manuscript No.
(will be inserted by the editor)

A Subspace Minimization Barzilai-Borwein Method for Multiobjective Optimization Problems

Jian Chen · Liping Tang · Xinmin Yang

Received: date / Accepted: date

Abstract Nonlinear conjugate gradient methods have recently garnered significant attention within the multiobjective optimization community. These methods aim to maintain consistency in conjugate parameters with their single-objective optimization counterparts. However, the preservation of the attractive conjugate property of search directions remains uncertain, even for quadratic cases, in multiobjective conjugate gradient methods. This loss of interpretability of the last search direction significantly limits the applicability of these methods. To shed light on the role of the last search direction, we introduce a novel approach called the subspace minimization Barzilai-Borwein method for multiobjective optimization problems (SMBBMO). In SMBBMO, each search direction is derived by optimizing a preconditioned Barzilai-Borwein subproblem within a two-dimensional subspace generated by the last search direction and the current Barzilai-Borwein descent direction. Furthermore, to ensure the global convergence of SMBBMO, we employ a modified Cholesky factorization on a transformed scale matrix, capturing the local curvature information of the problem within the two-dimensional subspace. Under mild assumptions, we establish both global and Q-linear convergence of the proposed method. Finally, comparative numerical experiments confirm the efficacy of SMBBMO, even when tackling large-scale and ill-conditioned problems.

Keywords Multiobjective optimization · Subspace method · Barzilai-Borwein's method · Global convergence

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

1 Introduction

An unconstrained multiobjective optimization problem (MOP) is typically formulated 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. This type of problem finds widespread applications across various domains, including engineering [29], economics [17], management science [13], and machine learning [35], among others. These applications often involve the simultaneous optimization of multiple objectives. However, achieving a single solution that optimizes all objectives is often impractical. Therefore, optimality is defined by *Pareto optimality* or *efficiency*. A solution is deemed Pareto optimal or efficient if no objective can be improved without sacrificing the others.

In the past two decades, multiobjective gradient descent methods have gained significant traction within the multiobjective optimization community. These methods determine descent directions by solving subproblems, followed by the application of line search techniques along these directions to ensure sufficient improvement across all objectives. The origins of multiobjective gradient descent methods can be traced back to pioneering works by Mukai [31] and Fliege and Svaiter [15]. The latter 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, 14, 16, 21, 28, 30, 32, 34] and references therein).

Recently, Lucambio Pérez and Prudente [28] made significant advancements by extending Wolfe line search and the Zoutendijk condition to multiobjective optimization, thereby facilitating the exploration of multiobjective nonlinear conjugate gradient methods. These methods leverage both the current steepest descent direction and the last search direction to construct the current search direction, ensuring consistency in conjugate parameters with their counterparts in single-objective optimization problems (SOPs), such as Fletcher–Reeves [28], Conjugate descent [28], Dai–Yuan [28], Polak–Ribière–Polyak [28], Hestenes–Stiefel [28], Hager–Zhang [20, 24] and Liu–Storey [19].

The linear conjugate gradient method exhibits finite termination for convex quadratic minimization, owing to its attractive conjugate property. In multiobjective optimization, Fukuda et al. [18] proposed a conjugate directions-type that achieves finite termination for strongly convex quadratic MOPs. Unfortunately, the method cannot be extended to non-quadratic cases. Moreover, the attractive conjugate property of search directions remains unknown for quadratic cases in existing multiobjective conjugate gradient methods. Conversely, in single-objective optimization, the absence of the conjugate property severely constrains the application of nonlinear conjugate gradient methods, particularly in large-scale non-quadratic cases. To tackle this challenge, Yuan and Stoer [38] devised the subspace minimization conjugate gradient (SMCG) method for SOPs. The search directions of SMCG are obtained by optimizing approximate models within two-dimensional subspaces generated by gradient and last search directions. Consequently, the conjugate parameters of SMCG is optimal with respect to approximate models. An advantage of SMCG is that lower-dimensional subspaces enable us to solve the corresponding subproblems efficiently. Furthermore, in many cases, the subspace approaches achieve comparable theoretical properties to their full-space counterparts. As described above, the extension of SMCG to MOPs is of great interest. Naturally, the key issues for such a method are how to choose the subspaces and how to obtain the approximate models for better curvature exploration.

In this paper, we propose a subspace minimization Barzilai-Borwein method for MOPs (SMBBMO), the choice of subspaces and approximate models are described as follows:

• *Subspace*: Chen et al. [5, 6] highlighted that imbalances among objectives seriously decelerate the convergence of steepest descent method. To alleviate the impact of imbalances, we propose constructing a subspace using both the current Barzilai-Borwein descent direction [5] and the previous search direction. This construction is defined as follows:

$$d_k = \begin{cases} v_k, & \text{if } k = 0, \\ \mu_k v_k + \nu_k d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

where $(\mu_k, \nu_k)^T \in \mathbb{R}^2$, v_k is the Barzilai-Borwein descent direction at x^k. Notably, d^k follows a formula similar to the spectral conjugate gradient direction due to the Barzilai-Borwein descent direction v_k. However, existing multiobjective spectral conjugate gradient methods [12, 22] confine their search directions to the subspace generated by the current steepest descent direction and the previous search direction. Consequently, these approaches may not fully leverage the benefits of spectral information for each objective.

• *Approximate model*: To strike a balance between per-iteration cost and improved curvature exploration, we adopt the preconditioned Barzilai-Borwein subproblem [7] as the initial approximate model. To circumvent the need for matrix calculations, we utilize finite differences of gradients to estimate matrix-vector products. Consequently, the local curvature information of the problem in the two-dimensional subspace is represented by a 2×2 matrix. Similar to SMCG (Li et al., 2024), the global convergence of SMBBMO in non-convex cases remains uncertain when employing the 2×2 matrix selection strategy of SMCG. Motivated by the work of Lapucci et al. [26], we apply a modified Cholesky factorization to a transformed scale matrix. This enables us to establish global convergence for the proposed method.

The primary objective of SMBBMO is to achieve a faster convergence rate compared to the Barzilai-Borwein descent method while maintaining lower computational costs than the preconditioned Barzilai-Borwein method.

