10.48047/jocaaa.2025.34.11.17

Barzilai–Borwein Directions in Restricted Subspaces for Vector Optimization

Salah Turki Mahgoob
Email: slaht956@gmail.com

Abstract

In recent years, nonlinear conjugate–gradient schemes have drawn growing interest in the context of multiobjective optimization. A central design goal has been to keep the role of the conjugacy parameters aligned with their well–studied single–objective analogues. Yet, for vector–valued objectives—even for quadratic models—the desirable "conjugacy" interpretation of successive search directions is not guaranteed, which obscures the meaning of the most recent direction and curtails practical use.

To clarify and exploit the information carried by the latest direction, we propose a new algorithm, the *Subspace Minimization Barzilai–Borwein method for Multiobjective Optimization* (SMBBMO). At each iteration, SMBBMO forms a two–dimensional subspace spanned by the previous search direction and the current Barzilai–Borwein (BB) descent direction, and then computes the next direction by exactly minimizing a preconditioned BB subproblem restricted to this subspace. To secure global convergence, we apply a modified Cholesky factorization to a suitably scaled (transformed) matrix that captures local curvature within the same two–dimensional subspace. Under standard assumptions, we establish global convergence as well as Q–linear convergence of the iterates. Numerical comparisons demonstrate that SMBBMO is effective and robust on large–scale and poorly conditioned instances.

Keywords: multiobjective optimization; subspace techniques; Barzilai–Borwein updates; global convergence

Introduction

An unconstrained multiobjective optimization problem (MOP) can be expressed as:

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

where $F:\mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable mapping. Such formulations are pervasive in engineering, economics, management science, and machine learning, where multiple

10.48047/jocaaa.2025.34.11.17

performance goals must be optimized simultaneously. In most practical contexts, obtaining a single solution that minimizes all objectives simultaneously is infeasible; therefore, the notion of Pareto optimality is used. A point is Pareto optimal (or efficient) if improving one objective necessarily worsens at least one of the others.

Over the last two decades, gradient-based algorithms for multiobjective optimization have gained substantial momentum. These approaches compute a descent direction by solving a subproblem and then perform a line search to guarantee adequate improvement across all objective functions. Foundational contributions were made by Mukai and by Fliege and Svaiter, who demonstrated that the multiobjective steepest descent direction reduces to the standard steepest descent direction in the single-objective setting. This observation inspired extensive research into the adaptation of classical numerical methods to the multiobjective framework.

Recently, Lucambio Pérez and Prudente extended the Wolfe line search and Zoutendijk condition to multiobjective problems, opening the door to nonlinear conjugate–gradient formulations in this setting. Such methods blend information from the current steepest descent direction and the previous search direction, aiming to preserve the conjugacy relationships familiar from single-objective methods such as those of Fletcher–Reeves, Polak–Ribiére–Polyak, Hestenes–Stiefel, Dai–Yuan, Conjugate Descent, Hager–Zhang, and Liu–Storey.

In single-objective optimization, the linear conjugate gradient method enjoys finite termination when applied to convex quadratic models due to its perfect conjugacy property. For vector-valued extensions, Fukuda and colleagues introduced a conjugate-directions variant that terminates finitely on strongly convex quadratic MOPs; however, this property does not extend to more general, non-quadratic settings. Moreover, even for quadratic instances, existing multiobjective conjugate–gradient formulations fail to ensure the conjugacy of search directions. In single-objective frameworks, losing this property limits the efficiency of nonlinear conjugate–gradient methods on large-scale non-quadratic problems. To mitigate this limitation, Yuan and Stoer developed the Subspace Minimization Conjugate Gradient (SMCG) method, which determines search directions by minimizing a model restricted to the two-dimensional subspace spanned by the latest gradient and previous direction. Within this reduced space, the conjugacy parameters are optimal relative to the local quadratic model, and the subproblem can be solved efficiently while preserving theoretical guarantees comparable to full-space methods.

Extending this subspace idea to multiobjective problems motivates our work. The

10.48047/jocaaa.2025.34.11.17

essential design questions are how to define the subspace and how to construct effective low-dimensional approximate models that capture local curvature.

