\right)\right) \\ \qquad + \exp\left(20\left(-(x_1+0.4)^2 - (x_2+0.8)^2\right)\right). \end{cases}$$

9. **FDS** [38]

$$\min_x \begin{cases} f_1(x) = \dfrac{1}{n^2}\displaystyle\sum_{i=1}^{n} i(x_i - i)^4, \\ f_2(x) = \exp\left(\dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i\right) + \displaystyle\sum_{i=1}^{n} x_i^2, \\ f_3(x) = \dfrac{1}{n(n+1)}\displaystyle\sum_{i=1}^{n} i(n-i+1)\exp(-x_i). \end{cases}$$

10. **IKK1** [39]

$$\min_x \begin{cases} f_1(x) = x_1^2, \\ f_2(x) = (x_1 - 20)^2, \\ f_3(x) = x_2^2. \end{cases}$$

11. **LE1** [40]

$$\min_x \begin{cases} f_1(x) = (x_1^2 + x_2^2)^{0.125}, \\ f_2(x) = ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{0.25}. \end{cases}$$

12. **LTDZ** [41]

$$\min_x \begin{cases} f_1(x) = -\left(3 - (1 + x_3) \cos\left(\frac{x_1 \pi}{2}\right) \cos\left(\frac{x_2 \pi}{2}\right)\right), \\ f_2(x) = -\left(3 - (1 + x_3) \cos\left(\frac{x_1 \pi}{2}\right) \sin\left(\frac{x_2 \pi}{2}\right)\right), \\ f_3(x) = -\left(3 - (1 + x_3) \sin\left(\frac{x_1 \pi}{2}\right)\right). \end{cases}$$

13. **MGH33** [42]

$$\min_x \left\{ f_{i+1}(x) = \left((i+1) \sum_{j=1}^{n} j \cdot x_j - 1\right)^2, \quad i = 0, 1, \dots, n-1. \right.$$

14. **MLF1** [43]

$$\min_x \begin{cases} f_1(x) = \left(1 + \frac{x_1}{20}\right) \sin(x_1), \\ f_2(x) = \left(1 + \frac{x_1}{20}\right) \cos(x_1). \end{cases}$$

15. **MMR4** [44]

$$\min_x \begin{cases} f_1(x) = x_1 - 2x_2 - x_3 - \dfrac{36}{2x_1 + x_2 + 2x_3 + 1}, \\ f_2(x) = -3x_1 + x_2 - x_3. \end{cases}$$

16. **MOP2** Van Veldhuizen [45]

$$\min_x \begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^{n} \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right), \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^{n} \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right). \end{cases}$$

17. **MOP3** [45]

$$\min_x \begin{cases} f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2, \\ f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2, \end{cases}$$

$$\text{where} \quad \begin{cases} A_1 = 0.5 \sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2), \\ A_2 = 1.5 \sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2), \\ B_1 = 0.5 \sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2), \\ B_2 = 1.5 \sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2). \end{cases}$$

18. **MOP5** [45]

$$\min_x \begin{cases} f_1(x) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2), \\ f_2(x) = \dfrac{(3x_1 - 2x_2 + 4)^2}{8} + \dfrac{(x_1 - x_2 + 1)^2}{27} + 15, \\ f_3(x) = \dfrac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2). \end{cases}$$

19. **SK1** [46]

$$\min_x \begin{cases} f_1(x) = x_1^4 + 3x_1^3 - 10x_1^2 - 10x_1 - 10, \\ f_2(x) = 0.5x_1^4 - 2x_1^3 - 10x_1^2 + 10x_1 - 5. \end{cases}$$

20. **TKLY1** [47]

$$\min_x \begin{cases} f_1(x) = x_1, \\ f_2(x) = \dfrac{A_2 A_3 A_4}{x_1}, \end{cases}$$

$$\text{where } A_i = 2 - \exp\left(-\left(\frac{x_i - 0.1}{4 \times 10^{-3}}\right)^2\right) - 0.8\exp\left(-\left(\frac{x_i - 0.9}{0.4}\right)^2\right), \quad i = 2, 3, 4.$$