The paper is organized as follows. Section 2 introduces necessary notations and definitions to be utilized later. In Section 3, we propose SMBBMO and explore the choice of subspace and approximate model. The global convergence and Q-linear convergence of SMBBMO are established in Section 4. Section 5 presents numerical results demonstrating the efficiency of SMBBMO. Finally, conclusions are drawn 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.*

3 Subspace minimization Barzilai-Borwein descent method for MOPs

Let us consider the *subspace minimization* descent direction subproblem:

$$\min_{d \in \Omega_k} \max_{i \in [m]} q_i^k(d), \tag{1}$$

where $q_i^k(\cdot)$ is approximation model for F_i at x^k, and Ω_k is a subspace. The key issues for the descent direction are how to choose the subspaces and how to obtain the approximate models in corresponding subspaces quickly.

3.1 Selection of subspace

Nonlinear conjugate gradient methods utilize the current steepest descent direction and the previous descent direction to construct new descent direction. In order to compare with nonlinear conjugate gradient methods, we denote $\Omega_k = Span\{v_k, d_{k-1}\}$ for $k > 1$. The choice for v_k is essential, recall that steepest descent direction often accepts a very small stepsize due to the imbalances between objectives, here we set v_k the Barzilai-Borwein descent direction [5] at x^k, namely,

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

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

$$\alpha_i^k = \begin{cases} \max\left\{\alpha_{\min}, \min\left\{\frac{\langle s_{k-1}, y_i^{k-1}\rangle}{\|s_{k-1}\|^2}, \alpha_{\max}\right\}\right\}, & \langle s_{k-1}, y_i^{k-1}\rangle > 0, \\ \max\left\{\alpha_{\min}, \min\left\{\frac{\|y_i^{k-1}\|}{\|s_{k-1}\|}, \alpha_{\max}\right\}\right\}, & \langle s_{k-1}, y_i^{k-1}\rangle < 0, \\ \alpha_{\min}, & \langle s_{k-1}, y_i^{k-1}\rangle = 0, \end{cases} \tag{3}$$

for all $i \in [m]$, where $\alpha_{\max}$ is a sufficient large positive constant and $\alpha_{\min}$ is a sufficient small positive constant, $s_{k-1} = x^k - x^{k-1}$, $y_i^{k-1} = \nabla F_i(x^k) - \nabla F_i(x^{k-1})$, $i \in [m]$.

3.2 Selection of approximate model

In general, iterative methods frequently leverage a quadratic model as it effectively approximates the objective function within a small neighborhood of the minimizer. Striving for a more optimal balance between computational cost and enhanced curvature exploration, we adopt the approximate model proposed in Chen et al. [7]:

$$q_i^k(d) := \frac{\langle \nabla F_i(x^k), d\rangle}{\bar{\alpha}_i^k} + \frac{1}{2}\|d\|^2_{B_k}, \tag{4}$$

where $\bar{\alpha}^k \in \mathbb{R}^m_{++}$ is as follows:

$$\bar{\alpha}_i^k = \begin{cases} \max\left\{\alpha_{\min}, \min\left\{\frac{\langle s_{k-1}, y_i^{k-1}\rangle}{\|s_{k-1}\|^2_{B_k}}, \alpha_{\max}\right\}\right\}, & \langle s_{k-1}, y_i^{k-1}\rangle > 0, \\ \max\left\{\alpha_{\min}, \min\left\{\frac{\|y_i^{k-1}\|}{\|B_k s_{k-1}\|}, \alpha_{\max}\right\}\right\}, & \langle s_{k-1}, y_i^{k-1}\rangle < 0, \\ \alpha_{\min}, & \langle s_{k-1}, y_i^{k-1}\rangle = 0, \end{cases} \tag{5}$$

and B_k is a positive definite matrix. From a preconditioning perspective, as described in [7], a judicious choice for B_k is to approximate the variable aggregated Hessian, i.e.,

$$B_k \approx \sum_{i\in[m]} \frac{\bar{\lambda}_i^{k-1}}{\bar{\alpha}_i^{k-1}} \nabla^2 F_i(x^k),$$

where $\bar{\lambda}^{k-1} \in \Delta_m$ the dual solution of (1) at x^{k-1}.

3.3 Subspace minimization Barzilai-Borwein descent method

Recall that $s_{k-1} = x^k - x^{k-1} = t_{k-1}d_{k-1}$, by substituting $d = \mu v_k + \nu s_{k-1}$ into (1), it can be reformulated as

$$\min_{(\mu,\nu)^T\in\mathbb{R}^2} \max_{i\in[m]} \left\langle \begin{pmatrix} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k \right\rangle \\ \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, s_{k-1} \right\rangle \end{pmatrix}, \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle + \frac{1}{2}\left\langle \begin{pmatrix} \mu \\ \nu \end{pmatrix}, \begin{pmatrix} \langle v_k, B_k v_k\rangle & \langle v_k, B_k s_{k-1}\rangle \\ \langle s_{k-1}, B_k v_k\rangle & \langle s_{k-1}, B_k s_{k-1}\rangle \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle. \tag{6}$$

Recall that $B_k \approx \sum_{i\in[m]} \frac{\bar{\lambda}_i^{k-1}}{\bar{\alpha}_i^{k-1}} \nabla^2 F_i(x^k)$, to avoid calculating the matrix, we use finite differences of gradient to estimate matrix-vector products, i,e,,

$$B_k s_{k-1} \approx \sum_{i\in[m]} \frac{\bar{\lambda}_i^{k-1}}{\bar{\alpha}_i^{k-1}} (\nabla F_i(x^k) - \nabla F_i(x^{k-1})),$$

and

$$B_k v_k \approx \sum_{i\in[m]} \frac{\bar{\lambda}_i^{k-1}}{\bar{\alpha}_i^{k-1}} (\nabla F_i(x^k) - \nabla F_i(x^k - v_k)).$$

