manuscript No.
(will be inserted by the editor)

Preconditioned Barzilai-Borwein Methods for Multiobjective Optimization Problems

Jian Chen · Liping Tang · Xinmin Yang

Received: date / Accepted: date

Abstract Preconditioning is a powerful approach for solving ill-conditioned problems in optimization, where a preconditioning matrix is used to reduce the condition number and speed up the convergence of first-order method. Unfortunately, it is impossible to capture the curvature of all objective functions with a single preconditioning matrix in multiobjective optimization. Instead, second-order methods for multiobjective optimization problems (MOPs) use different matrices for objectives in direction-finding subproblems, leading to a prohibitive per-iteration cost. To balance per-iteration cost and better curvature exploration, we propose a preconditioned Barzilai-Borwein descent method for MOPs (PBBMO). In the direction-finding subproblems, we employ a scale matrix to explore the curvature of an implicit scalarization function. The Barzilai-Borwein method is then applied to the matrix metric to tune the gradients of the objective functions, which can also be considered as an extra diagonal preconditioner based on the scale matrix for each objective, and mitigates the effect of imbalances among objectives. From a preconditioning perspective, we use BFGS update formula to approximate a trade-off of Hessian matrices. Under mild assumption, we give a simple convergence analysis for the Barzilai-Borwein quasi-Newton method. Finally, comparative numerical results confirm the efficiency of the proposed method, even when applied to large-scale and ill-conditioned problems.

Keywords Multiobjective optimization · Preconditioning · Barzilai-Borwein's method · BFGS

Mathematics Subject Classification (2010) 90C29 · 90C30

J. Chen
National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China, and School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
chenjian_math@163.com
L.P. Tang
National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China
tanglipings@163.com
✉X.M. Yang
National Center for Applied Mathematics in Chongqing, and School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China
xmyang@cqnu.edu.cn
The previous version is available at https://arxiv.org/abs/2309.06929.

1 Introduction

An unconstrained multiobjective optimization problem can be stated as follows:

$$\min_{x\in\mathbb{R}^n} F(x), \tag{MOP}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable function. In multiobjective optimization, the primary goal is to simultaneously optimize multiple objective functions. In general, finding a single solution that optimizes all objectives is infeasible. Therefore, optimality is defined by *Pareto optimality* or *efficiency*. A solution is considered Pareto optimal or efficient if no objective can be improved without sacrificing the others. As society and the economy advance, the applications of this type of problem have expanded into various domains, including engineering [24], economics [17], management science [13], and machine learning [36], among others.

Solution strategies play a pivotal role in the realm of applications involving multiobjective optimization problems (MOPs). Over the past two decades, multiobjective gradient descent methods have garnered increasing attention within the multiobjective optimization community. These methods generate descent directions by solving subproblems, eliminating the necessity for predefined parameters. Subsequently, line search techniques are employed along the descent direction to ensure sufficient improvement for all objectives. Attouch et al. [3] highlighted an appealing characteristic of this method in fields such as game theory, economics, social science, and management:*it improves each of the objective functions*. As far as we know, the study of multiobjective gradient descent methods can be traced back to the pioneering works by Mukai [27]. and Fliege and Svaiter [15]. The later clarified that the multiobjective steepest descent direction reduces to the steepest descent direction when dealing with a single objective. This observation inspired researchers to extend ordinary numerical algorithms for solving MOPs (see, e.g., [2, 4, 5, 14, 16, 18, 23, 25, 31, 35] and references therein).

1.1 First-order methods

Fliege and Svaiter [15] introduced the steepest descent method for MOPs (SDMO). The steepest descent direction is the optimal solution of the following subproblem:

$$\min_{d\in\mathbb{R}^n} \max_{i=1,2,\dots,m} \langle \nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2.$$

This subproblem can be reformulated as a quadratic problem and efficiently solved through its dual [36]. Subsequently, Graña Drummond and Iusem extended this method to constrained MOPs, proposing the projected gradient method for MOPs. For multiobjective composite optimization problems, Tanabe et al. [39] extended the proximal gradient method to MOPs. Analogous to most first-order methods for single-objective optimization problems (SOPs), these MOP counterparts enjoy cheap per-step computation cost but suffer slow convergence, especially for ill-conditioned problems. In response to this challenge, some classic methods were extended to MOPs, including Barzilai-Borwein's method [26], nonlinear conjugate gradient method [23], and Nesterov's accelerated method [37, 38, 40]. In addition to issues stemming from ill-conditioning, another inherent challenge arises from imbalances among objective functions. Chen et al. [7] highlighted that even when all objective functions are not ill-conditioned, imbalances among them can lead to slow convergence of first-order methods for MOPs. To address this issue, Chen et al. [7] applied Barzilai-Borwein's method to alleviate the impact of imbalances. They demonstrated that the Barzilai-Borwein proximal gradient method [8] converges at a rate of $\sqrt{1-\min_{i=1,2,\dots,m}\{\mu_i/L_i\}}$, where μ_i and L_i are the constants of strong convexity and smoothness of f_i, respectively. It is worth noting that the performance of this type of method also depends on the conditioning of problems.

1.2 Second-order methods

For ill-conditioned problems, Newton's method is much more efficient. Fliege et al. [14] proposed Newton's method for MOPs (NMO). The Newton direction is the optimal solution of

$$\min_{d\in\mathbb{R}^n} \max_{i=1,2,\ldots,m} \langle \nabla F_i(x), d\rangle + \frac{1}{2}\left\langle d, \nabla^2 F_i(x)d\right\rangle.$$

It has been proven that NMO possesses desirable properties [14], including local superlinear and quadratic convergence under standard assumptions. Furthermore, quasi-Newton methods have garnered considerable attention [31, 33–35] and demonstrate local superlinear convergence. While these methods for MOPs are superior in capturing the local geometry of objective functions and offering rapid convergence, the per-step cost is computationally expensive. In contrast to their single-objective problem counterparts, the high per-step computation cost arises not only from the computation of Hessian matrices and their inverses but also from the costly subproblems.[1] In order to reduce the computational cost of the subproblem, Ansary and Panda [1] utilized a single quasi-Newton approximation to approximate all Hessian matrices. Subsequently, this idea was adopted by Chen et al. [6] and Lapucci and Mansueto [22]. While efficient subproblem solving is possible with this approximation, Chen et al. [6] identified a limitation: The monotone line search cannot accept a unit step size, thereby hindering superlinear convergence.

In summary, the slow convergence observed in first-order methods for MOPs can be primarily attributed to the ill-conditioning and imbalances among objective functions. Meanwhile, second-order methods for MOPs fail to strike the right balance between per iteration cost and overall performance. This leads us to a natural and compelling question: Can we devise an algorithm that maintains an affordable per-step computation cost and is not sensitive to conditioning and imbalances?

1.3 Our contributions

To address this issue, this paper is devoted to the development of a preconditioned Barzilai-Borwein method for MOPs (PBBMO), the main contributions of the paper can be summarized in the following points:

- To achieve a low per-step cost, we use a single preconditioning matrix to capture the local geometry of an implicit scalarization function. In particular, the Barzilai-Borwein rule relative to the matrix metric is embedded to tune the gradients in the direction-finding subproblem, capturing the local geometry of each objective and effectively mitigating the imbalances among objective functions.
- To capture the problem's geometry more effectively, we employ the BFGS update formula to approximate a trade-off Hessian for the multiobjective Newton-type method. This provides a new insight into preconditioning of MOPs. Meanwhile, we emphasize that the Barzilai-Borwein

[1] The subproblems of second-order methods for MOPs can only be solved by reformulating into quadratic constrained problems, see [6], or the prime minimax problems. Solving these problems is much more time-consuming than quadratic dual problems of first-order methods for MOPs.

rule is indispensable in the quasi-Newton method, although the two approaches are rarely used simultaneously in SOPs.

The paper is organized as follows. In section 2, we introduce necessary notations and definitions that will be used later. Section 3 revisits several multiobjective gradient descent methods. Section 4, we propose a generic PBBMO and investigate the choice of Barzilai-Borwein parameters and preconditioning matrix. A Barzilai-Borwein quasi-Newton method for MOPs (BBQNMO) and its convergence analysis are described in section 5. The numerical results are presented in section 6, demonstrating the efficiency of BBQNMO. Finally, we draw some conclusions at the end of the paper.

2 Preliminaries

Throughout this paper, the n-dimensional Euclidean space $\mathbb{R}^n$ is equipped with the inner product $\langle\cdot,\cdot\rangle$ and the induced norm $\|\cdot\|$. Denote $\mathbb{S}^n_{++}(\mathbb{S}^n_{+})$ the set of symmetric (semi-)positive definite matrices in $\mathbb{R}^{n\times n}$. We denote by $JF(x)\in\mathbb{R}^{m\times n}$ the Jacobian matrix of F at x, by $\nabla F_i(x)\in\mathbb{R}^n$ the gradient of F_i at x and by $\nabla^2 F_i(x)\in\mathbb{R}^{n\times n}$ the Hessian matrix of F_i at x. For a positive definite matrix H, the notation $\|x\|_H=\sqrt{\langle x,Hx\rangle}$ is used to represent the norm induced by H on vector x. For simplicity, we denote $[m]:=\{1,2,...,m\}$, and

$$\Delta_m := \left\{\lambda : \sum_{i\in[m]}\lambda_i = 1, \lambda_i \geq 0,\ i\in[m]\right\}$$

the m-dimensional unit simplex. To prevent any ambiguity, we establish the order $\preceq(\prec)$ in $\mathbb{R}^m$ as follows:

$$u\preceq(\prec)v \ \Leftrightarrow\ v-u\in\mathbb{R}^m_{+}(\mathbb{R}^m_{++}),$$

and in $\mathbb{S}^n$ as follows:

$$U\preceq(\prec)V \ \Leftrightarrow\ V-U\in\mathbb{S}^n_{+}(\mathbb{S}^n_{++}).$$

In the following, we introduce the concepts of optimality for (MOP) in the Pareto sense.

Definition 2.1 A vector $x^*\in\mathbb{R}^n$ is called Pareto solution to (MOP), if there exists no $x\in\mathbb{R}^n$ such that $F(x)\preceq F(x^*)$ and $F(x)\neq F(x^*)$.

Definition 2.2 A vector $x^*\in\mathbb{R}^n$ is called weakly Pareto solution to (MOP), if there exists no $x\in\mathbb{R}^n$ such that $F(x)\prec F(x^*)$.

Definition 2.3 A vector $x^*\in\mathbb{R}^n$ is called Pareto critical point of (MOP), if

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

where $\mathrm{range}(JF(x^*))$ denotes the range of linear mapping given by the matrix $JF(x^*)$.

From Definitions 2.1 and 2.2, it is evident that Pareto solutions are always weakly Pareto solutions. The following lemma shows the relationships among the three concepts of Pareto optimality.

Lemma 2.1 (See Theorem 3.1 of [14]) *The following statements hold.*

(i) *If $x \in \mathbb{R}^n$ is a weakly Pareto solution to (MOP), then x is Pareto critical point.*
(ii) *Let every component F_i of F be convex. If $x \in \mathbb{R}^n$ is a Pareto critical point of (MOP), then x is weakly Pareto solution.*
(iii) *Let every component F_i of F be strictly convex. If $x \in \mathbb{R}^n$ is a Pareto critical point of (MOP), then x is Pareto solution.*

Definition 2.4 A differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is L-smooth if

$$\|\nabla h(y) - \nabla h(x)\| \leq \|y - x\|$$

holds for all $x, y \in \mathbb{R}^n$. And h is μ-strongly convex if

$$\langle \nabla h(y) - \nabla h(x), y - x\rangle \geq \mu \|y - x\|^2$$

holds for all $x, y \in \mathbb{R}^n$.

L-smoothness of h implies the following quadratic upper bound:

$$h(y) \leq h(x) + \langle \nabla h(x), y - x\rangle + \frac{L}{2}\|y - x\|^2 \text{ for all } x, y \in \mathbb{R}^n.$$

On the other hand, μ-strong convexity yields the quadratic lower bound:

$$h(y) \geq h(x) + \langle \nabla h(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2 \text{ for all } x, y \in \mathbb{R}^n.$$

When the Euclidean distance is replaced by $\|\cdot\|_B$, where B is a positive definite matrix, then h is L-smooth and μ-strongly convex relative to $\|\cdot\|_B$.

3 Gradient descent methods for MOPs

In this section, we revisit some gradient descent methods for MOPs.