21. **Toi9** [48]

$$\min_x \begin{cases} f_1(x) = (2x_1 - 1)^2 + x_2^2, \\ f_i(x) = i(2x_{i-1} - x_i)^2 - (i-1)x_{i-1}^2 + ix_i^2, \quad i = 2, \ldots, n-1, \\ f_n(x) = n(2x_{n-1} - x_n)^2 - (n-1)x_{n-1}^2. \end{cases}$$

22. **ZDT6** [49]

$$\min_x \begin{cases} f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1), \\ f_2(x) = g(x)\left(1 - \left(\dfrac{f_1(x)}{g(x)}\right)^2\right), \end{cases}$$

$$\text{where } g(x) = 1 + 9\left(\frac{1}{n-1}\sum_{i=2}^{n} x_i\right)^{0.25}.$$

B. Experimental results on the robust multiobjective optimization problem (rMOP)

Table 3: Detailed experimental results of the compared algorithms on the problem (rMOP)

Problem	Algo.	Time (ms)	Iter.	Stepsize	H-Ev	Global_Nond.	Purity	Hypervolume	Spread Γ
AP1	MPGE	**576.2658**	**47.8000**	**0.5891**	**150.8800**	**100**	**1.0000**	**34.7029**	1.6146
	AMPG1	2996.7546	276.0600	0.4094	828.1800	95	**1.0000**	32.5752	1.6781
	AMPG2	3243.8221	289.5600	0.3935	868.6800	94	**1.0000**	32.6701	1.6807
	PGM	4614.6050	999.0500	0.0019	2997.1500	4	0.8000	27.3362	**1.1273**
	accPGM	2922.4009	460.0800	0.0019	2757.4800	0	0.0000	11.2525	∞
AP4	MPGE	**437.2190**	**126.3500**	0.4829	**398.8000**	**76**	**1.0000**	21554.5277	1.6940
	AMPG1	764.5085	232.9900	**0.6839**	698.9700	47	0.9792	22018.9118	0.9868
	AMPG2	747.7667	227.1000	0.5602	681.3000	47	0.9592	**22690.7836**	1.5195
	PGM	3149.0390	979.3700	0.0021	2938.1100	7	0.5385	22077.6497	1.1691
	accPGM	3085.6504	496.3000	0.0021	2974.8000	2	0.2222	21657.1163	**0.9728**
DTLZ1	MPGE	2776.1323	710.0200	0.0012	2166.3300	**12**	**1.0000**	**9769262.7844**	1.0982
	AMPG1	11.3269	2.9700	0.3384	8.9100	10	0.3226	9657920.3643	1.5539
	AMPG2	**10.9528**	**2.9200**	**0.3467**	**8.7600**	5	0.1613	9657918.9377	1.4993
	PGM	1162.5842	309.9000	0.0000	929.7000	7	0.2917	9766502.4592	1.3567
	accPGM	1508.9393	211.5100	0.0000	1266.0600	3	0.1364	9765531.9542	**0.8335**
DTLZ2	MPGE	1087.9960	239.7164	0.9550	892.0900	66	**1.0000**	8.6311	1.1578
	AMPG1	**614.0677**	**165.5400**	**45977.6281**	**496.6200**	**94**	**1.0000**	8.5853	1.4611
	AMPG2	731.8608	197.5900	101.5040	592.7700	92	0.9892	8.5701	1.5345
	PGM	1841.1189	491.8700	0.1033	1475.6100	82	0.9213	**8.7393**	0.9712
	accPGM	1937.2285	271.8900	0.1034	1628.3400	62	0.7294	8.7093	**0.9270**