We denote

$$y^{k-1} := \sum_{i\in[m]} \frac{\bar{\lambda}_i^{k-1}}{\bar{\alpha}_i^{k-1}} (\nabla F_i(x^k) - \nabla F_i(x^{k-1})),$$

$$y_v^{k-1} := \sum_{i\in[m]} \frac{\bar{\lambda}_i^{k-1}}{\bar{\alpha}_i^{k-1}} (\nabla F_i(x^k) - \nabla F_i(x^k - v_k)),$$

and

$$H^k \approx \begin{pmatrix} \rho_1^k & \langle v_k, y^{k-1} \rangle \\ \langle v_k, y^{k-1} \rangle & \rho_2^k \end{pmatrix}, \tag{7}$$

where $\rho_1^k \approx \langle v_k, y_v^{k-1} \rangle$, $\rho_2^k \approx \langle s_{k-1}, y^{k-1} \rangle$. Then the *subspace minimization Barzilai-Borwein descent* direction $d_k = \mu_k v_k + \nu_k s_{k-1}$, where $(\mu_k, \nu_k)^T \in \mathbb{R}^2$ is the optimal solution of the following subproblem:

$$\min_{(\mu,\nu)^T \in \mathbb{R}^2} \max_{i\in[m]} \left\langle \begin{pmatrix} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k \right\rangle \\ \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, s_{k-1} \right\rangle \end{pmatrix}, \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle + \frac{1}{2} \left\langle \begin{pmatrix} \mu \\ \nu \end{pmatrix}, H^k \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle, \tag{8}$$

where $\bar{\alpha}^k \in \mathbb{R}_{++}^m$ is as follows:

$$\bar{\alpha}_i^k = \begin{cases} \max\left\{\alpha_{\min}, \min\left\{\frac{\langle s_{k-1}, y_i^{k-1} \rangle}{\rho_2^k}, \alpha_{\max}\right\}\right\}, & \langle s_{k-1}, y_i^{k-1} \rangle > 0, \\ \max\left\{\alpha_{\min}, \min\left\{\frac{\|y_i^{k-1}\|}{\|y^{k-1}\|}, \alpha_{\max}\right\}\right\}, & \langle s_{k-1}, y_i^{k-1} \rangle < 0, \\ \alpha_{\min}, & \langle s_{k-1}, y_i^{k-1} \rangle = 0. \end{cases} \tag{9}$$

To ensure that d_k is a descent direction, two conditions are required: $\rho_2^k > 0$ and H^k is positive definite. Here, we initially assume that these two conditions hold in (8). Denote

$$\theta(x^k) := \min_{(\mu,\nu)^T \in \mathbb{R}^2} \max_{i\in[m]} \left\langle \begin{pmatrix} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k \right\rangle \\ \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, s_{k-1} \right\rangle \end{pmatrix}, \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle + \frac{1}{2} \left\langle \begin{pmatrix} \mu \\ \nu \end{pmatrix}, H^k \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle,$$

and

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

Indeed, problem (8) can be equivalently rewritten as the following smooth quadratic problem:

$$\begin{aligned} \min_{(t,d)\in\mathbb{R}\times\mathbb{R}^n} \quad & t+\frac{1}{2}\left\langle \begin{pmatrix}\mu\\ \nu\end{pmatrix}, H^k\begin{pmatrix}\mu\\ \nu\end{pmatrix}\right\rangle, \\ \text{s.t.} \quad & \left\langle \begin{pmatrix} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k\right\rangle \\ \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, s_{k-1}\right\rangle \end{pmatrix}, \begin{pmatrix}\mu\\ \nu\end{pmatrix}\right\rangle \leq t,\ \ i\in[m]. \end{aligned} \tag{QP}$$

As described in [5], the problem (QP) can be efficiently solved via its dual. It is worth noting that $\bar{\lambda}^{k-1}$ should represent the dual solution of (QP) at x^{k-1} in this setting. By KKT conditions, we have

$$\theta(x^k)=\frac{1}{2}\mathcal{D}_{\bar{\alpha}^k}(x^k,d_k), \tag{10}$$

and

$$\mathcal{D}_{\bar{\alpha}^k}(x^k,d_k)=-\left\langle \begin{pmatrix}\mu_k\\ \nu_k\end{pmatrix}, H^k\begin{pmatrix}\mu_k\\ \nu_k\end{pmatrix}\right\rangle. \tag{11}$$

The remaining questions are: How do we ensure that $\rho_2^k>0$ and H^k is positive definite?

3.3.1 Selection of ρ_2^k

Note that in nonconvex cases $\langle s_{k-1},y^{k-1}\rangle\leq 0$ can hold, then we set

$$\rho_2^k=\begin{cases} \langle s_{k-1},y^{k-1}\rangle, & \langle s_{k-1},y^{k-1}\rangle>0,\\ \mathcal{D}_{\bar{\alpha}^{k-1}}(x^k,s_{k-1})-\sum\limits_{i\in[m]}\bar{\lambda}_i^{k-1}\left\langle \nabla F_i(x^{k-1})/\bar{\alpha}_i^{k-1},s_{k-1}\right\rangle, & \text{otherwise.}\end{cases} \tag{12}$$

We introduce the following Wolfe line search to ensure $\rho_2^k>0$.

$$(F_i(x^k+td_k)-F_i(x^k))/\bar{\alpha}_i^k\leq\sigma_1 t\mathcal{D}_{\bar{\alpha}^k}(x^k,d_k),\ \forall i\in[m], \tag{13}$$

$$\mathcal{D}_{\bar{\alpha}^k}(x^k+td_k,d_k)\geq\sigma_2\mathcal{D}_{\bar{\alpha}^k}(x^k,d_k). \tag{14}$$

To ensure the Wolfe line search is well-defined, we require the following assumption.

Assumption 3.1 *For any $x^0\in\mathbb{R}^n$, the level set $\mathcal{L}_F(x^0)=\{x:\ F(x)\preceq F(x^0)\}$ is compact.*

Proposition 3.1 *Suppose that Assumption 3.1 holds. Let d_k is a descent direction, $0<\sigma_1\leq\sigma_2<1$. Then, there exists an interval $[t_l,t_u]$, with $0<t_l<t_u$, such that for all $t\in[t_l,t_u]$ equations (13) and (14) hold.*

Proof The proof is similar to that in [27, Proposition 2], we omit it here.

Proposition 3.2 *If the stepsize is obtained by Wolfe line search, then ρ_2^k in (12) is positive.*

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

3.3.2 Selection of H^k

To guarantee the positive definiteness of H^k, adopting the strategy proposed by Yuan and Stoer [38]:

$$H^k=\begin{pmatrix}\rho_1^k & \langle v_k,y^{k-1}\rangle\\ \langle v_k,y^{k-1}\rangle & \rho_2^k\end{pmatrix},$$

where

$$\rho_1^k = \frac{2\left\langle v_k, y^{k-1}\right\rangle^2}{\rho_2^k}.$$

Another powerful strategy proposed by Dai and Kou [9]:

$$\rho_1^k = \tau^k \frac{\left\|y^{k-1}\right\|^2}{\rho_2^k} \left\|v_k\right\|^2 (\tau^k > 1).$$

However, it is important to note that both methods lack global convergence in non-convex cases. In global convergence analysis we will require the *sufficient descent condition* [28]:

$$\mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) \leq -c \left\|v_k\right\|^2, \tag{15}$$

for some $c > 0$ and for all $k \geq 0$. Motivated by [26], we provide a sufficient condition on H^k to ensure that the obtained search direction d_k is a sufficient descent direction.