3.1 Steepest descent method

For $x \in \mathbb{R}^n$, the steepest descent direction [15] is defined as the optimal solution of the following subproblem:

$$\min_{d\in\mathbb{R}^n} \max_{i\in[m]} \langle \nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2. \tag{1}$$

Since $d \mapsto \langle \nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2$ is strongly convex for $i \in [m]$, then (1) has a unique minimizer. We denote by $d_{SD}(x)$ and $\theta_{SD}(x)$ the optimal solution and optimal value of (1), respectively. Hence,

$$\theta_{SD}(x) = \min_{d\in\mathbb{R}^n} \max_{i\in[m]} \langle \nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2, \tag{2}$$

and

$$d_{SD}(x) = \arg\min_{d\in\mathbb{R}^n} \max_{i\in[m]} \langle \nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2. \tag{3}$$

Indeed, problem (1) can be equivalently rewritten as the following smooth quadratic problem:

$$\begin{aligned}\min_{(t,d)\in\mathbb{R}\times\mathbb{R}^n} \quad & t+\frac{1}{2}\|d\|^2, \\ \text{s.t.} \quad & \langle \nabla F_i(x), d\rangle \le t,\ i\in[m].\end{aligned} \tag{QP}$$

As described in [7], by KKT conditions, we have

$$d_{SD}(x) = -\sum_{i\in[m]} \lambda_i^{SD}(x)\nabla F_i(x), \tag{4}$$

where $\lambda^{SD}(x) \in \Delta_m$ is the solution to the dual problem:

$$\begin{aligned} -\min_{\lambda} \frac{1}{2}\left\|\sum_{i\in[m]} \lambda_i \nabla F_i(x)\right\|^2 \\ \text{s.t. } \lambda \in \Delta_m. \end{aligned} \tag{DP}$$

The KKT conditions also give

$$\theta(x) = -\frac{1}{2}\left\|\sum_{i\in[m]} \lambda_i^{SD}(x)\nabla F_i(x)\right\|^2 = -\frac{1}{2}\|d_{SD}(x)\|^2, \tag{5}$$

$$\langle \nabla F_i(x), d_{SD}(x)\rangle \le -\|d_{SD}(x)\|^2,\ i\in[m], \tag{6}$$

and

$$\langle \nabla F_i(x), d_{SD}(x)\rangle = t_{SD}(x) = -\|d_{SD}(x)\|^2,\ i\in\mathcal{A}_{SD}(x), \tag{7}$$

where

$$\mathcal{A}_{SD}(x) := \{i : \lambda_i^{SD}(x) > 0,\ i\in[m]\}$$

the set of active constraints at x. The following lemma shows that direction $d_{SD}(x)$ can be used to characterize Pareto critical points of problem (MOP).

Lemma 3.1 (Lemma 1 of [15]) *Let $d_{SD}(x)$ be defined as (3), then the following statements hold.*

(i) *If x is Pareto critical, then $d_{SD}(x) = 0$.*
(ii) *If x is not Pareto critical, then $d_{SD}(x) < 0$.*
(iii) *The mapping $x \mapsto d_{SD}(x)$ is continuous.*

3.2 Newton-type methods

Similar to its counterparts for SOPs, SDMO is sensitive to problem's conditioning. In response to this challenge, Fliege et al. [14] proposed Newton's method for MOPs. Newton's direction is the optimal solution to the following subproblem:

$$\min_{d\in\mathbb{R}^n}\max_{i\in[m]} \langle \nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2_{\nabla^2 F_i(x)}. \tag{8}$$

The dual problem can be expressed as

$$
\begin{aligned}
-\min_{\lambda} \ & \frac{1}{2}\left\|\sum_{i\in[m]}\lambda_i\nabla F_i(x)\right\|^2_{\left[\sum_{i\in[m]}\lambda_i\nabla^2 F_i(x)\right]^{-1}} \\
\text{s.t.} \ & \lambda\in\Delta_m.
\end{aligned}
$$

Denote $\lambda^N(x)\in\Delta_m$ the optimal solution of the dual problem, then

$$
d_N(x) = -\left[\sum_{i\in[m]}\lambda_i^N\nabla^2 F_i(x)\right]^{-1}\left(\sum_{i\in[m]}\lambda_i^N(x)\nabla F_i(x)\right),
$$

and

$$
\langle\nabla F_i(x), d_N(x)\rangle = \theta_N(x) - \frac{1}{2}\|d_N(x)\|^2_{\nabla^2 F_i(x)} \ \text{ for all } i\in\mathcal{A}_N(x). \tag{9}
$$

Since Hessian matrices are not readily available, Qu et al. [35] and Povalej [31] adopted BFGS formulation to approximate the Hessian matrices, namely, replacing $\nabla^2 F_i(x)$ by $B_i(x)$ in (8) for $i\in[m]$. While Newton-type methods offer attractive convergence properties like locally superlinear convergence [14, 31], the high per-step computational cost counteracts the efficiency of outer iterations, resulting in suboptimal performance from a computational perspective.

3.3 Modified quasi-Newton method

In order to balance per-step cost and better curvature exploration, Ansary and Panda [1] utilized a single positive matrix to approximate all the Hessian matrices. They developed the following modified quasi-Newton method, which represents the optimal solution of the following subproblem:

$$
\min_{d\in\mathbb{R}^n}\max_{i\in[m]} \langle\nabla F_i(x), d\rangle + \frac{1}{2}\|d\|^2_{B(x)}, \tag{10}
$$

where $B(x)$ is a positive definite matrix. The subproblem can be efficiently solved via its dual:

$$
\begin{aligned}
-\min_{\lambda} \ & \frac{1}{2}\left\|\sum_{i\in[m]}\lambda_i\nabla F_i(x)\right\|^2_{B(x)^{-1}} \\
\text{s.t.} \ & \lambda\in\Delta_m.
\end{aligned}
$$

Denote $\lambda^{MQN}(x)\in\Delta_m$ an optimal solution of the dual problem, then

$$
d_{MQN}(x) = -B(x)^{-1}\left(\sum_{i\in[m]}\lambda_i^{MQN}(x)\nabla F_i(x)\right),
$$

and

$$
\langle\nabla F_i(x), d_{MQN}(x)\rangle = -\|d_{MQN}(x)\|^2_{B(x)} \ \text{ for all } i\in\mathcal{A}_{MQN}(x). \tag{11}
$$

For each iteration k, once the unique descent direction $d^k_{MQN}\neq 0$ is obtained, the classical Armijo technique is employed for line search.

Algorithm 1: `Armijo_line_search`

Data: $x^k \in \mathbb{R}^n, d^k_{MQN} \in \mathbb{R}^n, JF(x^k) \in \mathbb{R}^{m\times n}, \sigma, \gamma \in (0,1), t_k = 1$

while $F(x^k + t_k d^k_{NQM}) - F(x^k) \not\preceq \sigma t_k JF(x^k) d^k_{MQN}$ **do**

  Update $t_k := \gamma t_k$

end

return t_k

Next, we give the lower and upper bounds of stepsize along with d^k_{VM}.

Proposition 3.1 *Assume that F_i is L^k_i-smooth and μ^k_i-strongly convex relative to $\|\cdot\|_{B_k}$, $i \in [m]$. Then the stepsize along with d^k_{MQN} satisfies $\min\left\{1, \frac{2\gamma(1-\sigma)}{L^k_{\max}}\right\} \leq t_k \leq \min\left\{\frac{2(1-\sigma)}{\mu^k_{\max}}, 1\right\}$, where $L^k_{\max} := \max\{L^k_i : i \in [m]\}$, $\mu^k_{\max} := \max\{\mu^k_i : i \in \mathcal{A}_{MQN}(x^k)\}$.*

Proof The proof is similar to that in [7, Lemmas 3,4], we omit it here.

Remark 3.1 The stepsize along with d^k_{MQN} can be relatively small when $L^k_{\max}$ has a significant value, even if F_i is not ill-conditioned relative to $\|\cdot\|_{B_k}$ (a relatively small value of L^k_i/μ^k_i). This small stepsize hampers the local superlinear convergence of MQNMO and leads to inferior performance.

Remark 3.2 In [6], an aggregated line search was employed to achieve larger stepsizes, resulting in local superlinear convergence for MQNMO. However, it is essential to note that the aggregated line search cannot guarantee that all objective functions decrease in each iteration, and the global convergence of MQNMO with the aggregated line search approach remains unestablished.

3.4 Barzilai-Borwein descent method

As described in [7], imbalances among objective functions lead to small stepsize in SDMO, which decelerates the convergence. This is primarily due to equation (6), where the steepest descent direction results in a similar decrease in objective values for different objectives between two consecutive iterations. Observe the equation (11), imbalances among objective functions also decelerate the convergence of MQNMO. It is worth noting that Newton's direction achieves distinctive inner products for different objectives, as shown in (9). This explains why Newton-type methods accept larger stepsize and have the potential to alleviate imbalances among objective functions.

To achieve distinctive inner products between descent direction and gradients for a first-order method, Chen et al. [7] devised the Barzilai-Borwein descent direction, which is the optimal solution of the following subproblem:

$$\min_{d\in\mathbb{R}^n} \max_{i\in[m]} \left\{ \frac{\langle \nabla F_i(x^k), d\rangle}{\alpha_i(x^k)} + \frac{1}{2}\|d\|^2 \right\}, \tag{12}$$

where $\alpha(x^k) \in \mathbb{R}^m_{++}$ is given by Barzilai-Borwein method:

$$\alpha_i(x^k) = \begin{cases} \max\left\{\alpha_{\min}, \min\left\{\frac{\langle s^{k-1}, y_i^{k-1}\rangle}{\|s^{k-1}\|^2}, \alpha_{\max}\right\}\right\}, & \langle s^{k-1}, y_i^{k-1}\rangle > 0, \\ \max\left\{\alpha_{\min}, \min\left\{\frac{\|y_i^{k-1}\|}{\|s^{k-1}\|}, \alpha_{\max}\right\}\right\}, & \langle s^{k-1}, y_i^{k-1}\rangle < 0, \\ \alpha_{\min}, & \langle s^{k-1}, y_i^{k-1}\rangle = 0, \end{cases} \tag{13}$$

for all $i \in [m]$, where $\alpha_{\max}$ is a sufficient large positive constant and $\alpha_{\min}$ is a sufficient small positive constant, $s^{k-1} = x^k - x^{k-1},\ y_i^{k-1} = \nabla F_i(x^k) - \nabla F_i(x^{k-1}),\ i \in [m]$. In this case, the dual problem can be written as

$$-\min_{\lambda} \frac{1}{2}\left\|\sum_{i\in[m]} \frac{\lambda_i \nabla F_i(x^k)}{\alpha_i(x^k)}\right\|^2$$
$$\text{s.t. } \lambda \in \Delta_m.$$

Denote $\lambda^{BB}(x^k)$ an optimal solution of the dual problem. Similarly, we have

$$d_{BB}^k = -\sum_{i\in[m]} \frac{\lambda_i^{BB}(x^k)\nabla F_i(x^k)}{\alpha_i(x^k)},$$

and

$$\langle \nabla F_i(x^k), d_{BB}^k\rangle = -\alpha_i(x^k)\|d_{BB}^k\|^2 \text{ for all } i \in \mathcal{A}_{BB}(x^k). \tag{14}$$

It is evident that $\langle \nabla F_i(x^k), d_{BB}^k\rangle \neq \langle \nabla F_j(x^k), d_{BB}^k\rangle$ for all $i, j \in \mathcal{A}_{BB}(x^k)$ due to the objective-based $\alpha_i(x^k)$. We establish the following bounds for the stepsize along the Barzilai-Borwein descent direction.

Lemma 3.2 (See Proposition 2 of [7]) *Assume that F_i is L_i-smooth and μ_i-strongly convex for $i \in [m]$, and let $\sigma \leq \frac{1}{2}$ in line search. Then the stepsize along with d_{BB}^k satisfies* $\min\{1, \bar{t}_{\min}\} \leq t_k \leq 1$, *where* $\bar{t}_{\min} := \min\left\{\frac{2\gamma(1-\sigma)\mu_i}{L_i} : i \in [m]\right\}$.

Remark 3.3 The Barzilai-Borwein descent method for MOPs (BBDMO) can attain relatively large stepsizes as long as all objective functions are not ill-conditioned. Recently, Chen et al. [8] demonstrated that BBDMO can mitigate interference and imbalances among objectives, resulting in improved convergence rates compared to SDMO. However, it is essential to note that BBDMO remains sensitive to conditioning, as observed from a theoretical perspective [8].

4 Preconditioned Barzilai-Borwein method for MOPs

This section attempts to develop a method that enjoys cheap per-step cost and is not sensitive to imbalances and conditioning. Before presenting the method, let us summarize the characteristics of the methods discussed in the previous section.