Continued on next page

Problem	Algo.	Time (ms)	Iter.	Stepsize	H-Ev	Global_Nond.	Purity	Hypervolume	Spread Γ
DTLZ3	MPGE	3570.3088	921.5960	0.0004	2769.7500	6	**0.8571**	223797870.0992	1.0169
	AMPG1	11.3796	3.0500	**0.8147**	9.1500	**16**	0.5333	221535531.7840	1.5093
	AMPG2	**10.5341**	**2.8200**	0.7996	**8.4600**	10	0.3333	221532462.1351	1.5120
	PGM	779.5435	206.0400	0.0000	618.1200	7	0.2917	**223978673.7612**	1.5284
	accPGM	1214.3230	169.9100	0.0000	1016.4600	3	0.1034	223958921.1001	**0.9995**
DTLZ4	MPGE	**1139.6110**	**243.4375**	0.9511	**935.9400**	47	**1.0000**	10.6182	1.0797
	AMPG1	1226.1603	328.3700	56783.8462	985.1100	**79**	0.9518	10.6352	1.7001
	AMPG2	1385.9015	369.9300	**187356477.8866**	1109.7900	75	0.9375	10.7074	1.5772
	PGM	2492.1448	669.3800	0.0681	2008.1400	**79**	0.9518	**10.8437**	1.0962
	accPGM	2520.3214	353.2500	0.0682	2116.5000	47	0.6104	10.8331	**0.9947**
FA1	MPGE	55.6687	17.4800	0.8834	52.5200	**100**	**1.0000**	**0.3708**	1.1164
	AMPG1	**27.6792**	**8.6900**	**3.4302**	**26.0700**	93	**1.0000**	0.3707	1.0744
	AMPG2	46.6674	14.6900	1.4148	44.0700	93	**1.0000**	0.3704	1.0615
	PGM	556.6778	174.4900	0.1645	523.4700	90	0.9677	0.3676	0.9484
	accPGM	764.3446	123.9700	0.1651	740.8200	89	0.9271	0.3658	**0.9471**
Far1	MPGE	332.5452	134.1900	0.5035	311.1000	**54**	**0.9643**	5.1740	1.2930
	AMPG1	**108.5667**	**50.9500**	**2.3930**	**101.9000**	19	0.4634	4.7415	1.3233
	AMPG2	353.9755	164.9900	1.9426	329.9800	25	0.7812	4.7804	1.2965
	PGM	1156.5933	535.1400	0.0639	1070.2800	47	0.9592	**5.3294**	**1.1359**
	accPGM	1336.4776	320.6500	0.0663	1280.6000	31	0.6596	5.3292	1.1422
FDS	MPGE	2505.2253	490.8700	0.2269	1869.5600	**100**	**1.0000**	**57448.9502**	0.9908
	AMPG1	**808.4463**	**198.5300**	**0.3344**	**595.5900**	**100**	**1.0000**	57418.5984	0.9328
	AMPG2	1075.2993	262.6100	0.2153	787.8300	**100**	**1.0000**	57437.7491	0.9361
	PGM	4098.8480	1000.0000	0.0042	3000.0000	11	0.5238	50227.9706	**0.9014**
	accPGM	3748.9198	500.0000	0.0044	2997.0000	0	0.0000	43405.6754	∞