Proposition 3.3 *Let* $\{H^k\} \in \mathbb{R}^{2\times 2}$ *be the sequence of symmetric matrices in (8) and assume that there exist constants* $0 < c_1 \leq c_2$ *such that*

$$c_1 \leq \lambda_{\min}(D_k^{-1} H^k D_k^{-1}) \leq \lambda_{\max}(D_k^{-1} H^k D_k^{-1}) \leq c_2 \tag{16}$$

holds for all k*, where*

$$D_k = \begin{pmatrix} \|v_k\| & 0 \\ 0 & \|s_{k-1}\| \end{pmatrix}.$$

Let $d_k = \mu_k v_k + \nu_k s_{k-1}$*, where* $(\mu_k, \nu_k)^T$ *is the solution of (8). Then, the search direction* d_k *satisfies the following conditions:*

$$\mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) \leq -\frac{c_1}{2} \left\|d_k\right\|^2, \tag{17}$$

$$\mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) \leq -\min_{i\in[m]} (\frac{\alpha_i^k}{\bar{\alpha}_i^k})^2 \frac{\left\|v_k\right\|^2}{c_2}, \tag{18}$$

and

$$\frac{\min_{i\in[m]}(\alpha_i^k/\bar{\alpha}_i^k)^2}{c_2 \max_{i\in[m]} \alpha_i^k/\bar{\alpha}_i^k} \left\|v_k\right\| \leq \left\|d_k\right\| \leq \frac{2\max_{i\in[m]} \alpha_i^k/\bar{\alpha}_i^k}{c_1} \left\|v_k\right\|. \tag{19}$$

Proof From (11), we derive that

$$\begin{aligned} -\mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) &= \left\langle \begin{pmatrix} \mu_k \\ \nu_k \end{pmatrix}, H^k \begin{pmatrix} \mu_k \\ \nu_k \end{pmatrix} \right\rangle \\ &= \left\langle D_k \begin{pmatrix} \mu_k \\ \nu_k \end{pmatrix}, D_k^{-1} H^k D_k^{-1} D_k \begin{pmatrix} \mu_k \\ \nu_k \end{pmatrix} \right\rangle \\ &\geq c_1(\mu_k^2 \left\|v_k\right\|^2 + \nu_k^2 \left\|s_{k-1}\right\|^2) \\ &\geq \frac{c_1}{2} \left\|d_k\right\|^2. \end{aligned}$$

This implies inequality (17). We use the relation (16) to get

$$\begin{aligned}\theta(x^k) &= \min_{(\mu,\nu)^T\in\mathbb{R}^2} \max_{i\in[m]} \left\langle \begin{pmatrix} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k \right\rangle \\ \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, s_{k-1} \right\rangle \end{pmatrix}, \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle + \frac{1}{2}\left\langle \begin{pmatrix} \mu \\ \nu \end{pmatrix}, H^k \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle \\ &\le \min_{(\mu,\nu)^T\in\mathbb{R}^2} \max_{i\in[m]} \left\langle \begin{pmatrix} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k \right\rangle \\ \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, s_{k-1} \right\rangle \end{pmatrix}, \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle + \frac{c_2}{2}\left\langle D_k \begin{pmatrix} \mu \\ \nu \end{pmatrix}, D_k \begin{pmatrix} \mu \\ \nu \end{pmatrix} \right\rangle \\ &\le \min_{\mu\in\mathbb{R}} \max_{i\in[m]} \left\langle \frac{\nabla F_i(x^k)}{\bar{\alpha}_i^k}, v_k \right\rangle \mu + \frac{c_2}{2}\|v_k\|^2\mu^2 \\ &\le \min_{\mu\in\mathbb{R}} \max_{i\in[m]} -\frac{\alpha_i^k}{\bar{\alpha}_i^k}\|v_k\|^2\mu + \frac{c_2}{2}\|v_k\|^2\mu^2 \\ &= -\min_{i\in[m]} \left(\frac{\alpha_i^k}{\bar{\alpha}_i^k}\right)^2 \frac{\|v_k\|^2}{2c_2},\end{aligned}$$

where the second inequality is given by setting $\nu = 0$, and the second inequality is due to $\left\langle \nabla F_i(x^k), v_k \right\rangle \le -\alpha_i^k \|v_k\|^2$. Plugging the preceding bound into (10) gives inequality (18). By the definition of $\mathcal{D}_\alpha(x,d)$, we have

$$\begin{aligned}\mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) &= \max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \left\langle \frac{\nabla F_i(x^k)}{\alpha_i^k}, d_k \right\rangle \\ &\ge \max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \max_{i\in[m]} \left\langle \frac{\nabla F_i(x^k)}{\alpha_i^k}, d_k \right\rangle \\ &\ge \max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \left\langle \sum_{i\in[m]} \lambda_{BBi}^k \frac{\nabla F_i(x^k)}{\alpha_i^k}, d_k \right\rangle \\ &\ge \max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \langle -v_k, d_k \rangle \\ &\ge -\max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \|v_k\| \|d_k\|,\end{aligned} \tag{20}$$

where the first inequality is due to the fact that $\max_{i\in[m]} \left\langle \nabla F_i(x^k)/\alpha_i^k, d_k \right\rangle \le 0$. By substituting the latter bound into (17) and (18), respectively, we derive the relation (19).

Remark 3.1 If $m = 1$, by setting $\bar{\alpha}^k = \alpha^k = 1$, the relations (18) and (19) reduce to (4.17) and (4.18) in [26], respectively.