In this study, we propose the *Subspace Minimization Barzilai–Borwein Method for Multiobjective Optimization* (SMBBMO). The framework is characterized by the following design features:

• **Subspace construction:** To address the imbalance among objectives—which can slow convergence in steepest descent schemes—we build a two-dimensional subspace spanned by the current Barzilai–Borwein descent direction and the previous search direction. Formally, the update is defined as

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

where $(\mu_k, \nu_k)^\top \in \mathbb{R}^2$ and v_k denotes the Barzilai–Borwein descent direction at iteration k. This recurrence shares similarities with spectral conjugate–gradient formulas but leverages richer spectral information from the Barzilai–Borwein mechanism, unlike earlier multiobjective spectral approaches that restrict the subspace to the steepest descent and previous direction.

• **Approximate model:** To balance computational cost and curvature accuracy, we employ a preconditioned Barzilai–Borwein subproblem as the local model. Finite differences of gradients approximate matrix–vector products, representing curvature within the chosen two-dimensional subspace by a compact 2×2 matrix. To ensure global convergence, we adopt a modified Cholesky factorization of a transformed scaling matrix, thereby incorporating curvature information in a numerically stable way.

The SMBBMO algorithm seeks to achieve faster convergence than the Barzilai–Borwein descent method while retaining a significantly lower computational overhead than fully preconditioned alternatives.

The paper is structured as follows. Section 2 presents the necessary notations and preliminary definitions. Section 3 introduces the SMBBMO algorithm, detailing its subspace and model selection. Section 4 establishes global and Q-linear convergence results. Section 5 reports numerical experiments illustrating the method's performance. Concluding remarks appear in

10.48047/jocaaa.2025.34.11.17

Section 6.

Preliminaries

Throughout this study, we work in the Euclidean space $\mathbb{R}^n$ equipped with the standard inner product $\langle\cdot,\cdot\rangle$ and the corresponding norm $\|\cdot\|$. Let $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ denote the sets of symmetric positive semidefinite and positive definite matrices in $\mathbb{R}^{n\times n}$, respectively. For any mapping $F:\mathbb{R}^n \to \mathbb{R}^m$, we denote by $JF(x) \in \mathbb{R}^{m\times n}$ its Jacobian at x, by $\nabla F_i(x) \in \mathbb{R}^n$ the gradient of the ith component, and by $\nabla^2 F_i(x) \in \mathbb{R}^{n\times n}$ its Hessian.

Given a symmetric positive definite matrix H, we define the H-norm as

$$\| x \|_H = \sqrt{\langle x, Hx\rangle}.$$

For notational simplicity, we set $[m] := \{1,2,\dots,m\}$ and define the m-dimensional simplex as

$$\Delta_m := \{\lambda \in \mathbb{R}^m : \textstyle\sum_{i\in[m]} \lambda_i = 1,\ \lambda_i \geq 0,\ i \in [m]\}.$$

We introduce partial order relations on $\mathbb{R}^m$ and $\mathbb{S}^n$ by

$$u \preceq v\ (\prec v) \Leftarrow v - u \in \mathbb{R}^m_+\ (\mathbb{R}^m_{++}), \qquad U \preceq V\ (\prec V) \Leftarrow V - U \in \mathbb{S}^n_+\ (\mathbb{S}^n_{++}).$$

Pareto Optimality Concepts

We now recall several optimality notions commonly used in multiobjective optimization.

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

A vector $x^* \in \mathbb{R}^n$ is a *weak Pareto solution* to (MOP) if there is no $x \in \mathbb{R}^n$ such that $F(x) \prec F(x^*)$.

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

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

where $range(JF(x^*))$ denotes the image of the Jacobian matrix $JF(x^*)$.

Clearly, every Pareto solution is also a weak Pareto solution. The following result summarizes the relationships among these optimality concepts.

[[?, Theorem 3.1]] The following statements hold: [(i)]

1. Every weak Pareto solution of (MOP) is a Pareto critical point.
2. If all component functions F_i are convex, every Pareto critical point is a weak

10.48047/jocaaa.2025.34.11.17

Pareto solution.

3. If all F_i are strictly convex, every Pareto critical point is a Pareto solution.

Subspace Minimization Barzilai–Borwein Descent Method for MOPs

We consider the descent direction at iteration k defined by the subproblem

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

where $q_i^k(\cdot)$ represents a local quadratic approximation of F_i around x_k, and Ω_k is a selected low-dimensional subspace. The crucial design issues are how to construct Ω_k and how to generate efficient local models q_i^k.