IKK1	MPGE	421.7982	138.7000	0.8622	416.6900	80	0.9524	138617860.6159	1.4575
	AMPG1	**228.2446**	**73.5500**	**2.0749**	**220.6500**	**97**	**0.9798**	138967230.9155	1.5455
	AMPG2	233.3914	76.4400	2.0650	229.3200	94	0.9495	**139401167.8794**	**1.4116**
	PGM	1648.5175	541.4200	0.1250	1624.2600	56	0.8889	132604539.0238	1.6193
	accPGM	2242.7557	376.3000	0.1250	2254.8000	18	0.4000	129512202.5507	1.4803
LE1	MPGE	83.1640	39.5100	0.9671	79.0200	0	0.0000	**0.0115**	∞
	AMPG1	**17.9688**	**8.5600**	**7.0109**	**17.1200**	**4**	**1.0000**	**0.0115**	∞
	AMPG2	21.8118	10.3800	4.5784	20.7600	0	0.0000	**0.0115**	∞
	PGM	87.7143	41.7700	0.9038	83.5400	0	0.0000	**0.0115**	∞
	accPGM	168.2198	41.7700	0.9038	165.0800	0	0.0000	**0.0115**	∞
LTDZ	MPGE	155.9028	47.6000	0.7215	142.8000	**82**	**0.9762**	0.8540	1.0029
	AMPG1	**28.9210**	**8.8700**	**386.0246**	**26.6100**	79	0.9634	**0.8548**	0.9719
	AMPG2	73.5266	22.5700	0.9528	67.7100	81	0.9759	0.8543	0.9756
	PGM	748.2972	229.3800	0.1106	688.1400	65	0.8667	0.8471	0.8274
	accPGM	950.7292	150.2700	0.1107	898.6200	58	0.8169	0.8468	**0.8234**
MGH33	MPGE	679.1152	205.4200	0.9296	637.0200	**100**	**1.0000**	1.9921	1.4644
	AMPG1	**76.0924**	**23.8400**	**818.4424**	**71.5200**	**100**	**1.0000**	2.0144	1.6152
	AMPG2	187.3719	58.3900	10.3787	175.1700	**100**	**1.0000**	**2.0172**	1.6162
	PGM	3046.0150	954.8700	0.0228	2864.6100	11	0.4583	1.9165	1.2429
	accPGM	2912.9376	467.5200	0.0237	2802.1200	6	0.2727	1.9050	**0.8672**
MLF1	MPGE	**3.9843**	**1.9200**	0.2989	**3.8400**	38	**1.0000**	3.2991	1.1949
	AMPG1	6.3098	3.0800	**0.7488**	6.1600	**39**	0.9750	**3.3094**	1.1734
	AMPG2	6.3220	3.0800	**0.7488**	6.1600	37	0.9250	**3.3094**	1.1571
	PGM	46.0414	22.3100	0.1666	44.6200	13	**1.0000**	3.2619	**0.9758**
	accPGM	68.8691	17.4100	0.1668	67.6400	9	0.6923	3.2618	1.0269
	MPGE	**11.7586**	**5.4400**	0.7842	**10.8800**	**90**	**1.0000**	86.9193	1.0831
	AMPG1	13.1029	6.1300	**2.1339**	12.2600	89	0.9889	**87.3719**	1.0394