In addition to ensuring positive definiteness, the selected H^k should also capture the problem's local curvature information in the low-dimensional subspace. Therefore, we set

$$H^k = \begin{pmatrix} \rho_1^k & \left\langle v_k, y^{k-1} \right\rangle \\ \left\langle v_k, y^{k-1} \right\rangle & \rho_2^k \end{pmatrix}, \tag{21}$$

where

$$\rho_1^k = \begin{cases} \left\langle v_k, y_v^{k-1} \right\rangle, & \left\langle v_k, y_v^{k-1} \right\rangle > 0, \\ \|v_k\| \left\| y_v^{k-1} \right\|, & \text{otherwise}. \end{cases} \tag{22}$$

As described in [26], to guarantee H^k satisfies condition (16), we can proceed as follows:

Algorithm 1: `modified_Cholesky_factorization`

Data: D^k, $0 < c_1 \leq c_2$

Update H^k as (21)

Set $\hat{H}^k = D_k^{-1} H^k D_k^{-1}$

Compute a triangular matrix $L \in \mathbb{R}^{2\times 2}$:

$$L_{11} = \begin{cases} \sqrt{\hat{H}_{11}^k}, & \sqrt{\hat{H}_{11}^k} > c_1, \\ \sqrt{c_2}, & \text{otherwise.} \end{cases}$$

$$L_{21} = \frac{\hat{H}_{21}^k}{L_{11}},$$

and

$$L_{22} = \begin{cases} \sqrt{\hat{H}_{22}^k - L_{21}^2}, & \hat{H}_{22}^k - L_{21}^2 > c_1, \\ \sqrt{c_2}, & \text{otherwise.} \end{cases}$$

Compute $\hat{H}^k = LL^T$

Set $H^k = D_k \hat{H}^k D_k$

The subspace minimization Barzilai-Borwein descent method for MOPs is described as follows.

Algorithm 2: `subspace_minimization_Barzilai-Borwein_descent_method_for_MOPs`

Data: $x^0 \in \mathbb{R}^n$, $0 < c_1 \leq c_2$, $0 < \sigma_1 \leq \sigma_2$

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

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

  Update α_i^k as (3), $i \in [m]$

  Compute v_k and λ^k as the solution and dual solution of (2), respectively

  if $v_k = 0$ **then**

    return Pareto critical point x^k

  else

    if $k = 0$ **then**

      Set $d_k = v_k$, $\bar{\lambda}^k = \lambda^k$, $\bar{\alpha}^k = \alpha^k$

    else

      Update H^k by Algorithm 1

      Update $\bar{\alpha}_i^k$ as (9), $i \in [m]$

      Compute $(\mu_k, \nu_k)^T$ and $\bar{\lambda}^k$ as the solution and dual solution of (8), respectively

      Set $d_k = \mu_k v_k + \nu_k s_{k-1}$

    end

    Compute a stepsize t_k satisfies equations (13) and (14)

    $x^{k+1} := x^k + t_k d_k$

  end

end

4 Convergence Analysis

This section presents the convergence results for Algorithm 2. Notably, Algorithm 2 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 2 produces an infinite sequence of noncritical points.

4.1 Global Convergence

In this subsection, we analyze the global convergence of Algorithm 2 without making any convexity assumptions.

Theorem 4.1 *Suppose that Assumption 3.1 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 2. Then $\{x_k\}$ has at least one accumulation point, and every accumulation point $x^* \in \mathcal{L}_F(x^0)$ is a Pareto critical point.*

Proof We use the relation (13) to deduce that $\{F_i(x^k)\}$ is monotone decreasing and that

$$F_i(x^{k+1}) - F_i(x^k) \leq \alpha_{\min}\sigma_1 t_k \mathcal{D}_{\bar{\alpha}^k}(x^k, d_k). \tag{23}$$

It follows that $\{x^k\} \subset \mathcal{L}_F(x^0)$ and $\{x_k\}$ has at least one accumulation point x^*, namely, there exists an infinite index set $\mathcal{K}$ such that $\lim_{k\in\mathcal{K}} x^k = x^*$. From the compactness of $\mathcal{L}_F(x^0)$ and continuity of F, we deduce that $\{F(x^k)\}$ is bounded. This, together with the monotonicity of $\{F_i(x^k)\}$, indicates that $\{F(x^k)\}$ is a Cauchy sequence. Therefore, there exists a point F^* such that

$$\lim_{k\to\infty} F(x^k) = F^* = F(x^*).$$

Summing the inequality (23) from $k = 0$ to infinity and substituting the preceding limit, we have

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

Plugging relation (17) into the latter inequality gives

$$\sum_{k=0}^{\infty} t_k \left\|d_k\right\|^2 < \infty.$$

It follows that

$$\lim_{k\in\mathcal{K}} t_k d_k = 0. \tag{24}$$

We use relation (14) to get

$$(\sigma_2 - 1)\mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) \leq \mathcal{D}_{\bar{\alpha}^k}(x^k + t_k d_k, d_k) - \mathcal{D}_{\bar{\alpha}^k}(x^k, d_k).$$

Taking the limit on both sides, the latter inequality, together with (24) and the continuity of ∇F_i, implies

$$\lim_{k\in\mathcal{K}} \mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) = 0.$$

Plugging the above limit into (18) gives

$$\lim_{k\in\mathcal{K}} v_k = 0.$$

It follows by the [5, Lemma 5(d)] that x^* is a Pareto critical point.

4.2 Linear convergence

This subsection is devoted to the linear convergence of Algorithm 2. Before presenting the convergence result, we introduce the following error bound condition.

Definition 4.1 The vector-valued function F satisfies a global error bound, if there exists a constant κ such that

$$u_0(x) \leq \kappa \left\| v(x) \right\|^2, \ \forall x \in \mathbb{R}^n,$$

where

$$u_0(x) := \sup_{y\in\mathbb{R}^n} \min_{i\in[m]} \{F_i(x) - F_i(y)\}$$

is a merit function for (MOP) (see [36, Theorem 3.1]).

Remark 4.1 Since $\|v_k\|$ and $\left\|d^k_{SD}\right\|$ are equivalent, the definition 4.1 is equivalent to the multiobjective PL-inequality [36] for unconstrained multiobjective optimization problems. As a result, strong convexity of F is a sufficient condition for the definition 4.1.

To establish the linear convergence result of SMBBMO, we must first derive a lower bound for the stepsize t_k.