Naturally, we aim to leverage the strengths of both MQNMO and BBDMO in the development of the descent direction:

$$d^k := \arg\min_{d\in\mathbb{R}^n} \max_{i\in[m]} \left\{\frac{\langle \nabla F_i(x^k), d\rangle}{\alpha_i^k} + \frac{1}{2}\|d\|_{B_k}^2\right\}, \tag{15}$$

Algorithm	**cheap** per-step cost	**not** sensitive to imbalances	**not** sensitive to conditioning
SDMO	✓	✗	✗
NMO	✗	✓	✓
MQNMO	✓	✗	✓
BBDMO	✓	✓	✗

Table 1: The characteristics of SDMO, NMO, VMMO, and BBDMO.

where $\alpha^k \succ 0$ mitigates the imbalances among objective functions, and $B_k \succ 0$ is applied to better capture the local geometry of the problem. We denote by $\theta(x^k)$ the optimal value of (15), hence

$$\theta(x^k) := \min_{d\in\mathbb{R}^n} \max_{i\in[m]} \left\{ \frac{\langle \nabla F_i(x^k), d\rangle}{\alpha_i^k} + \frac{1}{2}\|d\|^2_{B_k} \right\}.$$

Notably, the subproblem (15) can also be efficiently solved via its dual:

$$\begin{aligned} -\min_{\lambda} \ & \frac{1}{2}\left\| \sum_{i\in[m]} \frac{\lambda_i \nabla F_i(x^k)}{\alpha_i^k} \right\|^2_{B_k^{-1}} \\ \text{s.t. } & \lambda \in \Delta_m, \end{aligned} \tag{16}$$

provided that B_k^{-1} can be efficiently evaluated. Denote λ^k an optimal solution of the dual problem. It is evident that

$$d^k = -B_k^{-1}\left(\sum_{i\in[m]} \frac{\lambda_i^k \nabla F_i(x^k)}{\alpha_i^k} \right), \tag{17}$$

and

$$\langle \nabla F_i(x^k), d^k\rangle = -\alpha_i^k \|d^k\|^2_{B_k} \text{ for all } \lambda_i^k > 0. \tag{18}$$

Denote

$$\mathcal{D}_{\alpha^k}(x^k, d^k) := \max_{i\in[m]} \left\langle \frac{\nabla F_i(x^k)}{\alpha_i^k}, d^k \right\rangle,$$

It can be reformulated as

$$\mathcal{D}_{\alpha^k}(x^k, d^k) = -\left\|d^k\right\|^2_{B_k}. \tag{19}$$

Remark 4.1 Given that α_i^k is objective-based, equation (18) implies that different objective functions achieve distinct descent along d^k.

Next, we will present several properties of d^k.

Proposition 4.1 *Assume that $0 \le \alpha_l \le \alpha_i^k \le \alpha_u$, $aI \preceq B_k \preceq bI (a > 0)$ for all $k \ge 0,\ i \in [m]$. Let d^k be defined as (15), then the following statements hold.*

(i) *the following assertions are equivalent:*
  (a) *The point x^k is non-critical;*
  (b) $d^k \neq 0$;
  (c) *d^k is a descent direction.*

(ii) *If there exists a convergent subsequence $x^k \xrightarrow{\mathcal{K}} x^*$ such that $d^k \xrightarrow{\mathcal{K}} 0$, then x^* is Pareto critical.*

Proof (i) The assertions can be obtained by using the same arguments as in the proof of [31, Lemma 3.2].

(ii) By the definition of d^k, we have

$$
\begin{aligned}
-\frac{\left\|d^k\right\|_{B_k}^2}{2} &= \min_{d\in\mathbb{R}^n}\max_{i\in[m]}\left\{\frac{\langle\nabla F_i(x^k),d\rangle}{\alpha_i^k}+\frac{1}{2}\left\|d\right\|_{B_k}^2\right\} \\
&\leq \min_{d\in\mathbb{R}^n}\max_{i\in[m]}\left\{\frac{\langle\nabla F_i(x^k),d\rangle}{\alpha_u}+\frac{b}{2}\left\|d\right\|^2\right\} \\
&= \frac{1}{b\alpha_u^2}\min_{d\in\mathbb{R}^n}\max_{i\in[m]}\left\{\langle\nabla F_i(x^k),b\alpha_u d\rangle+\frac{1}{2}\left\|b\alpha_u d\right\|^2\right\} \\
&= \frac{1}{b\alpha_u^2}\min_{d\in\mathbb{R}^n}\max_{i\in[m]}\left\{\langle\nabla F_i(x^k),d\rangle+\frac{1}{2}\left\|d\right\|^2\right\} \\
&= -\frac{1}{2b\alpha_u^2}\left\|d_{SD}^k\right\|^2,
\end{aligned}
\tag{20}
$$

Since $d^k \xrightarrow{\mathcal{K}} 0$, it follows by the latter inequality and the uniformly positive definiteness of B_k that $d_{SD}^k \xrightarrow{\mathcal{K}} 0$. This, together with the continuity of $d_{SD}(\cdot)$ and the fact that $x^k \xrightarrow{\mathcal{K}} x^*$, implies $d_{SD}(x^*)=0$. Therefore, x^* is Pareto critical.

A first natural question that arises is how to select α^k and B_k to preserve the benefits of MQNMO and BBDMO.

4.1 Barzilai-Borwein method with variable metric

Firstly, we consider how to select an appropriate α^k to accelerate convergence. From the preconditioning perspective, the metric matrix B_k in (15) is selected to make a change of variable $\hat{x}=Px$ where $P=B_k^{1/2}$. By denoting $g_i(\hat{x})=F_i(x)=F_i(P^{-1}\hat{x})$, $i\in[m]$, in contrast to BBDMO, here we approximate the secant equation relative to g_i by setting

$$
\alpha_i^k=\frac{\left\langle\nabla g_i(\hat{x}^{k+1})-\nabla g_i(\hat{x}^k),\hat{x}^{k+1}-\hat{x}^k\right\rangle}{\|\hat{x}^{k+1}-\hat{x}^k\|^2}.
$$

Note that $\hat{x}=Px$ and $\nabla g_i(\hat{x})=P^{-1}\nabla F_i(x)$, the last equation can be rewritten as

$$
\alpha_i^k=\frac{\left\langle\nabla F_i(x^{k+1})-\nabla F_i(x^k),x^{k+1}-x^k\right\rangle}{\|x^{k+1}-x^k\|_{B_k}^2}.
$$

In general, we set $\alpha^k\in\mathbb{R}_{++}^m$ as follows:

$$
\alpha_i^k=\begin{cases}
\max\left\{\alpha_{\min},\min\left\{\dfrac{\langle s^{k-1},y_i^{k-1}\rangle}{\|s^{k-1}\|_{B_k}^2},\alpha_{\max}\right\}\right\}, & \langle s^{k-1},y_i^{k-1}\rangle>0,\\
\max\left\{\alpha_{\min},\min\left\{\dfrac{\|y_i^{k-1}\|}{\|B_k s^{k-1}\|},\alpha_{\max}\right\}\right\}, & \langle s^{k-1},y_i^{k-1}\rangle<0,\\
\alpha_{\min}, & \langle s^{k-1},y_i^{k-1}\rangle=0.
\end{cases}
\tag{21}
$$

Thus, the descent direction d^k is a preconditioned Barzilai-Borwein descent direction for MOPs.

We also give the lower and upper bounds of stepsize along with d^k.

Proposition 4.2 *Assume that F_i is L_i^k-smooth and μ_i^k-strongly convex relative to $\|\cdot\|_{B_k}$, $i \in [m]$, and let $\sigma \leq \frac{1}{2}$ in line search. Then the stepsize along with d^k satisfies* $\min\{1, t_{\min}\} \leq t_k \leq 1$, *where* $t_{\min} := \min\left\{\frac{2\gamma(1-\sigma)\mu_i^k}{L_i^k} : i \in [m]\right\}$.

Proof From the relative L_i^k-smoothness and μ_i^k-strong convexity of F_i, we derive that

$$\mu_i^k \leq \alpha_i^k \leq L_i^k.$$

Then, the lower and upper bounds can be obtained by the similar argument as presented in the proof of Proposition 3.1.

Remark 4.2 If F_i is not ill-conditioned relative to $\|\cdot\|_{B_k}$, then line search along with d^k can achieve a relatively large stepsize.

Up to now, we do not specify explicitly how we calculate B_k, then a generic preconditioned Barzilai-Borwein method for MOPs is described as follows.

Algorithm 2: `Generic_Preconditioned_Barzilai-Borwein_method_for_MOPs`

Data: $x^0 \in \mathbb{R}^n,\ B_0 \succ 0,\ \tau > 1$
Choose x^{-1} in a small neighborhood of x^0
for $k = 0, \ldots$ **do**
  Update α_i^k as (21), $i \in [m]$
  Compute λ^k a solution of (16)
  Update d^k as (17)
  if $\theta(x^k) = 0$ **then**
    return Pareto critical point x^k
  else
    repeat
      for $i = 1, \ldots, m$ **do**
        if $F_i(x^k + d^k) - F_i(x^k) > \langle \nabla F_i(x^k), d^k \rangle + \frac{\alpha_i^k}{2} \|d^k\|_{B_k}^2$ **then**
          Update $\alpha_i^k := \tau \alpha_i^k$
        end
      end
      Compute λ^k a solution of (16)
      Update d^k as (17)
    until $F_i(x^k + d^k) - F_i(x^k) \leq \langle \nabla F_i(x^k), d^k \rangle + \frac{\alpha_i^k}{2} \|d^k\|_{B_k}^2, \ \forall i \in [m]$;
    Update $x^{k+1} := x^k + d^k$
    Update $B_{k+1} \succ 0$
  end
end

Remark 4.3 In Algorithm 2, the *repeat loop* estimates the local smoothness parameter for $F_i,\ i \in [m]$. If F_i is L_i^k-smooth relative to $\|\cdot\|_{B_k}$, $i \in [m]$, then the *repeat loop* of Algorithm 2 terminates in a finite number of iterations, and $\alpha_i^k < \tau L_i^k,\ i \in [m]$.

4.2 Metric selection

In order to improve the gradient direction, the scale matrix B_k plays the role of a preconditioner, which should adjust the method to the right geometry of the problem. Naturally, a remaining question is: How do we choose B_k to enhance performance?

As already said, each iteration of the preconditioned Barzilai-Borwein descent method can be interpreted as the corresponding one of BBDMO for problem with transformed variable $\hat{x} = B_k^{1/2}x$. From the perspective of preconditioning, we estimate the performance of the generic algorithm without specifying explicitly how to select B_k. For simplicity, we fix the matrix $B_k = B$.

Theorem 4.1 *Assume that F_i is L_i^B-smooth and μ_i^B-strongly convex relative to $\|\cdot\|_B$, $i \in [m]$. Let $\{x^k\}$ be the sequence generated by Algorithm 2. Then, the following statements hold.*

(i) *$\{x^k\}$ converges to some Pareto solution x^*.*

(ii) $\|x^{k+1} - x^*\|_B \leq \sqrt{1 - \min_{i\in[m]} \left\{ \frac{\mu_i^B}{\tau L_i^B} \right\}} \|x^k - x^*\|_B$.

Proof The proof is analogous to that in [8, Theorem 4.6], we omit it here.

In the realm of SOPs, preconditioned methods work well with good metric matrix for which the transformed problem has moderate condition number. A powerful approach is to set $B_k = \nabla^2 F(x^k)$, which corresponds to Newton's method. Similarly, the results in Theorem 4.1 also suggest that we choose an appropriate B_k to reduce the largest condition number. However, especially when the Hessian matrices are different, we cannot use a single matrix to simultaneously approximate all the Hessian matrices for MOPs. The following example shows that the largest condition number for multi-objective problems cannot be reduced by preconditioning.

Example 4.1 Consider the multiobjective optimization problem

$$\min_x (x_1^2 + 1000x_2^2, 1000x_1^2 + x_2^2)^T.$$

By direct calculation, we have

$$\nabla^2 f_1(x) = \begin{pmatrix} 2 & 0 \\ 0 & 2000 \end{pmatrix}, \ \nabla^2 f_2(x) = \begin{pmatrix} 2000 & 0 \\ 0 & 2 \end{pmatrix}.$$

Then $\kappa_1 = \kappa_2 = 1000$. For any positive definite matrix

$$B = \begin{pmatrix} B_{11} & B_{21} \\ B_{12} & B_{22} \end{pmatrix},$$

there exist $M_1, M_2, m_1, m_2 > 0$ such that $\hat{\kappa}_1 = M_1/m_1$, $\hat{\kappa}_2 = M_2/m_2$, $m_1 B \preceq \nabla^2 f_1(x) \preceq M_1 B$ and $m_2 B \preceq \nabla^2 f_2(x) \preceq M_2 B$. Multiplied by eigenvectors $(1,0)^T$ and $(0,1)^T$, we have the following equations

$$\begin{cases} m_1 B_{11} \leq 2 \leq M_1 B_{11}, \\ m_2 B_{11} \leq 2000 \leq M_2 B_{11}, \\ m_1 B_{22} \leq 2000 \leq M_1 B_{22}, \\ m_2 B_{22} \leq 2 \leq M_2 B_{22}. \end{cases}$$

The equations can be reformulated as

$$\begin{cases} m_1 \leq 2/B_{11} \leq M_1, \\ m_2 \leq 2000/B_{11} \leq M_2, \\ m_1 \leq 2000/B_{22} \leq M_1, \\ m_2 \leq 2/B_{22} \leq M_2. \end{cases}$$

We use the equations to obtain

$$\hat{\kappa}_1 = \frac{M_1}{m_1} \geq \frac{\max\{2/B_{11}, 2000/B_{22}\}}{\min\{2/B_{11}, 2000/B_{22}\}}, \ \hat{\kappa}_2 = \frac{M_2}{m_2} \geq \frac{\max\{2000/B_{11}, 2/B_{22}\}}{\min\{2000/B_{11}, 2/B_{22}\}}.$$

By distinguishing three cases, it is easy to verify that $\max\{\hat{\kappa}_1, \hat{\kappa}_2\} \geq 1000 = \max\{\kappa_1, \kappa_2\}$, and the equality can hold when $B_{11} = B_{22}$.