Continued on next page

Problem	Algo.	Time (ms)	Iter.	Stepsize	H-Ev	Global_Nond.	Purity	Hypervolume	Spread Γ
MMR4	AMPG2	14.1681	6.6500	1.4348	13.3000	89	**1.0000**	87.2275	1.0494
	PGM	38.0369	17.6700	0.5930	35.3400	86	0.9451	87.3628	1.0097
	accPGM	71.6703	17.6700	0.5930	68.6800	88	0.9670	87.3628	**0.9936**
MOP2	MPGE	**14.9876**	**6.9600**	0.8391	**13.9200**	**100**	**1.0000**	**0.5040**	0.8748
	AMPG1	513.6472	238.6000	**2.2206**	477.2000	77	**1.0000**	0.5017	**0.8746**
	AMPG2	358.7017	166.9400	1.6603	333.8800	85	**1.0000**	0.5017	0.9221
	PGM	97.2019	45.1900	0.1867	90.3800	90	0.9000	0.5040	0.9167
	accPGM	187.2848	45.1600	0.1867	178.6400	90	0.9000	0.5040	0.9257
MOP3	MPGE	**26.8931**	**11.8200**	**0.4636**	**25.1000**	**55**	**0.9821**	505.5453	1.5268
	AMPG1	44.1653	20.6700	0.4298	41.3400	39	0.9512	497.1894	1.0967
	AMPG2	47.0185	21.9100	0.3788	43.8200	39	0.9512	503.7323	**1.0810**
	PGM	516.8419	241.3000	0.0209	482.6000	18	0.8182	**505.8076**	1.2930
	accPGM	780.0390	186.4500	0.0211	743.8000	15	0.7143	505.8074	1.3434
MOP5	MPGE	78.2217	22.7300	**0.8565**	71.7900	**64**	0.9697	1.2824	1.1439
	AMPG1	**64.9881**	20.0300	0.6573	60.0900	60	0.9677	1.2827	**1.0549**
	AMPG2	65.4959	**20.0100**	0.6569	**60.0300**	62	0.9394	**1.2831**	1.0676
	PGM	872.8161	269.3200	0.0642	807.9600	54	**1.0000**	1.2821	1.0954
	accPGM	1358.5140	215.3200	0.0643	1288.9200	20	0.3704	1.2821	1.1831
SK1	MPGE	1499.6446	702.3500	0.0011	1405.8100	45	**1.0000**	72740.7978	1.6924
	AMPG1	38.8284	**18.6600**	**0.0883**	**37.3200**	83	0.8300	**73504.6260**	1.7960
	AMPG2	**38.7953**	**18.6600**	**0.0883**	**37.3200**	**92**	0.9200	**73504.6260**	1.8258
	PGM	1932.8558	921.4800	0.0002	1842.9600	6	**1.0000**	71925.2589	**0.5469**
	accPGM	1906.1956	465.6000	0.0002	1860.4000	3	0.5000	71923.0473	0.7848
TKLY1	MPGE	1210.6492	431.7900	0.0405	1101.5900	**41**	**0.8723**	**67.4317**	1.6020
	AMPG1	**257.9350**	**118.1000**	**0.1088**	**236.2000**	21	0.4565	67.1369	1.2807
	AMPG2	355.7288	158.7100	0.0978	317.4200	23	0.4107	66.9437	1.0804
	PGM	1899.9565	869.4400	0.0038	1738.8800	19	0.4043	65.6382	1.0554
	accPGM	1882.3088	441.3600	0.0042	1763.4400	5	0.1389	65.3250	**0.9970**
Toi9	MPGE	1155.2413	**191.6600**	0.2879	986.2700	95	0.9794	4442.4761	1.3319
	AMPG1	1498.8140	322.3800	**0.7222**	1289.5200	94	0.9792	4443.4869	1.4427
	AMPG2	**1043.5319**	224.3800	0.3163	**897.5200**	**97**	**1.0000**	**4446.9624**	1.6204
	PGM	3932.0160	842.3800	0.0081	3369.5200	47	0.6184	4431.6766	1.5510
	accPGM	4180.3569	454.9200	0.0081	3635.3600	19	0.2533	4418.5017	**1.2154**
ZDT6	MPGE	2849.4601	977.8300	0.0229	2043.8000	1	0.5000	4.3694	∞
	AMPG1	**170.4963**	**61.7900**	**0.3336**	**123.5800**	**34**	**1.0000**	4.4400	0.9912
	AMPG2	571.2491	203.1300	0.1312	406.2600	33	**1.0000**	**4.4441**	**0.9287**
	PGM	2770.0384	1000.0000	0.0016	2000.0000	0	0.0000	1.2179	∞
	accPGM	1985.4818	383.6500	0.0022	1532.6000	0	0.0000	1.2152	∞

C. Experimental results on the biobjective supervised feature selection problems (B-SFS)

Table 4: Detailed experimental results of the compared algorithms on the problem (B-SFS)

Dataset	Algo.	Time (ms)	Iter	Stepsize	Global_Nond.	Hypervolume	Purity	Spread Γ
Arrhythmia	MPGE	256430.7740	97.2100	0.8818	**100**	0.0024	**1.0000**	1.2275
	AMPG1	98360.2071	**35.4200**	1.5301	**100**	**0.0025**	**1.0000**	1.1256
	AMPG2	**86501.0178**	35.4800	**1.5314**	**100**	0.0025	**1.0000**	**1.1252**
	PGM	1157697.8619	472.3700	0.1250	99	0.0025	0.9900	1.2463
	accPGM	484339.8561	206.6100	0.1250	**100**	0.0024	**1.0000**	1.2047
	MPGE	1759.5639	304.7700	0.6446	**100**	0.0060	**1.0000**	1.2588
	AMPG1	313.1037	**60.1600**	**2.1024**	**100**	**0.0060**	**1.0000**	1.2396