Assumption 4.1 *For each $i \in [m]$, the gradient ∇F_i is Lipschitz continuous with constant L_i.*

Lemma 4.1 *Suppose that Assumption 4.1 holds. If the stepsize t_k is obtained by Wolfe line search, then*

$$t_k \geq t_{\min} := \frac{(1-\sigma_2)c_1\alpha_{\min}}{2L_{\max}}, \tag{25}$$

where $L_{\max} := \max_{i\in[m]}\{L_i\}$.

Proof Using relation (14) and Assumption 4.1, we have

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

By substituting (17) into the above inequality, the desired result follows.

Next, we show the Q-linear convergence of $\{u_0(x^k)\}$.

Theorem 4.2 *Suppose that F satisfies definition 4.1 and Assumption 4.1 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 2. Then*

$$u_0(x^{k+1}) \leq (1-r)\, u_0(x^k),$$

where $r := \sigma_1 t_{\min}\alpha^3_{\min}/(c_2\kappa\alpha^2_{\max})$.

Proof Using (23) and (18), we have

$$\begin{aligned} F_i(x^{k+1}) - F_i(x^k) &\leq \alpha_{\min}\sigma_1 t_k \mathcal{D}_{\bar{\alpha}^k}(x^k, d_k) \\ &\leq -\alpha_{\min}\sigma_1 t_{\min} \min_{i\in[m]} (\frac{\alpha_i^k}{\bar{\alpha}_i^k})^2 \frac{\|v_k\|^2}{c_2} \\ &\leq -\sigma_1 t_{\min}\alpha_{\min}^3/(c_2\alpha_{\max}^2) \|v_k\|^2 \\ &\leq -\sigma_1 t_{\min}\alpha_{\min}^3/(c_2\kappa\alpha_{\max}^2) u_0(x^k), \end{aligned}$$

where the last inequality is due to the error bound. Denoting $r := \sigma_1 t_{\min}\alpha_{\min}^3/(c_2\kappa\alpha_{\max}^2)$, rearranging and taking the minimum and supremum with respect to $i \in [m]$ and $x \in \mathbb{R}^n$ on both sides, respectively, we obtain

$$\max_{x\in\mathbb{R}^n} \min_{i\in[m]} \{F_i(x^{k+1}) - F_i(x)\} \leq \max_{x\in\mathbb{R}^n} \min_{i\in[m]} \{F_i(x^k) - F_i(x)\} - ru_0(x^k).$$

Hence, the desired result follows.

5 Numerical results

In this section, we present numerical results to demonstrate the performance of SMBBMO for various problems. We also compare SMBBMO with Barzilai-Borwein descent method for MOPs (BBDMO) [5] and Barzilai-Borwein quasi-Newton method for MOPs (BBQNMO) [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, BBQNMO and SMBBMO, 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 [27], and set $\sigma_1 = 10^{-4}$, $\sigma_2 = 0.1$ in Wolfe line search. To ensure that the algorithms terminate after a finite number of iterations, for all tested algorithms we use the stopping criterion:

$$\theta(x) \geq -5 \times \texttt{eps}^{1/2},$$

where $\theta(x) = -1/2 \|v(x)\|^2$ for BBDMO and SMBBMO, and $\theta(x) = -1/2 \|d(x)\|^2_{B(x)}$ for BBQNMO, respectively, and $\texttt{eps} = 2^{-52} \approx 2.22 \times 10^{-16}$ is the machine precision. 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. Dual subproblems of different algorithms are efficiently solved by Frank-Wolfe method. The recorded averages from the 200 runs include the number of iterations, the number of function evaluations, and the CPU time.

5.1 Ordinary test problems

The tested algorithms are executed on several test problems, and the problem illustration is given in Table 1. 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.

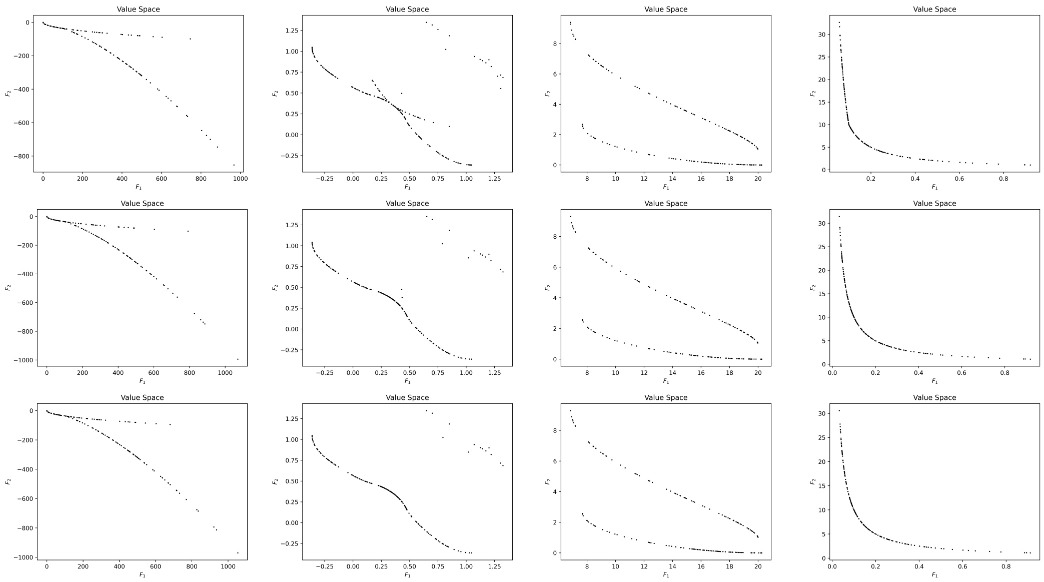
Table 1: Description of all test problems used in numerical experiments

Problem	n	m	x_L	x_U	Reference
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)	[25]
FDS	5	3	(-2,...,-2)	(2,...,2)	[14]
FF1	2	2	(-1,-1)	(1,1)	[25]
Hil1	2	2	(0,0)	(1,1)	[23]
Imbalance1	2	2	(-2,-2)	(2,2)	[5]
Imbalance2	2	2	(-2,-2)	(2,2)	[5]
LE1	2	2	(-5,-5)	(10,10)	[25]
PNR	2	2	(-2,-2)	(2,2)	[33]
VU1	2	2	(-3,-3)	(3,3)	[25]
WIT1	2	2	(-2,-2)	(2,2)	[37]
WIT2	2	2	(-2,-2)	(2,2)	[37]
WIT3	2	2	(-2,-2)	(2,2)	[37]
WIT4	2	2	(-2,-2)	(2,2)	[37]
WIT5	2	2	(-2,-2)	(2,2)	[37]
WIT6	2	2	(-2,-2)	(2,2)	[37]