Remark 4.4 The vanilla gradient direction is optimal when the contour of objective function is a hyper-sphere. For general functions, the efficiency of the vanilla gradient direction in the transformed space can be improved by appropriate scale matrix to make the transformed contour closer to a hyper-sphere, which is what the preconditioning actually does. However, if the contours of the objectives are distinct hyper-ellipsoids in MOPs, there is no preconditioner that simultaneously makes the transformed contours closer to hyper-spheres.

Based on the above discussion, can we claim that preconditioning plays no role in multiobjective optimization? It is worth noting that the worst-case convergence guarantees for gradient methods obtained by considering only the largest and smallest eigenvalues of the Hessian may be too pessimistic. In what follows, we show that one possible way to speed up convergence is to approximate the aggregated Hessian rather than any single one.

From a scalarization perspective, SDMO can be interpreted as an implicit gradient descent method with adaptive scalarization (where the weight vector is an optimal solution of the dual problem). Similarly, NMO can also be interpreted as an implicit Newton method with adaptive scalarization:

$$-\left(\sum_{i\in[m]} \lambda_i^N(x^k)\nabla^2 F_i(x^k)\right)^{-1} \left(\sum_{i\in[m]} \lambda_i^N(x^k)\nabla F_i(x^k)\right).$$

Recall that the Barzilai-Borwein descent direction with variable metric is

$$-B_k^{-1}\left(\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}\nabla F_i(x^k)\right).$$

Intuitively, a judicious choice for B_k is to approximate the variable aggregated Hessian, i.e.,

$$B_k \approx \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}\nabla F_i^2(x^k). \tag{22}$$

In the following, we attempt to confirm the efficiency of the metric selection from a theoretical perspective.

Theorem 4.2 *Assume that F_i is L_i^k-smooth ($L_i^k \leq L$) and μ_i^k-strongly convex ($\mu_i^k \geq \mu > 0$) relative to $\|\cdot\|_{B_k}$, $i \in [m]$, $aI \preceq B_k \preceq bI (a > 0)$. Let $\{x^k\}$ be the sequence generated by Algorithm 2. Then, the following statements hold.*

(i) $\{x^k\}$ *converges to some Pareto solution* x^*.

(ii) $\|x^{k+1} - x^*\|_{B_k} \leq \sqrt{1 - \frac{\|x^k - x^*\|^2_{\sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(\bar{x}^k)}{\alpha_i^k}}}{\|x^k - x^*\|^2_{B_k}}}\, \|x^k - x^*\|_{B_k}$.

Proof (i) By the relatively strong convexity of F_i and the uniformly positive definiteness of B_k, we conclude that F_i is strongly convex. Using the line search condition, we have

$$F_i(x^{k+1}) - F_i(x^k) \leq \left\langle \nabla F_i(x^k), d^k \right\rangle + \frac{\alpha_i^k}{2} \left\|d^k\right\|^2_{B_k} \leq -\frac{\alpha_i^k}{2} \left\|d^k\right\|^2_{B_k} \leq -\frac{a\mu}{2} \left\|d^k\right\|^2. \tag{23}$$

It follows that $\{F(x^k)\}$ is monotone decreasing, which, together with the strong convexity of F_i, implies the compactness of $\{x : F(x) \preceq F(x^0)\}$ and there exists F^* such that $F^* \preceq F(x^k)$. Taking the sum of the inequality (23), we obtain

$$\sum_{k\geq 0} \frac{a\mu}{2} \left\|d^k\right\|^2 \leq \sum_{k\geq 0} (F_i(x^k) - F_i(x^{k+1})) \leq F_i(x^0) - F_i^* < \infty.$$

Therefore, $d^k \to 0$. On the other hand, since $\{x^k\}$ is a subset of the compact $\{x : F(x) \preceq F(x^0)\}$, there exists an infinite index set K such that $x^k \xrightarrow{K} x^*$. This, together with $d^k \to 0$ and Proposition 4.1(ii), implies that x^* is a Pareto critical point. Since F is strongly convex, we conclude that x^* is a Pareto solution, and the latter limit is enough to prove that $\{x^k\}$ converges to x^*.

(ii) Given the twice continuity of F_i, we use Newton-Leibniz formula to get

$$F_i(b) - F_i(a) = \left\langle \int_0^1 \nabla F_i(a + t(b - a))dt, b - a \right\rangle. \tag{24}$$

Again using the Newton-Leibniz formula for the average gradient, we have

$$\int_0^1 (\nabla F_i(a + t(b - a)) - \nabla F_i(a))dt = \int_0^1 \int_0^1 \nabla^2 F_i(a + st(b - a))ds(t(b - a))dt.$$

Plugging this into (24) gives

$$F_i(b) - F_i(a) = \langle \nabla F_i(a), b - a\rangle + \left\langle b - a, \int_0^1 \int_0^1 \nabla^2 F_i(a + st(b - a))ds(t(b - a))dt \right\rangle. \tag{25}$$

By substituting $b = x,\ a = x^k$ into (25), we have

$$\begin{aligned}
&F_i(x^{k+1}) - F_i(x) \\
&= (F_i(x^{k+1}) - F_i(x^k)) + (F_i(x^k) - F_i(x)) \\
&\leq \left(\left\langle \nabla F_i(x^k), x^{k+1} - x^k \right\rangle + \frac{\alpha_i^k}{2} \|x^{k+1} - x^k\|^2_{B_k} \right) \\
&\quad + \left\langle \nabla F_i(x^k), x^k - x \right\rangle - \left\langle x - x^k, \int_0^1 \int_0^1 \nabla^2 F_i(x^k + st(x - x^k))ds(t(x - x^k))dt \right\rangle \\
&= \left\langle \nabla F_i(x^k), x^{k+1} - x \right\rangle + \frac{\alpha_i^k}{2} \|x^{k+1} - x^k\|^2_{B_k} \\
&\quad - \left\langle x - x^k, \int_0^1 \int_0^1 \nabla^2 F_i(x^k + st(x - x^k))ds(t(x - x^k))dt \right\rangle.
\end{aligned} \tag{26}$$

It follows that

$$\begin{aligned}
&\sum_{i\in[m]} \lambda_i^k \frac{F_i(x^{k+1}) - F_i(x)}{\alpha_i^k} \\
&\leq \left\langle \sum_{i\in[m]} \lambda_i^k \frac{\nabla F_i(x^k)}{\alpha_i^k}, x^{k+1} - x \right\rangle + \frac{1}{2}\|x^{k+1} - x^k\|_{B_k}^2 \\
&\quad - \left\langle x - x^k, \int_0^1 \int_0^1 \sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(x^k + st(x - x^k))}{\alpha_i^k} ds(t(x - x^k))dt \right\rangle \\
&= \langle B_k(x^k - x^{k+1}), x^{k+1} - x\rangle + \frac{1}{2}\|x^{k+1} - x^k\|_{B_k}^2 \\
&\quad - \left\langle x - x^k, \int_0^1 \int_0^1 \sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(x^k + st(x - x^k))}{\alpha_i^k} ds(t(x - x^k))dt \right\rangle \\
&= \left(\frac{1}{2}\|x^k - x\|_{B_k}^2 - \frac{1}{2}\|x^{k+1} - x\|_{B_k}^2 - \frac{1}{2}\|x^{k+1} - x^k\|_{B_k}^2\right) + \frac{1}{2}\|x^{k+1} - x^k\|_{B_k}^2 \\
&\quad - \left\langle x - x^k, \int_0^1 \int_0^1 \sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(x^k + st(x - x^k))}{\alpha_i^k} ds(t(x - x^k))dt \right\rangle \\
&= \frac{1}{2}\|x^k - x\|_{B_k}^2 - \frac{1}{2}\|x^{k+1} - x\|_{B_k}^2 \\
&\quad - \left\langle x - x^k, \int_0^1 \int_0^1 \sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(x^k + st(x - x^k))}{\alpha_i^k} ds(t(x - x^k))dt \right\rangle.
\end{aligned}$$

By substituting $x = x^*$, there exists $\bar{x}^k \in [x^k, x^*]$ (line segment between x^k and x^*) such that

$$0 \leq \sum_{i\in[m]} \lambda_i^k \frac{F_i(x^{k+1}) - F_i(x^*)}{\alpha_i^k} \leq \frac{1}{2}\|x^k - x^*\|_{B_k}^2 - \frac{1}{2}\|x^{k+1} - x^*\|_{B_k}^2 - \frac{1}{2}\|x^k - x^*\|^2_{\sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(\bar{x}^k)}{\alpha_i^k}}.$$

Hence,

$$\|x^{k+1} - x^*\|_{B_k} \leq \sqrt{1 - \frac{\|x^k - x^*\|^2_{\sum_{i\in[m]} \lambda_i^k \frac{\nabla^2 F_i(\bar{x}^k)}{\alpha_i^k}}}{\|x^k - x^*\|_{B_k}^2}}\|x^k - x^*\|_{B_k}.$$

This completes the proof.

Remark 4.5 Instead of the lower quadratic bound with strong convexity, we use the exact Newton-Leibniz formulation in (26), which is more accurate in most cases. Although Theorem 4.2(ii) cannot be treated as a basis for linear convergence directly, it confirms that the algorithm enjoys a fast global convergence with appropriate preconditioning.

Interestingly, Theorem 4.2(ii) also shows that optimizing multiple objective functions simultaneously can be easier than optimizing any single one.

Remark 4.6 As a by-product, Theorem 4.2(ii) also shows that BBDMO enjoys fast linear convergence in a region of the Pareto front. For the BBDMO, we have $B_k = I$. Assume that

$\mu_i \preceq \nabla^2 F_i(x) \preceq L_i,\ i \in [m]$, we use Theorem 4.2(ii) to get

$$\begin{aligned}
\|x^{k+1}-x^*\| &\leq \sqrt{1-\frac{\|x^k-x^*\|^2_{\sum\limits_{i\in[m]}\lambda_i^k\frac{\nabla^2 F_i(\bar{x}^k)}{\alpha_i^k}}}{\|x^k-x^*\|^2}}\|x^k-x^*\| \\
&\leq \sqrt{1-\frac{\|x^k-x^*\|^2_{\sum\limits_{i\in[m]}\lambda_i^k\frac{\nabla^2 F_i(\bar{x}^k)}{L_{\max}}}}{\|x^k-x^*\|^2}}\|x^k-x^*\| \\
&\leq \sqrt{1-\lambda_{\min}(\sum_{i\in[m]}\lambda_i^k\nabla^2 F_i(\bar{x}^k))/L_{\max}}\|x^k-x^*\|.
\end{aligned}$$

Interestingly, even if each of the objective functions is ill-conditioned, the following example shows that BBDMO can enjoy fast linear convergence. Consider the problem in Example 4.1, by simple calculation, for any x^k, we have

$$\begin{aligned}
\left\|x^{k+1}-x^*\right\| &\leq \sqrt{1-\frac{\min\{1000\lambda_1^k+\lambda_2^k,\lambda_1^k+1000\lambda_2^k\}}{1000}}\left\|x^k-x^*\right\| \\
&= \sqrt{1-\frac{\min\{999\lambda_1^k+1,1000-999\lambda_1^k\}}{1000}}\left\|x^k-x^*\right\|.
\end{aligned}$$

Then, in the region with $\lambda_1^k \neq 0, 1$, BBDMO enjoys fast linear convergence.

For a generic case, we further explain the choice of B_k by the following asymptotic convergence result.