Continued on next page

Dataset	Method	Time (ms)	Iter	Stepsize	Global_Nond	Hypervolume	Purity	Spread Γ
CMC	AMPG2	**312.4244**	**60.1600**	2.1024	**100**	**0.0060**	**1.0000**	1.2396
	PGM	4418.8061	767.8400	0.1250	89	0.0060	**1.0000**	**1.2245**
	accPGM	2079.3729	311.3900	0.1250	94	0.0060	**1.0000**	1.2489
Darwin	MPGE	560401.7639	54.5800	0.9035	**100**	0.0025	**1.0000**	1.0510
	AMPG1	277142.0691	26.6600	**1.1928**	**100**	0.0025	**1.0000**	1.0826
	AMPG2	**274931.2272**	**26.5800**	1.1829	**100**	**0.0025**	**1.0000**	1.0909
	PGM	3033667.8884	293.1600	0.1250	**100**	0.0024	**1.0000**	**1.0112**
	accPGM	2090838.0499	181.2700	0.1250	**100**	0.0024	**1.0000**	1.0354
Gallstone	MPGE	1273.1335	82.3800	0.9199	**100**	0.0189	**1.0000**	1.1230
	AMPG1	**385.0532**	**33.5800**	**1.5025**	**100**	**0.0195**	**1.0000**	**1.0641**
	AMPG2	459.8124	**33.5800**	1.5024	**100**	0.0195	**1.0000**	1.0641
	PGM	6400.8465	488.5500	0.1250	**100**	0.0188	**1.0000**	1.1245
	accPGM	3116.1742	224.3000	0.1250	**100**	0.0188	**1.0000**	1.1309
Heart	MPGE	380.1007	53.4800	0.9492	**100**	**0.0273**	**1.0000**	1.1799
	AMPG1	**165.9007**	**27.6700**	**1.2398**	**100**	0.0272	**1.0000**	1.1116
	AMPG2	203.7466	27.6900	1.2397	**100**	0.0272	**1.0000**	**1.1114**
	PGM	2540.0634	330.8300	0.1250	99	0.0273	**1.0000**	1.1766
	accPGM	1505.9578	208.9600	0.1250	**100**	0.0273	**1.0000**	1.1633
Ionosphere	MPGE	510.0865	36.6100	0.9158	**100**	0.0296	**1.0000**	1.0054
	AMPG1	**334.3382**	33.0900	**1.1610**	**100**	0.0291	**1.0000**	0.9481
	AMPG2	394.7728	**32.6800**	1.1581	**100**	0.0291	**1.0000**	**0.9453**
	PGM	2835.0387	218.5500	0.1250	**100**	0.0297	**1.0000**	1.0239
	accPGM	1844.0850	160.6200	0.1250	**100**	**0.0297**	**1.0000**	1.0085
Musk-v1	MPGE	36199.8582	67.4700	0.9164	**100**	0.0144	**1.0000**	1.2644
	AMPG1	15900.1191	**29.8600**	**1.3398**	**100**	**0.0146**	**1.0000**	**1.1855**
	AMPG2	**15735.3804**	**29.8600**	1.3394	**100**	0.0146	**1.0000**	1.1856
	PGM	201274.9066	379.4700	0.1250	**100**	0.0144	**1.0000**	1.2494
	accPGM	95059.7835	182.0200	0.1250	**100**	0.0143	**1.0000**	1.2538
Parkinsons	MPGE	304.4090	41.6400	0.8916	**100**	0.0250	**1.0000**	1.0658
	AMPG1	200.0549	**29.5400**	1.2039	**100**	0.0244	**1.0000**	**1.0307**
	AMPG2	**195.3287**	29.6700	**1.2046**	**100**	0.0244	**1.0000**	1.0424
	PGM	1686.3651	240.2700	0.1250	**100**	0.0250	**1.0000**	1.1260
	accPGM	1428.8263	175.3900	0.1250	**100**	**0.0250**	**1.0000**	1.1120