Table 2: Number of average iterations (iter), number of average function evaluations (feval), and average CPU time (time(ms)) of BBDMO, BBQNMO, and SMBBMO implemented on different test problems

Problem	BBDMO			BBQNMO			SMBBMO		
	iter	feval	time	iter	feval	time	iter	feval	time
DD1	5.77	**5.91**	**1.36**	7.82	16.09	2.87	**5.38**	8.08	1.60
Deb	3.53	5.59	**0.96**	**3.17**	**4.51**	1.40	3.28	5.81	1.04
Far1	32.07	32.56	7.18	**6.94**	**16.11**	**2.74**	15.24	35.96	7.89
FDS	4.12	4.35	**2.60**	4.54	5.77	4.90	**3.83**	**4.23**	4.87
FF1	4.08	5.30	**0.63**	**3.37**	**5.12**	0.90	3.50	5.83	1.13
Hil1	9.19	9.96	1.46	**3.85**	**7.26**	**1.13**	6.34	10.91	2.41
Imbalance1	2.55	**3.48**	**0.40**	2.46	7.33	0.62	**2.00**	4.86	0.62
Imbalance2	**1.00**	**1.00**	0.27	**1.00**	**1.00**	0.29	**1.00**	**1.00**	**0.21**
LE1	3.61	**5.77**	**0.58**	3.78	5.93	0.90	**3.57**	7.85	1.11
PNR	3.30	**3.58**	**0.88**	3.38	4.40	0.73	**3.17**	4.28	0.89
VU1	13.68	13.73	1.86	**7.73**	**12.41**	**1.70**	11.49	16.47	3.32
WIT1	2.95	3.04	**0.42**	2.77	3.23	0.59	**2.54**	**2.91**	0.70
WIT2	3.27	3.37	**0.48**	3.09	3.23	0.68	**2.81**	**2.99**	0.76
WIT3	4.17	4.26	**0.59**	3.87	3.97	0.80	**3.52**	**3.77**	1.02
WIT4	4.33	4.38	**0.58**	4.08	4.15	0.84	**3.59**	**3.85**	1.00
WIT5	3.43	3.45	**0.50**	3.36	3.40	0.72	**2.94**	**3.04**	0.83
WIT6	**1.00**	**1.00**	**0.22**	**1.00**	**1.00**	0.24	**1.00**	**1.00**	0.23

For each test problem, Table 2 presents the average number of iterations (iter), average function evaluations (feval), and average CPU time (time(ms)) for the different algorithms. It is observed that BBQNMO and SMBBMO surpass BBDMO in terms of average iterations, suggesting their superior ability to capture the local geometry of the tested problems. Notably, SMBBMO demonstrates superior performance over BBQNMO, particularly when $n = 2$; thus, SMBBMO effectively captures the local geometry of the problems across the entire space. However, compared to BBDMO and BBQNMO, SMBBMO shows a relatively poorer performance in CPU time. This can be attributed to the well-conditioning of the test problems and the necessity to solve two subproblems in SMBBMO.



(a) DD1 (b) Hil1 (c) PNR (d) VU1

Fig. 1: Numerical results in value space obtained by BBDMO (**top**), BBQNMO (**middle**) and SMBBMO for problems DD1, Hil1, PNR, and VU1.

5.2 Quadratic ill-conditioned problems

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

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

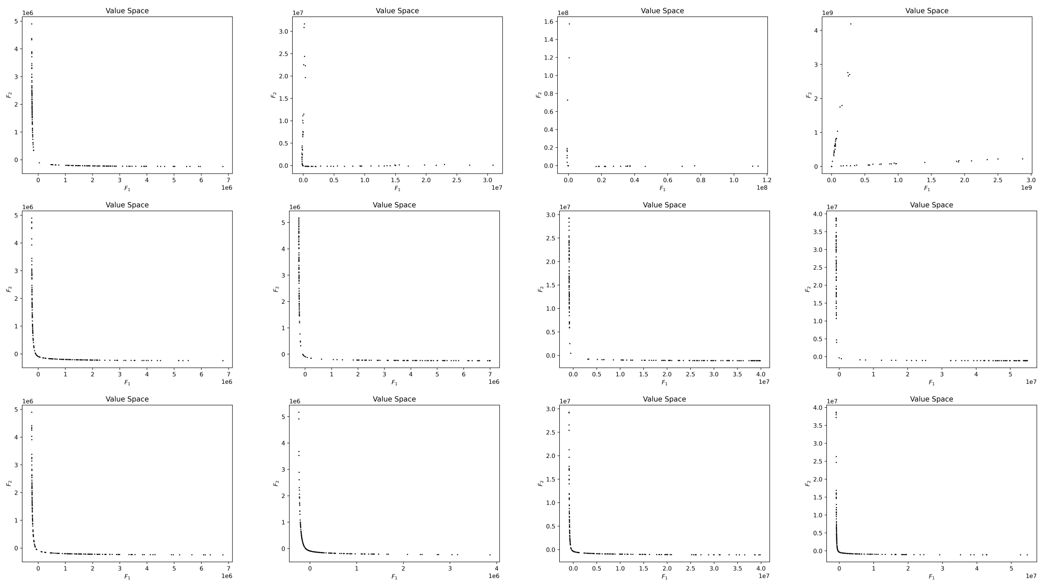
where A_i is a positive definite matrix. We set $A_i = H_i D_i H_i^T$, where H_i is a random orthogonal matrix and $D_i = Diag(d_i^1, d_i^2, ..., d_i^n)$ with $\max_j d_i^j / \min_j d_i^j = \kappa_i$. The problem illustration is given in Table 3. The second and third columns present the objective functions' dimension and condition numbers, respectively, while x_L and x_U represent the lower and upper bounds of the variables, respectively.