1. Choice of Subspace

In nonlinear conjugate–gradient methods, each new direction is generated from a combination of the current steepest descent direction and the previous search direction. To maintain a similar spirit, we define for $k > 1$:

$$\Omega_k = Span\{v_k, d_{k-1}\}.$$

The choice of v_k is essential. Since steepest descent methods often require very small stepsizes due to imbalance among objectives, we instead adopt the Barzilai–Borwein (BB) descent direction at x_k, given by

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

where $\alpha^k \in \mathbb{R}^m_{++}$ is determined according to the Barzilai–Borwein spectral rule:

$$\alpha_i^k = \begin{cases} \max(\alpha_{\min}, \min(\frac{\langle s_{k-1}, y_{i,k-1}\rangle}{\|s_{k-1}\|^2}, \alpha_{\max})), & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle > 0, \\ \max(\alpha_{\min}, \min(\frac{\|y_{i,k-1}\|}{\|s_{k-1}\|}, \alpha_{\max})), & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle < 0, \\ \alpha_{\min}, & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle = 0, \end{cases}$$

for all $i \in [m]$, with $s_{k-1} = x_k - x_{k-1}$ and $y_{i,k-1} = \nabla F_i(x_k) - \nabla F_i(x_{k-1})$. The constants $\alpha_{\min}$ and $\alpha_{\max}$ bound the spectral parameters from below and above, respectively, to ensure numerical stability.

2. Selection of the Local Model

10.48047/jocaaa.2025.34.11.17

Quadratic surrogates are a natural choice in iterative optimization, as they often capture the objective's behavior in a small neighborhood around the current iterate. To balance per-iteration effort with richer curvature usage, we adopt the preconditioned BB-type model of [?]:

$$q_i^k(d) := \frac{\langle \nabla F_i(x_k), d\rangle}{\bar{\alpha}_i^k} + \frac{1}{2} \parallel d \parallel_{B_k}^2, \tag{3}$$

where $\bar{\alpha}^k \in \mathbb{R}_{++}^m$ is defined componentwise by

$$\bar{\alpha}_i^k = \begin{cases} \max(\alpha_{\min}, \min(\frac{\langle s_{k-1}, y_{i,k-1}\rangle}{\|s_{k-1}\|_{B_k}^2}, \alpha_{\max})), & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle > 0, \\ \max(\alpha_{\min}, \min(\frac{\|y_{i,k-1}\|}{\|B_k s_{k-1}\|}, \alpha_{\max})), & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle < 0, \\ \alpha_{\min}, & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle = 0, \end{cases}$$

with $s_{k-1} := x_k - x_{k-1}$ and $y_{i,k-1} := \nabla F_i(x_k) - \nabla F_i(x_{k-1})$. Here $B_k \in \mathbb{S}_{++}^n$ is a positive definite scaling (preconditioner), and the induced norm is $\parallel d \parallel_{B_k} := \sqrt{\langle d, B_k d\rangle}$.

From a preconditioning viewpoint, it is beneficial to let B_k mimic an aggregated (scaled) Hessian of the vector objective, namely

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

where $\bar{\lambda}^{k-1} \in \Delta_m$ denotes the dual weights associated with the subspace subproblem at x_{k-1}.

3. SMBB Descent Direction via Two-Dimensional Subspace

Recall that $s_{k-1} = x_k - x_{k-1} = t_{k-1} d_{k-1}$. Restricting (3) to the plane spanned by the current BB descent vector v_k and the previous displacement s_{k-1}, we parameterize d as $d = \mu v_k + \nu s_{k-1}$ with $(\mu, \nu)^\top \in \mathbb{R}^2$. The subspace problem

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

then becomes the 2-D optimization

$$\min_{(\mu,\nu)\in\mathbb{R}^2} \max_{i\in[m]} \left\{ \left\langle \frac{\nabla F_i(x_k)}{\bar{\alpha}_i^k}, \mu v_k + \nu s_{k-1} \right\rangle + \frac{1}{2} \begin{bmatrix} \mu \\ \nu \end{bmatrix}^\top \underbrace{\begin{bmatrix} \langle v_k, B_k v_k\rangle & \langle v_k, B_k s_{k-1}\rangle \\ \langle s_{k-1}, B_k v_k\rangle & \langle s_{k-1}, B_k s_{k-1}\rangle \end{bmatrix}}_{=:H_k} \begin{bmatrix} \mu \\ \nu \end{bmatrix} \right\}. \tag{4}$$