Theorem 4.3 *Let $x^{k+1} = x^k + t_k d^k$, d^k be denoted as (15), suppose that the following assumptions hold:*

(a) $\{x^k\}$ *converges to some Pareto solution* x^* *and* $F(x^*) \preceq F(x^k)$*,*
(b) $t_k = 1$ *for sufficiently large* k*,*
(c) $\left\{\sum_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\right\}$ *is bounded,*
(d) $aI \preceq \lim\limits_{k\to\infty}\sum\limits_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(x^*) \preceq bI,\ (a>0)$*,*
(e) $\lim\limits_{k\to\infty}\nabla^2 F_i(x^k) = \nabla^2 F_i(x^*)$ *for all* $i \in [m]$*,*
(f) $\lim\limits_{k\to\infty}\frac{\left\|(B_k - \sum\limits_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(x^k))s_k\right\|}{\|s_k\|} = 0.$

Then, $\{x^k\}$ *converges to* x^* *superlinearly.*

Proof By substituting $b = x^{k+1},\ a = x^k$ and $b = x^*,\ a = x^k$ into (25), respectively, we have

$$\begin{aligned}
0 &\le F_i(x^{k+1}) - F_i(x^*) \\
&= (F_i(x^{k+1}) - F_i(x^k)) - (F_i(x^*) - F_i(x^k)) \\
&= \left\langle \nabla F_i(x^k), x^{k+1} - x^k \right\rangle + \left\langle x^{k+1} - x^k, \int_0^1 \int_0^1 \nabla^2 F_i(x^k + st(x^{k+1} - x^k)) ds (t(x^{k+1} - x^k)) dt \right\rangle \\
&\quad + \left\langle \nabla F_i(x^k), x^k - x^* \right\rangle - \left\langle x^* - x^k, \int_0^1 \int_0^1 \nabla^2 F_i(x^k + st(x^* - x^k)) ds (t(x^* - x^k)) dt \right\rangle \\
&= \left\langle \nabla F_i(x^k), x^{k+1} - x^* \right\rangle + \left\langle x^{k+1} - x^k, \int_0^1 \int_0^1 \nabla^2 F_i(x^k + st(x^{k+1} - x^k)) ds (t(x^{k+1} - x^k)) dt \right\rangle \\
&\quad - \left\langle x^* - x^k, \int_0^1 \int_0^1 \nabla^2 F_i(x^k + st(x^* - x^k)) ds (t(x^* - x^k)) dt \right\rangle .
\end{aligned}$$

Then there exist $\bar{x}_1^k \in [x^k, x^{k+1}]$ (line segment between x^k and x^{k+1}) and $\bar{x}_2^k \in [x^k, x^*]$ such that

$$\begin{aligned}
0 &\le \left\langle \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla F_i(x^k), x^{k+1} - x^* \right\rangle + \frac{1}{2} \left\| x^{k+1} - x^k \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_1^k)} - \frac{1}{2} \left\| x^k - x^* \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k)} \\
&= \left\langle -B_k d^k, x^{k+1} - x^* \right\rangle + \frac{1}{2} \left\| x^{k+1} - x^k \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_1^k)} - \frac{1}{2} \left\| x^k - x^* \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k)} \\
&= \left\langle B_k(x^k - x^{k+1}), x^{k+1} - x^* \right\rangle + \frac{1}{2} \left\| x^{k+1} - x^k \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_1^k)} - \frac{1}{2} \left\| x^k - x^* \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k)} \\
&= \left\langle B_k(x^k - x^{k+1}), x^{k+1} - x^* \right\rangle + \frac{1}{2} \left\| x^{k+1} - x^k \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_1^k)} - \frac{1}{2} \left\| x^k - x^{k+1} + x^{k+1} - x^* \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k)} \\
&= \left\langle \left(B_k - \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k) \right) (x^k - x^{k+1}), x^{k+1} - x^* \right\rangle + \frac{1}{2} \left\| x^{k+1} - x^k \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_1^k)} \\
&\quad - \frac{1}{2} \left\| x^{k+1} - x^k \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k)} - \frac{1}{2} \left\| x^{k+1} - x^* \right\|^2_{\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k)} ,
\end{aligned}$$

where the second equality follows by $t_k = 1$ for sufficient large k. Without loss of generality, for any $\epsilon > 0$, there exists k_ϵ such that, for all $k \ge k_\epsilon$ and $j \in \{1, 2\}$,

$$\left\| \left(\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_j^k) - B_k \right) s_k \right\| \le \epsilon \|s_k\|, \tag{27}$$

and

$$\left\| \left(\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_1^k) - \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla^2 F_i(\bar{x}_2^k) \right) s_k \right\| \le \epsilon \|s_k\|, \tag{28}$$

where (27) is given by assumptions (c), (e) and (f), (28) follows by (c) and (e). Then, we use relations (27) and (28) to get

$$\begin{aligned}
&\left\|x^{k+1}-x^*\right\|^2_{\sum\limits_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(\bar{x}_2^k)}\\
&\leq 2\left\langle\left(B_k-\sum_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(\bar{x}_2^k)\right)(x^k-x^{k+1}),x^{k+1}-x^*\right\rangle\\
&\quad+\left\langle\left(\sum_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(\bar{x}_1^k)-\sum_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(\bar{x}_2^k)\right)(x^{k+1}-x^k),x^{k+1}-x^k\right\rangle\\
&\leq 2\epsilon\left\|s_k\right\|\left\|x^{k+1}-x^*\right\|+\epsilon\left\|s_k\right\|^2.
\end{aligned}\tag{29}$$

On the other hand, by assumptions (c), (d) and (e), we have

$$\left\|x^{k+1}-x^*\right\|^2_{\sum\limits_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2 F_i(\bar{x}_2^k)}\geq a\left\|x^{k+1}-x^*\right\|^2.$$

Rearranging and substituting the above relation into (29), we obtain

$$a\left\|x^{k+1}-x^*\right\|^2-2\epsilon\left\|x^{k+1}-x^*\right\|\left\|s_k\right\|-\epsilon\left\|s_k\right\|^2\leq 0.$$

Dividing by $\left\|s_k\right\|^2$, it is easy to get

$$\frac{\left\|x^{k+1}-x^*\right\|}{\left\|s_k\right\|}\in\left[\frac{\epsilon-\sqrt{\epsilon^2+a\epsilon}}{a},\frac{\epsilon+\sqrt{\epsilon^2+a\epsilon}}{a}\right].$$

By the arbitrary of $\epsilon>0$, it follows that

$$\lim_{k\to\infty}\frac{\left\|x^{k+1}-x^*\right\|}{\left\|s_k\right\|}=0.$$

Notice that $\left\|s_k\right\|\leq\left\|x^{k+1}-x^*\right\|+\left\|x^k-x^*\right\|$, then

$$0\leq\lim_{k\to\infty}\frac{\left\|x^{k+1}-x^*\right\|}{\left\|x^{k+1}-x^*\right\|+\left\|x^k-x^*\right\|}\leq\lim_{k\to\infty}\frac{\left\|x^{k+1}-x^*\right\|}{\left\|s_k\right\|}=0.$$

Therefore,

$$\lim_{k\to\infty}\frac{\left\|x^{k+1}-x^*\right\|}{\left\|x^k-x^*\right\|}=0,$$

and hence the rate of convergence is superlinear.

5 Barzilai-Borwein quasi-Newton method for MOPs

A specific approach to achieving the idea in (22) is explored in this section.

5.1 Trade-off of quasi-Newton approximation

In general $\frac{\lambda^k}{\alpha^k} \notin \Delta_m$, but there exists a $\bar{\lambda}^k := \frac{\lambda^k}{\alpha^k}/(\sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}) \in \Delta_m$ such that

$$d^k \approx -\left(\sum_{i\in[m]} \bar{\lambda}_i^k \nabla^2 F_i(x^k)\right)^{-1} \left(\sum_{i\in[m]} \bar{\lambda}_i^k \nabla F_i(x^k)\right).$$

As a result, the selected B_k can be perceived as the trade-off among Hessian matrices, with the weight vector being adaptively updated in each iteration. Unfortunately, λ^k is unavailable before computing the subproblem, and α^k is determined using B_k. As an alternative, we can replace $\frac{\lambda^k}{\alpha^k}$ with $\frac{\lambda^{k-1}}{\alpha^{k-1}}$. However, two remaining shortcomings exist with $B_k = \sum_{i\in[m]} \frac{\lambda_i^{k-1}}{\alpha_i^{k-1}} \nabla^2 F_i(x^k)$: Hessian matrices are not readily available, and obtaining the inverse of B_k is computationally expensive. To address the issues, we attempt to update B_k by using BFGS method. In terms of (16) and (17), we directly update

$$B_{k+1}^{-1} = \left(I - \rho_k s_k y_k^T\right) B_k^{-1} \left(I - \rho_k y_k s_k^T\right) + \rho_k s_k s_k^T, \tag{30}$$

where $s_k = x^{k+1} - x^k$, $y_k = \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}(\nabla F_i(x^{k+1}) - \nabla F_i(x^k))$, and

$$\rho_k = \begin{cases} 1/\langle s_k, y_k\rangle, & \langle s_k, y_k\rangle > 0, \\ 1/(\mathcal{D}_{\alpha^k}(x^{k+1}, s_k) - \sum_{i\in[m]} \lambda_i^k \langle \nabla F_i(x^k)/\alpha_i^k, s_k\rangle), & \text{otherwise}, \end{cases} \tag{31}$$

By taking the inverse of (30), we obtain the update formula for B_{k+1}:

$$\begin{aligned} B_{k+1} = B_k &- \frac{\rho_k^{-1} B_k s_k s_k^T B_k}{(\rho_k^{-1} - s_k^T y_k)^2 + \rho_k^{-1} s_k^T B_k s_k} \\ &+ \frac{s_k^T B_k s_k y_k y_k^T}{(\rho_k^{-1} - s_k^T y_k)^2 + \rho_k^{-1} s_k^T B_k s_k} \\ &+ (\rho_k^{-1} - s_k^T y_k) \frac{y_k s_k^T B_k + B_k s_k y_k^T}{(\rho_k^{-1} - s_k^T y_k)^2 + \rho_k^{-1} s_k^T B_k s_k} \end{aligned} \tag{32}$$

Remark 5.1 The BFGS-type updating (30) and (32) are similar to (15) and (17) in [33].

The Barzilai-Borwein BFGS method for MOPs is described as follows:

Remark 5.2 In SOPs, when Dennis-Moré condition holds, the Barzilai-Borwein parameter

$$\begin{aligned} \lim_{k\to\infty} \alpha^k &= \lim_{k\to\infty} \frac{\langle \nabla F(x^k + s^k) - \nabla F(x^k), s^k\rangle}{s_k^T B_k s_k} \\ &= \lim_{k\to\infty} \frac{s_k^T \nabla^2 F(x^k) s_k}{s_k^T B_k s_k} = 1. \end{aligned}$$

From a preconditioning perspective, if the preconditioner B_k is not good enough, the Barzilai-Borwein method can be seen as diagonal preconditioning based on the metric $\|\cdot\|_{B_k}$. The Dennis-Moré condition states that B_k is good enough as a preconditioner if x^k belongs to a small neighborhood of the optimal solution. For this reason, the two approaches are rarely used

Algorithm 3: `Barzilai-Borwein_quasi-Newton_method_for_MOPs`

Data: $x^0 \in \mathbb{R}^n,\ 0 \prec B_0,\ 0 < \sigma_1 \leq \sigma_2 < 1$

Choose x^{-1} in a small neighborhood of x^0

for $k = 0, \ldots$ **do**

  Update α_i^k as (21), $i \in [m]$

  Compute λ^k a solution of (16)

  Update d^k as (17)

  if $\theta(x^k) = 0$ **then**

    return Pareto critical point x^k

  else

    Compute a stepsize t_k that satisfies

$$(F_i(x^k + td^k) - F_i(x^k))/\alpha_i^k \leq \sigma_1 t \mathcal{D}_{\alpha^k}(x^k, d^k),\ \forall i \in [m], \tag{33}$$

$$\mathcal{D}_{\alpha^k}(x^k + td^k, d^k) \geq \sigma_2 \mathcal{D}_{\alpha^k}(x^k, d^k). \tag{34}$$

    Update $x^{k+1} := x^k + t_k d^k$

    Update B_{k+1}^{-1} as (30)

    Update B_{k+1} as (32)

  end

end

simultaneously in SOPs. In MOPs, as we described in Remark 4.4, there is no preconditioner that simultaneously captures the correct geometry for different objective functions, so the Barzilai-Borwein parameters (21) serve as preconditioners for different objective functions.

To ensure the Algorithm 3 is well-defined, we require the following assumption.

Assumption 1 *For any $z \in \mathbb{R}^n$, the level set $\mathcal{L}_F(z) = \{x : \ F(x) \preceq F(z)\}$ is bounded.*

Proposition 5.1 *Suppose that Assumption 1 holds and that d^k is a descent direction, $0 < \sigma_1 \leq \sigma_2 < 1$. Then, there exists an interval $[t_l, t_u]$, with $0 < t_l < t_u < 1$, such that for all $t \in [t_l, t_u]$ equations (33) and (34) hold.*

Proof The proof is similar to that in [22, Proposition 2], we omit it here.

The following result shows that Algorithm 3 is well-defined without convexity assumption.

Proposition 5.2 *If the stepsize t_k is obtained by Wolfe line search (33) and (34), then ρ_k in (31) is positive. Moreover, the metric matrix B_k is positive definite.*

Proof The assertions are obvious, we omit the proof here.

5.2 Global convergence

This section presents the global convergence results for Algorithm 3. Notably, Algorithm 3 terminates with a Pareto critical point in a finite number of iterations or generates an infinite sequence of noncritical points. In the sequel, we will assume that Algorithm 3 produces an infinite sequence of noncritical points. Before presenting the global convergence of Algorithm 3, we state the following convexity assumption, which is also required in [22, 33].