Table 3: Description of quadratic problems

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	1000	$(10^4, 10^4)$	1000[-1,...,-1]	1000[1,...,1]
QPh	1000	$(10^5, 10^5)$	1000[-1,...,-1]	1000[1,...,1]

Table 4: Number of average iterations (iter), number of average function evaluations (feval), and average CPU time (time(ms)) of BBDMO, BBQNMO, and SMBBMO implemented on quadratic problems

Problem	BBDMO			BBQNMO			SMBBMO		
	iter	feval	time	iter	feval	time	iter	feval	time
QPa	12.06	13.44	**1.38**	**9.55**	13.77	1.89	9.96	**10.65**	2.67
QPb	42.24	67.46	5.04	**20.16**	38.92	**4.32**	20.67	**31.60**	5.92
QPc	53.39	82.49	**8.47**	**34.59**	65.80	10.52	36.20	**42.68**	10.39
QPd	180.45	356.16	31.72	**42.81**	88.31	**13.32**	58.78	**81.57**	17.80
QPe	184.43	343.49	111.07	**64.94**	110.92	830.70	81.60	**87.49**	**47.68**
QPf	436.72	1168.17	432.60	**116.48**	279.83	1286.07	121.55	**203.98**	**80.80**
QPg	320.00	909.17	1164.93	157.15	511.41	8483.27	**154.84**	**189.42**	**468.84**
QPh	500.00	2856.25	3513.05	**262.81**	1106.15	15049.72	375.41	**785.56**	**1542.79**



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

Fig. 2: Numerical results in value space obtained by BBDMO (**top**), BBQNMO (**middle**) and SMBBMO for problems QPe, QPf, QPg, and QPh.

Table 4 illustrates the average number of iterations (iter), average number of function evaluations (feval), and average CPU time (time in milliseconds) obtained from 200 experimental runs for each quadratic problem. BBDMO, being a first-order method, exhibits competence in handling moderately ill-conditioned problems (QPb-e) owing to the Barzilai-Borwein rule, yet it struggles to converge within 500 iterations on extremely ill-conditioned problems (QPf-h). Conversely, for ill-conditioned and high-dimensional problems (QPe-h), SMBBMO demonstrates a notable superiority over BBQNMO in terms of CPU time efficiency. It is notable that SMBBMO shows promise in capturing the local curvature of ill-conditioned problems. To sum up, the primary experimental results underscore that SMBBMO achieves a faster convergence rate than BBDMO while maintaining a lower computational cost than BBQNMO.

6 Conclusions

In this paper, we introduce a novel subspace minimization Barzilai-Borwein method for MOPs, which outperforms BBDMO in terms of convergence rate while requiring lower computational resources compared to BBQNMO. We employ a modified Cholesky factorization to ensure global convergence of the proposed method in non-convex scenarios. Our numerical experiments demonstrate that SMBBMO exhibits promising performance for tackling large-scale and ill-conditioned MOPs.

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

- By selecting different subspaces, more historical iteration information (see [1, 8]) can be utilized to construct the subspace.
- By selecting different approximate models, SMCG with cubic regularization [39] can also be extended to MOPs.

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].

References

1. N. Andrei. An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization. *Numerical Algorithms*, 65:859–874, 2014.
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. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM Journal on Optimization*, 15(4):953–970, 2005.
4. 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.
5. 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.
6. 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.
7. J. Chen, L. P. Tang, and X. M. Yang. Preconditioned Barzilai-Borwein methods for multiobjective optimization problems. *optimization-online*, 2024.
8. W. Chen, X. M. Yang, and Y. Zhao. Memory gradient method for multiobjective optimization. *Applied Mathematics and Computation*, 443:127791, 2023.
9. Y. H. Dai and C. X. Kou. A Barzilai-Borwein conjugate gradient method. *Science China Mathematics*, 59:1511–1524, 2016.
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. Y. Elboulqe and M. El Maghri. An explicit spectral Fletcher–Reeves conjugate gradient method for bi-criteria optimization. *IMA Journal of Numerical Analysis*, page drae003, 2024.
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. E. H. Fukuda, L. M. Graña Drummond, and A. M. Masuda. A conjugate directions-type procedure for quadratic multiobjective optimization. *Optimization*, 71(2):419–437, 2022.
19. M. L. N. Gonçalves, F. S. Lima, and L. F. Prudente. Globally convergent Newton-type methods for multiobjective optimization. *Computational Optimization and Applications*, 83(2):403–434, 2022.
20. M. L. N. Gonçalves and L. F. Prudente. On the extension of the Hager–Zhang conjugate gradient method for vector optimization. *Computational Optimization and Applications*, 76(3):889–916, 2020.
21. 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.
22. Q. R. He, C. R. Chen, and S. J. Li. Spectral conjugate gradient methods for vector optimization problems. *Computational Optimization and Applications*, 86(2):457–489, 2023.
23. C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.
24. Q. J. Hu, L. P. Zhu, and Y. Chen. Alternative extension of the Hager–Zhang conjugate gradient method for vector optimization. *Computational Optimization and Applications*, pages 1–34, 2024.
25. 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.
26. M. Lapucci, G. Liuzzi, S. Lucidi, and M. Sciandrone. A globally convergent gradient method with momentum. *arXiv preprint arXiv.2403.17613*, 2024.
27. M. Lapucci and P. Mansueto. A limited memory quasi-Newton approach for multi-objective optimization. *Computational Optimization and Applications*, 85(1):33–73, 2023.
28. L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM Journal on Optimization*, 28(3):2690–2720, 2018.
29. R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
30. 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.
31. H. Mukai. Algorithms for multicriterion optimization. *IEEE Transactions on Automatic Control*, 25(2):177–186, 1980.
32. Ž. Povalej. Quasi-Newton's method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 255:765–777, 2014.

33. 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.
34. S. J. Qu, M. Goh, and F. T. Chan. Quasi-Newton methods for solving multiobjective optimization. *Operations Research Letters*, 39(5):397–399, 2011.
35. 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.
36. H. Tanabe, E. H. Fukuda, and N. Yamashita. New merit functions for multiobjective optimization and their properties. *Optimization*, pages 1–38, 2023.
37. K. Witting. *Numerical algorithms for the treatment of parametric multiobjective optimization problems and applications*. PhD thesis, Paderborn, Universität Paderborn, Diss., 2012, 2012.
38. Y. X. Yuan and J. Stoer. A subspace study on conjugate gradient algorithms. *Zeitschrift für Angewandte Mathematik und Mechanik*, 75(1):69–77, 1995.
39. T. Zhao, H. W. Liu, and Z. X. Liu. New subspace minimization conjugate gradient methods based on regularization model for unconstrained optimization. *Numerical Algorithms*, 87:1501–1534, 2021.