To avoid forming B_k explicitly, we approximate its action by finite differences of gradients using the aggregated curvature weights:

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

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

Define the aggregated curvature surrogates

10.48047/jocaaa.2025.34.11.17

$$y_{k-1} := \sum_{i\in[m]} \bar{\lambda}_i^{k-1}\, \bar{\alpha}_i^{k-1}(\nabla F_i(x_k) - \nabla F_i(x_{k-1})), \qquad y_{k-1}^{v} :=$$

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

and the 2×2 matrix

$$H_k \approx \begin{bmatrix} \rho_k^{(1)} & \langle v_k, y_{k-1}\rangle \\ \langle v_k, y_{k-1}\rangle & \rho_k^{(2)} \end{bmatrix}, \qquad \rho_k^{(1)} \approx \langle v_k, y_{k-1}^{v}\rangle, \quad \rho_k^{(2)} \approx \langle s_{k-1}, y_{k-1}\rangle. \tag{5}$$

The SMBB search direction is then $d_k = \mu_k v_k + \nu_k s_{k-1}$, where (μ_k, ν_k) solves (4). In this subproblem, we also update the spectral parameters using the reduced-space curvature,

$$\bar{\alpha}_i^k = \begin{cases} \max(\alpha_{\min}, \min(\frac{\langle s_{k-1}, y_{i,k-1}\rangle}{\rho_k^{(2)}}, \alpha_{\max})), & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle > 0, \\ \max(\alpha_{\min}, \min(\frac{\|y_{i,k-1}\|}{\|y_{k-1}\|}, \alpha_{\max})), & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle < 0, \\ \alpha_{\min}, & \text{if} \langle s_{k-1}, y_{i,k-1}\rangle = 0. \end{cases}$$

For later use, define the value of the subspace model and a BB-scaled directional measure:

$$\vartheta(x_k) := \min_{(\mu,\nu)\in\mathbb{R}^2} \max_{i\in[m]} \left\{ \left\langle \frac{\nabla F_i(x_k)}{\bar{\alpha}_i^k}, \mu v_k + \nu s_{k-1} \right\rangle + \frac{1}{2} \begin{bmatrix} \mu \\ \nu \end{bmatrix}^{\top} H_k \begin{bmatrix} \mu \\ \nu \end{bmatrix} \right\}, \qquad D_\alpha(x, d) :=$$

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

Dual viewpoint and KKT relations.

Introducing an epigraph variable $t \in \mathbb{R}$, (4) is equivalent to the smooth QP

$$\begin{aligned} \min_{t\in\mathbb{R},(\mu,\nu)\in\mathbb{R}^2} \quad & t + \frac{1}{2} \begin{bmatrix} \mu \\ \nu \end{bmatrix}^{\top} H_k \begin{bmatrix} \mu \\ \nu \end{bmatrix} \\ \text{s.t.} \quad & \left\langle \frac{\nabla F_i(x_k)}{\bar{\alpha}_i^k}, \mu v_k + \nu s_{k-1} \right\rangle \le t, \quad i \in [m]. \end{aligned} \tag{6}$$

As in [?], the dual of (6) can be solved efficiently, yielding the optimal multipliers $\bar{\lambda}^k \in \Delta_m$. From the KKT conditions one obtains

$$\vartheta(x_k) = \frac{1}{2} D_{\bar{\alpha}^k}(x_k, d_k), \qquad D_{\bar{\alpha}^k}(x_k, d_k) = -\begin{bmatrix} \mu_k \\ \nu_k \end{bmatrix}^{\top} H_k \begin{bmatrix} \mu_k \\ \nu_k \end{bmatrix}. \tag{7}$$

Descent requirements.