Assumption 2 *(i) $\nabla^2 F_i$ is Lipschitz continuous with constant $L_i > 0$ for $i \in [m]$. (ii) The level set $\mathcal{L}_F(x^0) = \{x : \ F(x) \preceq F(x^0)\}$ is convex and there exist constants $a, b > 0$, $i \in [m]$, such that*

$$a \|z\|^2 \leq \left\langle z, \nabla^2 F_i(x) z \right\rangle \leq b \|z\|^2 , \ \forall i \in [m], \tag{35}$$

for all $z \in \mathbb{R}^n$ and $x \in \mathcal{L}_F(x^0)$.

Under the Assumption 2, $\langle s_k, y_k \rangle > 0$ and hence $\rho_k = 1/\langle s_k, y_k \rangle$ for all $k \geq 0$. In this case, we have

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

We denote

$$\cos \beta^k = \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|},$$

so that β^k the angel between s_k and $B_k s_k$. The following lemma shows that a subsequence of $\{\cos \beta^k\}$ is uniformly bounded away from 0 .

Lemma 5.1 *Suppose that Assumption 2 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3. Then, there exist a constant $\delta > 0$ and a subsequence of indices $\mathcal{K}$ such that*

$$\cos \beta^k \geq \delta, \ \forall k \in \mathcal{K}. \tag{36}$$

Proof The proof is similar to that in [29, Theorem 8.5], we omit it here.

Before presenting the global convergence of Algorithm 3, we establish the following bound.

Lemma 5.2 *Suppose that Assumption 2 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3. Then, for all $k \geq 0$,*

$$\mathcal{D}_{\alpha^k}(x^k, d^k) \leq -\frac{\cos \beta^k}{\alpha_{\max}} \left\|d^k\right\| \left\|d^k_{SD}\right\|. \tag{37}$$

Proof By the definition of $\cos \beta^k$, we have

$$\cos \beta^k = \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|} = \frac{d_k^T B_k d_k}{\|d_k\| \|B_k d_k\|}.$$

This, together with relation (19), gives

$$\begin{aligned}
\mathcal{D}_{\alpha^k}(x^k, d^k) &= -\left\|d^k\right\|^2_{B_k} \\
&= -\cos \beta^k \left\|d^k\right\| \left\|B_k d^k\right\| \\
&= -\cos \beta^k \left\|d^k\right\| \left\| \sum_{i \in [m]} \frac{\lambda_i^k}{\alpha_i^k} \nabla F_i(x^k) \right\| \\
&= -\cos \beta^k \left\|d^k\right\| \left(\sum_{i \in [m]} \frac{\lambda_i^k}{\alpha_i^k} \right) \left\| \sum_{i \in [m]} \bar{\lambda}_i^k \nabla F_i(x^k) \right\| \\
&\leq -\left(\sum_{i \in [m]} \frac{\lambda_i^k}{\alpha_i^k} \right) \cos \beta^k \left\|d^k\right\| \left\|d^k_{SD}\right\| \\
&\leq -\frac{\cos \beta^k}{\alpha_{\max}} \left\|d^k\right\| \left\|d^k_{SD}\right\|,
\end{aligned} \tag{38}$$

where $\bar{\lambda}_i^k = \frac{\lambda_i^k}{\alpha_i^k}/(\sum_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k})$, the first inequality follows by $\bar{\lambda}^k \in \Delta_m$ and the definition of d_{SD}^k, and the last inequality is given by $\bar{\lambda}^k \in \Delta_m$ and $\alpha_i^k \leq \alpha_{\max},\ i \in [m]$.

We are now in a position to establish a global convergence for the Algorithm 3.

Theorem 5.1 *Suppose that Assumption 2 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3. Then, $\{x^k\}$ converges to a Pareto solution x^* of (MOP).*

Proof By (34) and Assumption 2, we have

$$\begin{aligned}(\sigma_2 - 1)\mathcal{D}_{\alpha^k}(x^k, d^k) &\leq \mathcal{D}_{\alpha^k}(x^k + t_k d^k, d^k) - \mathcal{D}_{\alpha^k}(x^k, d^k)\\ &\leq \max_{i\in[m]}\left\langle \frac{\nabla F_i(x^k + t_k d^k) - \nabla F_i(x^k)}{\alpha_i^k}, d^k \right\rangle\\ &\leq \frac{b}{\alpha_{\min}} t_k \left\|d^k\right\|^2.\end{aligned}$$

Since $\mathcal{D}_{\alpha^k}(x^k, d^k) < 0$ and $\left\|d^k\right\| \neq 0$, we obtain

$$\frac{\mathcal{D}^2_{\alpha^k}(x^k, d^k)}{\left\|d^k\right\|^2} \leq t_k \frac{\mathcal{D}_{\alpha^k}(x^k, d^k)}{\sigma_2 - 1}. \tag{39}$$

We use the relation (33) to deduce that $\{F_i(x^k)\}$ is monotone decreasing and that

$$F_i(x^{k+1}) - F_i(x^k) \leq \alpha_{\min}\sigma_1 t_k \mathcal{D}_{\alpha^k}(x^k, d^k).$$

It follows that $\{x^k\} \subset \mathcal{L}_F(x^0)$. Observe the strong convexity of F_i on $\mathcal{L}_F(x^0)$, then $\mathcal{L}_F(x^0)$ is compact, and there exists F_i^* such that

$$\sum_{k\geq 0} -\alpha_{\min}\sigma_1 t_k \mathcal{D}_{\alpha^k}(x^k, d^k) \leq F_i(x^0) - F_i^* < \infty.$$

Plugging the bound into (39) gives

$$\sum_{k\geq 0} \frac{\mathcal{D}^2_{\alpha^k}(x^k, d^k)}{\left\|d^k\right\|^2} < \infty. \tag{40}$$

By substituting (36) and (37) into (40), we have

$$\sum_{k\in\mathcal{K}} \frac{\delta^2}{\alpha_{\max}^2}\left\|d_{SD}^k\right\|^2 < \infty,$$

and hence

$$\lim_{k\in\mathcal{K}} d_{SD}^k = 0. \tag{41}$$

Observe the compactness of $\mathcal{L}_F(x^0)$, then there exists an infinite index set $\mathcal{K}' \subset \mathcal{K}$ and a point x^* such that $\lim_{k\in\mathcal{K}'} x^k = x^*$. This, together with (41) and the continuity of $d_{SD}(\cdot)$, implies that x^* is a Pareto critical point. Since F is strongly convex on $\mathcal{L}_F(x^0)$, we conclude that x^* is a Pareto solution, and the latter limit is enough to prove that $\{x^k\}$ converges to x^*.

In the remainder of this subsection, we investigate the linear convergence of BBQNMO. Before presenting the main result, we start with the following technique result.

Lemma 5.3 *Suppose that Assumption 2 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3 and x^* be as in Theorem 5.1. Then, for all $k \geq 0$, the following statements hold.*

(i) $\left\|x^k - x^*\right\| \leq \frac{2}{a}\left\|d^k_{SD}\right\|$.
(ii) $\left\|s^k\right\| \geq \frac{(1-\sigma_2)\alpha_{\min}}{b\alpha_{\max}} \cos\beta^k \left\|d^k_{SD}\right\|$.

Proof (i) The assertion can be obtained by using the same arguments as in the proof of [33, Lemma 4.5]

(ii) By Assumption 2, the mean value theorem gives

$$\left\|F_i(x^{k+1}) - \nabla F_i(x^k)\right\| = \left\|\nabla^2 F_i(\bar{x}^k)(x^{k+1} - x^k)\right\| \leq b\left\|s^k\right\|,$$

where $\bar{x}^k \in [x^k, x^{k+1}]$. We use the latter bound and relation (34) to obtain

$$\begin{aligned}
(\sigma_2 - 1)\mathcal{D}_{\alpha^k}(x^k, d^k) &\leq \mathcal{D}_{\alpha^k}(x^{k+1}, d^k) - \mathcal{D}_{\alpha^k}(x^k, d^k) \\
&\leq \left\langle \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}\nabla F_i(x^{k+1}), d^k \right\rangle - \left\langle \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}\nabla F_i(x^k), d^k \right\rangle \\
&\leq \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k}\left\|\nabla F_i(x^{k+1}) - \nabla F_i(x^k)\right\|\left\|d^k\right\| \\
&\leq \sum_{i\in[m]} \frac{\lambda_i^k}{\alpha_i^k} b\left\|s^k\right\|\left\|d^k\right\| \\
&\leq \frac{b}{\alpha_{\min}}\left\|s^k\right\|\left\|d^k\right\|,
\end{aligned}$$

where the last inequality is due to the facts that $\lambda^k \in \Delta_m$ and $\alpha_i^k \geq \alpha_{\min},\ i \in [m]$. This, together with (37) and the fact that $\sigma_2 < 1$, implies

$$(1 - \sigma_2)\frac{\cos\beta^k}{\alpha_{\max}}\left\|d^k\right\|\left\|d^k_{SD}\right\| \leq \frac{b}{\alpha_{\min}}\left\|s^k\right\|\left\|d^k\right\|.$$

This concludes the proof.

Theorem 5.2 *Suppose that Assumption 2 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3. Then, $\{x^k\}$ converges R-linearly to a Pareto solution x^* of (MOP). Furthermore, we have*

$$\sum_{k\geq 0}\left\|x^k - x^*\right\| < \infty.$$

Proof The proof is similar to that in [22, Proposition 5], we omit it here.

5.3 Superlinear convergence

In the following, we consider the local superlinear convergence of Algorithm 3 under Assumption 2. As described in Theorem 4.3, to ensure the superlinear convergence of the proposed algorithm, the remaining assumptions are (b) and (f). First, we present a sufficient condition for assumption (f).

Proposition 5.3 *Suppose that $\{\lambda^k/\alpha^k\}$ is convergent and Assumption 2 holds, then*

$$\lim_{k\to\infty}\frac{\left\|(B_k-\sum\limits_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2F_i(x^k))s_k\right\|}{\|s_k\|}=0,$$

i.e., assumption (f) in Theorem 4.3 holds.

Proof Observe the continuity of $\{\lambda^k/\alpha^k\}$, the assertion can be obtained by using the similar arguments as in the proof of [29, Theorem 8.6] by regarding $\sum\limits_{i\in[m]}\frac{\lambda_i^k}{\alpha_i^k}\nabla^2F_i(x^k)$ as a whole.

The remaining assumption is how to ensure the unit stepsize.

Proposition 5.4 *Suppose that $\{\lambda^k/\alpha^k\}$ is convergent and Assumption 2 holds, then (34) holds with $t_k=1$ for sufficient large k.*

Proof By Proposition 5.3, we conclude that the assumption (f) holds. Without loss of generality, for a sufficiently small $\epsilon>0$, there exists K_ϵ such that, for all $k\geq K_\epsilon$,

$$\begin{aligned}
\mathcal{D}_{\alpha^k}(x^k+d^k,d^k)&\geq\left\langle\sum_{i\in[m]}\lambda_i^k\frac{\nabla F_i(x^k+d^k)}{\alpha_i^k},d^k\right\rangle\\
&\geq\left\langle\sum_{i\in[m]}\lambda_i^k\frac{\nabla F_i(x^k)}{\alpha_i^k},d^k\right\rangle+\left\|d^k\right\|^2_{\sum\limits_{i\in[m]}\lambda_i^k\frac{\nabla^2F_i(x^k)}{\alpha_i^k}}-\epsilon\left\|d^k\right\|^2\\
&=-\left\|d^k\right\|^2_{B_k}+\left\|d^k\right\|^2_{\sum\limits_{i\in[m]}\lambda_i^k\frac{\nabla^2F_i(x^k)}{\alpha_i^k}}-\epsilon\left\|d^k\right\|^2\\
&\geq-2\epsilon\|d^k\|^2\\
&\geq-\sigma_2\|d^k\|^2_{B_k}\\
&=\sigma_2\mathcal{D}_{\alpha^k}(x^k,d^k),
\end{aligned}$$

where the second inequality follows by the continuity of ∇^2F_i, the third inequality is given by assumption (f) and the last inequality is due to assumption (f), strong convexity of F_i and the fact that $\epsilon>0$ is sufficiently small. This concludes the proof.

Remark 5.3 Although relation (34) accepts unit stepsize for sufficiently large k, unit stepsize may not hold for relation (33) due to the Barzilai-Borwein parameter α^k. As is well known, the Barzilai-Borwein method without line search does not guarantee convergence. Therefore, to guarantee the unit stepsize in Wolfe line search (33)-(34), one possible way is to use line search as the repeat loop in Algorithm 2. However, each repeat loop in Algorithm 2 requires solving a subproblem, then the efficiency of the proposed method may be hampered by the line search. Notably, the Barzilai-Borwein method often achieves unit stepsize in numerical experiments [7].