To guarantee that d_k is a genuine descent direction for the max-model, we need

$$\rho_k^{(2)} > 0 \quad \text{and} \quad H_k \in \mathbb{S}_{++}^2.$$

In the next section we address how to enforce these properties (e.g., by a modified Cholesky procedure on a transformed scaling), thereby ensuring well-posedness and global convergence of the SMBB step.

10.48047/jocaaa.2025.34.11.17

1. Choice of $\rho_k^{(2)}$rho_k^2

In nonconvex settings it may occur that $\langle s_{k-1}, y_{k-1}\rangle \le 0$. We therefore define

$$\rho_k^{(2)} = \begin{cases} \langle s_{k-1}, y_{k-1}\rangle, & \text{if} \langle s_{k-1}, y_{k-1}\rangle > 0, \\ D_{\bar{\alpha}^{k-1}}(x_k, s_{k-1}) - \sum_{i\in[m]} \bar{\lambda}_i^{k-1} \langle \frac{\nabla F_i(x_{k-1})}{\bar{\alpha}_i^{k-1}}, s_{k-1}\rangle, & \text{otherwise,} \end{cases} \tag{8}$$

which matches (8) with the notation used previously.

To enforce $\rho_k^{(2)} > 0$, we employ a Wolfe-type line search: find $t > 0$ such that

$$\frac{F_i(x_k + t d_k) - F_i(x_k)}{\bar{\alpha}_i^k} \le \sigma_1 \, t \, D_{\bar{\alpha}^k}(x_k, d_k), \quad \forall i \in [m], \tag{9}$$

$$D_{\bar{\alpha}^k}(x_k + t d_k, d_k) \ge \sigma_2 \, D_{\bar{\alpha}^k}(x_k, d_k), \tag{10}$$

with constants $0 < \sigma_1 \le \sigma_2 < 1$.

For any starting point $x_0 \in \mathbb{R}^n$, the level set $L_F(x_0) := \{x \in \mathbb{R}^n : F(x) \preceq F(x_0)\}$ is compact.

Suppose the above assumption holds and d_k is a descent direction. If $0 < \sigma_1 \le \sigma_2 < 1$, then there exist $0 < t_\ell < t_u$ such that for all $t \in [t_\ell, t_u]$, the conditions (9)–(10) are satisfied.

(The proof follows the same lines as [?, Proposition~2] and is omitted.)

If the step length t is chosen by the Wolfe procedure (9)–(10), then the quantity $\rho_k^{(2)}$ defined in (8) is strictly positive.

(The claim is immediate from the Wolfe conditions and we do not reproduce the details.)

2. Choice of H_kHk

To secure $H_k \in \mathbb{S}_{++}^2$, one classic option (Yuan–Stoer [?]) is

$$H_k = \begin{bmatrix} \rho_k^{(1)} & \langle v_k, y_{k-1}\rangle \\ \langle v_k, y_{k-1}\rangle & \rho_k^{(2)} \end{bmatrix}, \qquad \rho_k^{(1)} = \frac{2\,\langle v_k, y_{k-1}\rangle^2}{\rho_k^{(2)}}.$$

Another effective alternative due to Dai–Kou [?] is

$$\rho_k^{(1)} = \tau_k \, \frac{\|y_{k-1}\|^2}{\|v_k\|^2} \, \rho_k^{(2)}, \qquad \tau_k > 1.$$

However, neither scheme alone guarantees global convergence for nonconvex problems. In our analysis we will invoke a uniform *sufficient descent* requirement (cf. [?]):

$$D_{\bar{\alpha}^k}(x_k, d_k) \le -c \, \| v_k \|^2 \quad \text{forsome} c > 0 \text{andall} k \ge 0. \tag{11}$$

Inspired by [?], we now state a matrix condition that ensures (11) holds for the solution of the

10.48047/jocaaa.2025.34.11.17

2-D subproblem.