6 Numerical results

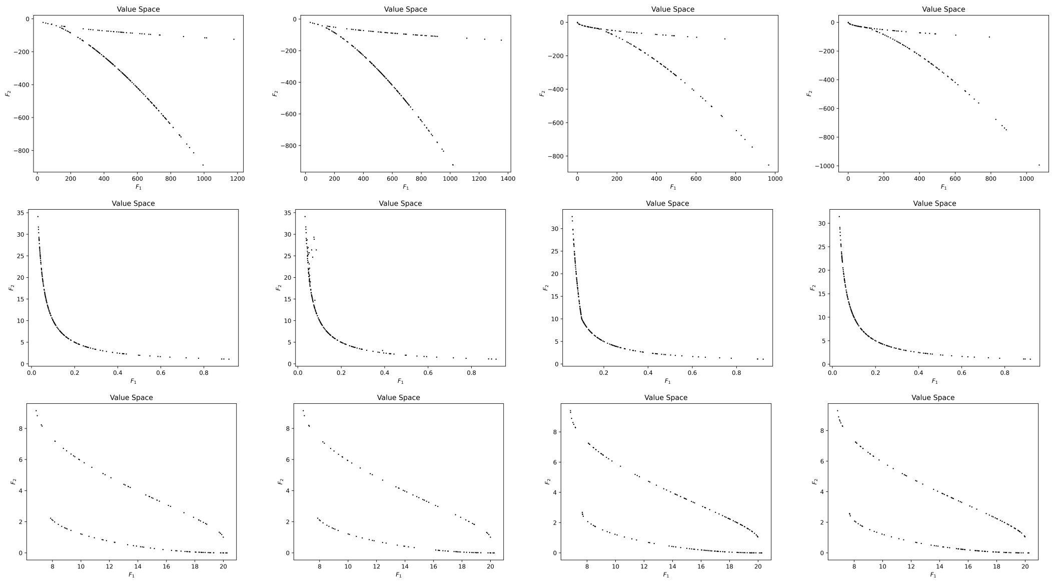
In this section, we present numerical results to demonstrate the performance of BBQNMO for various problems, where the variable metric is the trade-off of quasi-Newton approximation. We also compare BBQNMO with QNMO with Wolfe line search [33], MQNMO [1, 6, 22] with Wolfe line search and BBDMO [7] to show its efficiency. All numerical experiments were implemented in Python 3.7 and executed on a personal computer with an Intel Core i7-11390H, 3.40 GHz processor, and 16 GB of RAM. For BBDMO and BBQNMO, we set $\alpha_{\min} = 10^{-3}$ and $\alpha_{\max} = 10^3$ to truncate the Barzilai-Borwein's parameter. We use the Wolfe line search as in algorithm 3 in [22], and set $\sigma_1 = 10^{-4}$, $\sigma_2 = 0.1$ in Wolfe line search. We use $B_i^k = I$, $i \in [m]$ in QNMO, and $B_k = I$ in MQNMO and BBQNMO. To ensure that the algorithms terminate after a finite number of iterations, we use the stopping criterion $|\theta(x)| < 10^{-8}$ for all tested algorithms. We also set the maximum number of iterations to 500. For each problem, we use the same initial points for different tested algorithms. The initial points are randomly selected within the specified lower and upper bounds. The subproblem of QNMO is solved by `scipy.optimize`, a Python-embedded modelling language for optimization problems. Based on the Frank-Wolfe method, our codes solve the subproblems of MQNMO, BBDMO, and BBQNMO. The recorded averages from the 200 runs include the number of iterations, the number of function evaluations, and the CPU time.

6.1 Ordinary test problems

The tested algorithms are executed on several test problems, and the problem illustration is given in Table 2. The dimensions of variables and objective functions are presented in the second and third columns, respectively. x_L and x_U represent lower bounds and upper bounds of variables, respectively.

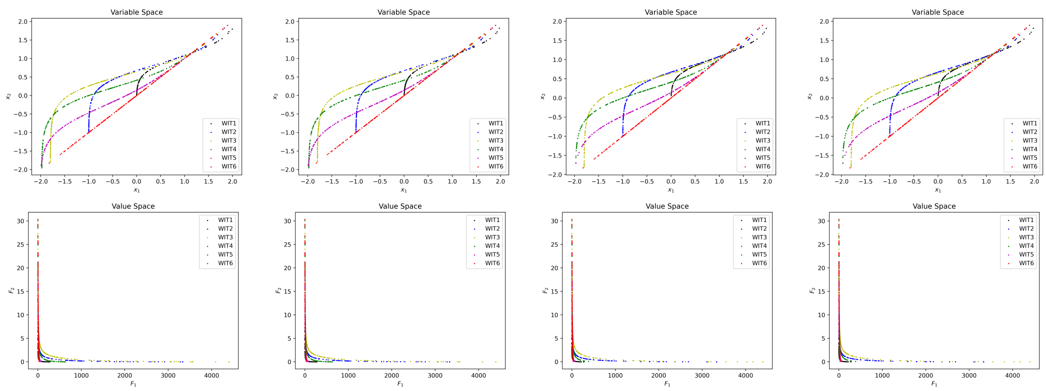
Problem	n	m	x_L	x_U	Reference
BK1	2	2	(-5,-5)	(10,10)	[20]
DD1	5	2	(-20,...,-20)	(20,...,20)	[10]
Deb	2	2	(0.1,0.1)	(1,1)	[11]
Far1	2	2	(-1,-1)	(1,1)	[20]
FDS	5	3	(-2,...,-2)	(2,...,2)	[14]
FF1	2	2	(-1,-1)	(1,1)	[20]
Hil1	2	2	(0,0)	(1,1)	[19]
Imbalance1	2	2	(-2,-2)	(2,2)	[7]
Imbalance2	2	2	(-2,-2)	(2,2)	[7]
JOS1a	50	2	(-2,...,-2)	(2,...,2)	[21]
JOS1b	100	2	(-2,...,-2)	(2,...,2)	[21]
JOS1c	100	2	(-50,...,-50)	(50,...,50)	[21]
JOS1d	100	2	(-100,...,-100)	(100,...,100)	[21]
LE1	2	2	(-5,-5)	(10,10)	[20]
PNR	2	2	(-2,-2)	(2,2)	[32]
VU1	2	2	(-3,-3)	(3,3)	[20]
WIT1	2	2	(-2,-2)	(2,2)	[41]
WIT2	2	2	(-2,-2)	(2,2)	[41]
WIT3	2	2	(-2,-2)	(2,2)	[41]
WIT4	2	2	(-2,-2)	(2,2)	[41]
WIT5	2	2	(-2,-2)	(2,2)	[41]
WIT6	2	2	(-2,-2)	(2,2)	[41]

Table 2: Description of all test problems used in numerical experiments.



(a) QNMO (b) MQNMO (c) BBDMO (d) BBQNMO

Fig. 1: Numerical results in value space for problems DD1, VU1 and PNR.



(a) QNMO (b) MQNMO (c) BBDMO (d) BBQNMO

Fig. 2: Numerical results in variable space and value space obtained by the tested algorithms for problems WIT1-6.

For each test problem, the number of average iterations (iter), number of average function evaluations (feval), and average CPU time (time(ms)) of the different algorithms are listed in Table 3. The problems DD1, Deb, FDS, Imbalance1-2, VU1 and WIT1-2 involve imbalanced objective functions, such as higher-order and exponential functions, leading to poor MQNMO performance. In contrast to MQNMO, the other methods perform well on these problems, demonstrating their ability to alleviate objectives' imbalances. Nevertheless, BBDMO and BBQNMO require much less CUP time, particularly for high-dimensional problems, than QNMO. The BBDMO and BBQNMO exhibit superior performance for the test problems due to the good conditioning.

Problem	QNMO			MQNMO			BBDMO			BBQNMO		
	iter	feval	time	iter	feval	time	iter	feval	time	iter	feval	time
BK1	1.00	2.00	4.02	**1.00**	2.00	**0.39**	**1.00**	**1.00**	0.40	**1.00**	**1.00**	0.55
DD1	11.74	12.81	40.04	43.14	163.92	10.77	**6.06**	**6.20**	**1.41**	8.22	16.72	3.20
Deb	5.37	10.58	9.00	55.14	350.18	16.50	3.74	5.85	**0.99**	**3.36**	**4.70**	1.51
Far1	9.24	13.05	16.29	39.13	193.53	20.71	49.26	49.76	11.37	**7.73**	**18.76**	**4.04**
FDS	10.68	16.87	59.00	169.66	1132.75	242.12	**4.23**	**4.47**	**3.08**	4.66	6.07	5.22
FF1	3.88	**5.31**	6.97	16.00	61.75	4.40	4.28	5.50	**1.17**	**3.57**	**5.32**	1.51
Hil1	5.78	9.58	9.60	12.87	49.58	4.34	10.11	10.93	2.66	**4.46**	**8.64**	**2.05**
Imbalance1	2.51	5.14	6.52	53.43	212.21	13.03	2.57	**3.50**	**0.79**	**2.50**	7.37	1.18
Imbalance2	1.51	5.38	4.71	227.05	1587.22	61.53	**1.00**	**1.00**	**0.39**	**1.00**	**1.00**	0.55
JOS1a	2.47	6.85	33.38	2.51	6.88	0.79	**1.00**	**1.00**	**0.39**	**1.00**	**1.00**	0.55
JOS1b	2.65	7.96	70.40	2.85	8.19	1.10	**1.00**	**1.00**	**0.39**	**1.00**	**1.00**	0.64
JOS1c	3.16	8.73	129.75	3.36	8.94	1.34	**1.00**	**1.00**	**0.39**	**1.00**	**1.00**	0.61
JOS1d	3.37	9.03	187.07	3.67	9.32	1.34	**1.00**	**1.00**	**0.39**	**1.00**	**1.00**	0.57
LE1	**3.66**	**5.85**	6.48	7.54	17.61	1.98	3.86	6.11	**1.10**	3.99	6.14	1.74
PNR	**2.55**	5.12	6.11	7.68	19.83	1.92	3.47	**3.75**	**0.94**	3.58	4.59	1.41
VU1	36.12	37.32	65.04	332.61	2713.48	93.35	13.98	14.03	3.50	**8.32**	**13.49**	**3.16**
WIT1	**2.52**	5.39	6.31	48.57	248.86	12.53	3.06	**3.14**	**0.79**	2.93	3.39	1.17
WIT2	3.85	9.04	9.58	76.10	382.56	20.10	3.42	3.52	**1.11**	**3.22**	**3.37**	1.34
WIT3	**3.83**	7.81	9.28	35.08	131.15	8.91	4.39	4.48	**1.12**	4.10	**4.20**	1.63
WIT4	**2.98**	5.05	7.08	6.90	13.71	1.84	4.56	4.61	**1.13**	4.28	**4.35**	1.68
WIT5	**3.05**	4.59	6.82	4.87	8.45	1.32	3.61	3.63	**0.94**	3.55	**3.59**	1.78
WIT6	1.05	2.00	3.29	**1.00**	2.00	0.56	**1.00**	**1.00**	**0.40**	**1.00**	**1.00**	0.57

Table 3: Number of average iterations (iter), number of average function evaluations (feval), and average CPU time (time(ms)) of QNMO, MQNMO, BBDMO, and BBQNMO implemented on different test problems.

6.2 Quadratic ill-conditioned problems

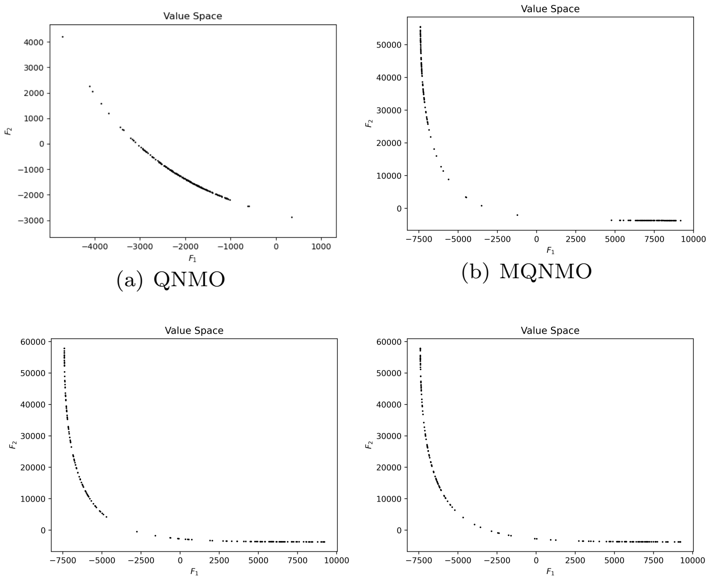
In this subsection, we test the algorithm on ill-conditioned problems. We consider a series of quadratic problems defined as follows:

$$F_i(x) = \frac{1}{2}\langle x, A_i x\rangle + \langle b_i, x\rangle, \ i = 1, 2,$$

where A_i is a positive definite matrix. We set $A_i = H_i D_i H_i^T$, where H_i is a random orthogonal matrix and $D_i = Diag(d_i^1, d_i^2, ..., d_i^n)$ with $\max_j d_i^j / \min_j d_i^j = \kappa_i$. The problem illustration is given in Table 4. The second and third columns present the objective functions' dimension and condition numbers, respectively. While x_L and x_U represent the lower and upper bounds of the variables, respectively.

Problem	n	(κ_1, κ_2)	x_L	x_U
QPa	10	(10, 10)	10[-1,...,-1]	10[1,...,1]
QPb	10	$(10^2, 10^2)$	10[-1,...,-1]	10[1,...,1]
QPc	100	$(10^2, 10^2)$	100[-1,...,-1]	100[1,...,1]
QPd	100	$(10^3, 10^3)$	100[-1,...,-1]	100[1,...,1]
QPe	500	$(10^3, 10^3)$	500[-1,...,-1]	500[1,...,1]
QPf	500	$(10^4, 10^4)$	500[-1,...,-1]	500[1,...,1]
QPg	100	$(10^5, 10^2)$	100[-1,...,-1]	100[1,...,1]

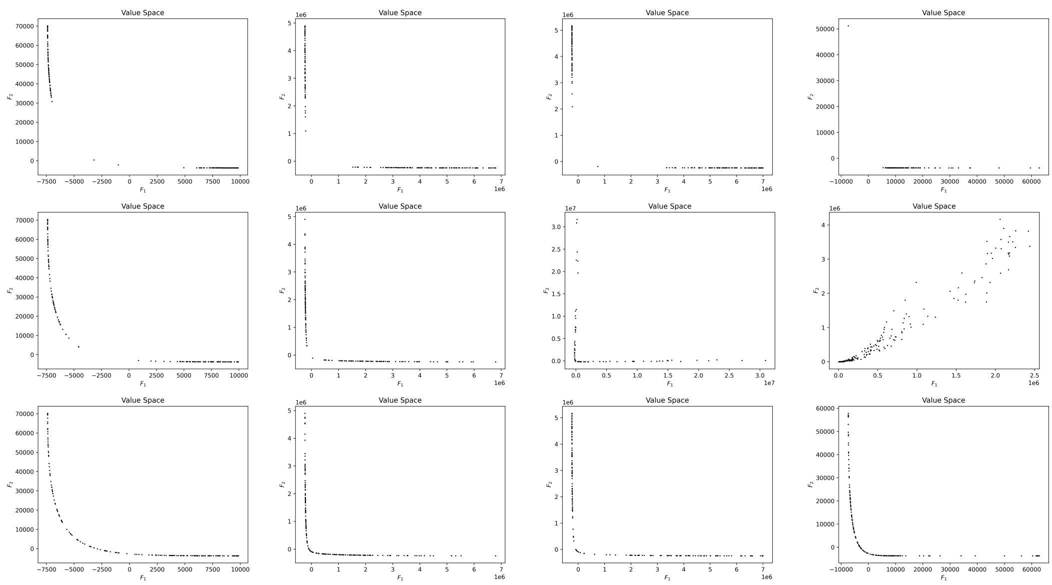
Table 4: Description of quadratic problems.



(a) QNMO (b) MQNMO

(c) BBDMO (d) BBQNMO

Fig. 3: Numerical results in value space for problem **QPc**.



(a) QPd (b) QPe (c) QPf (d) QPg

Fig. 4: Numerical results in value space obtained by MQNMO (**top**), BBDMO (**middle**) and BBQNMO for problems QPd, QPe, QPf, and QPg.

Problem	QNMO			MQNMO			BBDMO			BBQNMO		
	iter	feval	time	iter	feval	time	iter	feval	time	iter	feval	time
QPa	16.44	32.44	122.56	16.03	34.61	3.77	12.95	14.45	2.92	**10.19**	**14.58**	**3.67**
QPb	**19.30**	**38.56**	156.21	27.31	81.60	**6.93**	46.08	74.38	9.92	21.63	41.78	7.65
QPc	89.99	414.92	31355.87	109.59	648.34	44.16	57.44	89.41	**12.38**	**36.78**	**69.93**	15.55
QPd	–	–	–	140.64	804.35	56.43	195.52	388.28	44.40	**45.41**	**93.64**	**19.24**
QPe	–	–	–	458.18	4140.66	5792.92	196.19	369.88	**133.19**	**68.94**	**117.23**	795.10
QPf	–	–	–	473.45	4203.23	6039.30	456.93	1217.09	**456.13**	**121.86**	**291.31**	1378.27
QPg	–	–	–	226.40	1457.88	96.50	464.46	2933.31	190.53	**80.61**	**172.28**	**35.37**

Table 5: Number of average iterations (iter), number of average function evaluations (feval), and average CPU time (time(ms)) of QNMO, MQNMO, BBDMO, and BBQNMO implemented on quadratic problems.

Table 5 presents the number of average iterations (iter), number of average function evaluations (feval), and average CPU time (time(ms)) over 200 experimental runs for every quadratic problem. All the tested methods exhibit convergence for well-conditioned and low-dimensional problems (QPa-b), except that QNMO requires significantly more CPU time due to the expensive per-step cost. The CPU time required by QNMO increases substantially for problem QPc, making it impractical for high-dimensional problems. For ill-conditional and high-dimensional problems (QPd-f), QNMO fails to converge (this is mainly due to the solver of the subproblems.), while BBQNMO significantly outperforms MQNMO and BBDMO. It is worth noting that MQNMO and BBQNMO are second-order methods which have the potential to capture the local curvature for ill-conditioned problems. However, MQNMO can not handle the imbalances among the objectives, resulting in biased solutions (see Fig. 4). On the other hand, BBDMO is a first-order method that can cope with weakly ill-conditioned problems (QPb-e) due to the Barzilai-Borwein rule, but fails to converge on extremely ill-conditioned problems(QPf). In summary, the primary experiment results confirm that for ill-conditional and high-dimensional MOPs, the proposed B-BQNMO can better balance the curvature exploration and per-step cost than QNMO, MQNMO and BBDMO.

7 Conclusions

In this paper, we proposed a preconditioned Barzilai-Borwein descent method for MOPs that enjoys cheap per-step cost and is not sensitive to imbalances and conditioning. Theoretical analysis indicates that this method can effectively mitigate imbalances among objectives and achieve rapid convergence with appropriate metric selection. Our numerical results validate the superiority of the proposed method, incorporating the trade-off of quasi-Newton approximation. It significantly outperforms QNMO, MQNMO, and BBDMO, particularly in the case of large-scale and ill-conditioned problems. In addition, this paper provides a new insight into preconditioning, and highlights the essential role of the Barzilai-Borwein method for preconditioning in multiobjective optimization.

From a methodological perspective, it may be worth considering the following points:

- Prudent and Souza [34] studied the global convergence of the BFGS-type algorithm for nonconvex MOPs, it is worth considering the global convergence of BBQNMO. Note that we update a common matrix B_k, then the Dennis-Moré condition and asymptotic convergence hold without all objectives being locally convex.
- To balance the per-iteration cost and better curvature exploration, we choose a preconditioner from the perspective of implicit scalarization, and the Barzilai-Borwein method is embedded in the preconditioning method. This paves the way for the development of efficient high-order [12] and high-order regularized methods [28] for MOPs.

– Chen et al. recently [9] established superlinear convergence of the Newton-type proximal method for MOPs. Consequently, it is meaningful to extend PBBMO for solving ill-conditioned multiobjective composite problems. Given the potential for expensive proximal operators with non-diagonal matrices, exploring approaches that capture the local geometry using diagonal matrices, such as diagonal Barzilai-Borwein stepsize [30] is practical.

References

1. M. A. T. Ansary and G. Panda. A modified quasi-Newton method for vector optimization problem. *Optimization*, 64(11):2289–2306, 2015.
2. M. A. T. Ansary and G. Panda. A globally convergent SQCQP method for multiobjective optimization problems. *SIAM Journal on Optimization*, 31(1):91–113, 2021.
3. H. Attouch, G. Garrigos, and X. Goudou. A dynamic gradient approach to Pareto optimization with nonsmooth convex objective functions. *Journal of Mathematical Analysis and Applications*, 422(1):741–771, 2015.
4. H. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM Journal on Optimization*, 15(4):953–970, 2005.
5. G. A. Carrizo, P. A. Lotito, and M. C. Maciel. Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem. *Mathematical Programming*, 159(1):339–369, 2016.
6. J. Chen, G. X. Li, and X. M. Yang. Variable metric method for unconstrained multiobjective optimization problems. *Journal of the Operations Research Society of China*, 11(3):409–438, 2023.
7. J. Chen, L. P. Tang, and X. M. Yang. A Barzilai-Borwein descent method for multiobjective optimization problems. *European Journal of Operational Research*, 311(1):196–209, 2023.
8. J. Chen, L. P. Tang, and X. M. Yang. Barzilai-Borwein proximal gradient methods for multiobjective composite optimization problems with improved linear convergence. *arXiv preprint arXiv:2306.09797v2*, 2023.
9. J. Chen, L. P. Tang, and X. M. Yang. On the convergence of Newton-type proximal gradient method for multiobjective optimization problems. *arXiv preprint arXiv:2308.10140v1*, 2023.
10. I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
11. K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.
12. N. Doikov and Y. Nesterov. High-order optimization methods for fully composite problems. *SIAM Journal on Optimization*, 32(3):2402–2427, 2022.
13. G. Evans. Overview of techniques for solving multiobjective mathematical programs. *Management Science*, 30(11):1268–1282, 1984.
14. J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626, 2009.
15. J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
16. J. Fliege and A. I. F. Vaz. A method for constrained multiobjective optimization based on SQP techniques. *SIAM Journal on Optimization*, 26(4):2091–2119, 2016.
17. J. Fliege and R. Werner. Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research*, 234(2):422–433, 2014.

18. L. M. Graña Drummond and A. N. Iusem. A projected gradient method for vector optimization problems. *Computational Optimization and Applications*, 28(1):5–29, 2004.
19. C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.
20. S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.
21. Y. Jin, M. Olhofer, and B. Sendhoff. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1042–1049, 2001.
22. M. Lapucci and P. Mansueto. A limited memory quasi-Newton approach for multi-objective optimization. *Computational Optimization and Applications*, 85(1):33–73, 2023.
23. L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM Journal on Optimization*, 28(3):2690–2720, 2018.
24. R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
25. V. Morovati and L. Pourkarimi. Extension of Zoutendijk method for solving constrained multiobjective optimization problems. *European Journal of Operational Research*, 273(1):44–57, 2019.
26. V. Morovati, L. Pourkarimi, and H. Basirzadeh. Barzilai and Borwein's method for multiobjective optimization problems. *Numerical Algorithms*, 72(3):539–604, 2016.
27. H. Mukai. Algorithms for multicriterion optimization. *IEEE Transactions on Automatic Control*, 25(2):177–186, 1980.
28. Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.
29. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.
30. Y. Park, S. Dhar, S. Boyd, and M. Shah. Variable metric proximal gradient method with diagonal Barzilai-Borwein stepsize. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3597–3601, 2020.
31. Ž. Povalej. Quasi-Newton's method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 255:765–777, 2014.
32. M. Preuss, B. Naujoks, and G. Rudolph. Pareto set and EMOA behavior for simple multimodal multiobjective functions", booktitle="parallel problem solving from nature - ppsn ix. pages 513–522, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
33. L. F. Prudente and D. R. Souza. A quasi-Newton method with Wolfe line searches for multiobjective optimization. *Journal of Optimization Theory and Applications*, 194(3):1107–1140, 2022.
34. L. F. Prudente and D. R. Souza. Global convergence of a BFGS-type algorithm for nonconvex multiobjective optimization problems. *arXiv preprint arXiv:2307.08429*, 2023.
35. S. J. Qu, M. Goh, and F. T. Chan. Quasi-Newton methods for solving multiobjective optimization. *Operations Research Letters*, 39(5):397–399, 2011.
36. O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
37. K. Sonntag and S. Peitz. Fast convergence of inertial multiobjective gradient-like systems with asymptotic vanishing damping. *arXiv preprint arXiv:2307.00975*, 2023.
38. K. Sonntag and S. Peitz. Fast multiobjective gradient methods with Nesterov acceleration via inertial gradient-like systems. *Journal of Optimization Theory and Applications*, pages 1–44, 2024.

39. H. Tanabe, E. H. Fukuda, and N. Yamashita. Proximal gradient methods for multiobjective optimization and their applications. *Computational Optimization and Applications*, 72:339–361, 2019.
40. H. Tanabe, E. H. Fukuda, and N. Yamashita. An accelerated proximal gradient method for multiobjective optimization. *Computational Optimization and Applications*, 86:421–455, 2023.
41. K. Witting. *Numerical algorithms for the treatment of parametric multiobjective optimization problems and applications.* PhD thesis, Paderborn, Universität Paderborn, Diss., 2012, 2012.

Acknowledgements This work was funded by the Major Program of the National Natural Science Foundation of China [grant numbers 11991020, 11991024]; the National Natural Science Foundation of China [grant numbers 11971084, 12171060]; NSFC-RGC (Hong Kong) Joint Research Program [grant number 12261160365]; the Team Project of Innovation Leading Talent in Chongqing [grant number CQYC20210309536]; the Natural Science Foundation of Chongqing [grant number ncamc2022-msxm01]; Major Project of Science and Technology Research Rrogram of Chongqing Education Commission of China [grant number KJZD-M202300504]; and Foundation of Chongqing Normal University [grant numbers 22XLB005, 22XLB006].