Let $\{H_k\} \subset \mathbb{R}^{2\times 2}$ be the sequence of symmetric matrices in the subproblem (4). Suppose there exist constants $0 < c_1 \leq c_2$ such that, for all k,

$$c_1 \;\leq\; \lambda_{\min}(D_k^{-1}H_kD_k^{-1}) \;\leq\; \lambda_{\max}(D_k^{-1}H_kD_k^{-1}) \;\leq\; c_2, \tag{12}$$

where

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

Let $d_k = \mu_k v_k + \nu_k s_{k-1}$ be the optimizer of the 2-D subproblem. Then the following bounds hold:

$$D_{\bar{\alpha}^k}(x_k, d_k) \;\leq\; -\frac{c_1}{2} \| d_k \|^2, \tag{13}$$

$$D_{\bar{\alpha}^k}(x_k, d_k) \;\leq\; -\frac{1}{c_2} \min_{i\in[m]} \left(\frac{\alpha_i^k}{\bar{\alpha}_i^k}\right)^2 \| v_k \|^2, \tag{14}$$

and, consequently,

$$\min_{i\in[m]} \left(\frac{\alpha_i^k}{\bar{\alpha}_i^k}\right)^2 \frac{\|v_k\|}{c_2} \;\leq \| d_k \| \leq\; \frac{2}{c_1} \max_{i\in[m]} \left(\frac{\alpha_i^k}{\bar{\alpha}_i^k}\right) \| v_k \|. \tag{15}$$

Proof. From the KKT identity (7), we have

$$-D_{\bar{\alpha}^k}(x_k, d_k) \;=\; \begin{bmatrix}\mu_k \\ \nu_k\end{bmatrix}^{\top} H_k \begin{bmatrix}\mu_k \\ \nu_k\end{bmatrix} \;=\; \langle D_k \begin{bmatrix}\mu_k \\ \nu_k\end{bmatrix},\; (D_k^{-1}H_kD_k^{-1})\, D_k \begin{bmatrix}\mu_k \\ \nu_k\end{bmatrix}\rangle \;\geq\; c_1$$

$$(\mu_k^2 \| v_k \|^2 + \nu_k^2 \| s_{k-1} \|^2) \;\geq\; \frac{c_1}{2} \| d_k \|^2, \tag{16}$$

which yields (13). Using (12), we bound the model value

$$\vartheta(x_k) \;\leq\; \min_{(\mu,\nu)\in\mathbb{R}^2} \max_{i\in[m]} \left\{ \langle \frac{\nabla F_i(x_k)}{\bar{\alpha}_i^k}, \mu v_k + \nu s_{k-1} \rangle + \frac{c_2}{2} (\mu^2 \| v_k \|^2 + \nu^2 \| s_{k-1} \|^2) \right\} \;\leq$$

$$\min_{\mu\in\mathbb{R}} \max_{i\in[m]} \left\{ \langle \frac{\nabla F_i(x_k)}{\bar{\alpha}_i^k}, v_k \rangle \mu + \frac{c_2}{2} \| v_k \|^2 \mu^2 \right\}.$$

Since $\langle \nabla F_i(x_k), v_k \rangle \leq -\alpha_i^k \| v_k \|^2$, the right-hand side is bounded by

$$-\frac{1}{2c_2} \min_{i\in[m]} \left(\frac{\alpha_i^k}{\bar{\alpha}_i^k}\right)^2 \| v_k \|^2.$$

Combining with $\vartheta(x_k) = \frac{1}{2} D_{\bar{\alpha}^k}(x_k, d_k)$ from (7) gives (14). Finally, using

$$D_{\bar{\alpha}^k}(x_k, d_k) = \max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \langle \frac{\nabla F_i(x_k)}{\alpha_i^k}, d_k \rangle \;\geq\; \max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \langle -v_k, d_k \rangle \;\geq\; -\max_{i\in[m]} \frac{\alpha_i^k}{\bar{\alpha}_i^k} \| v_k \| \| d_k \|, \tag{17}$$

and combining (17) with (13) and (14) leads to the two-sided norm estimate (15).

Remark :

10.48047/jocaaa.2025.34.11.17

When $m = 1$ and we fix $\bar{\alpha}_k = \alpha_k = 1$, the bounds given earlier (the counterparts of (14) and (15)) specialize exactly to (4.17) and (4.18) in [1], respectively.

Besides positive definiteness, the 2×2 matrix H_k should reflect the local curvature on the chosen plane. We therefore adopt the symmetric form

$$H_k = \begin{bmatrix} \rho_k^{(1)} & \langle v_k, y_{k-1} \rangle \\ \langle v_k, y_{k-1} \rangle & \rho_k^{(2)} \end{bmatrix}, \tag{18}$$

with the diagonal entry $\rho_k^{(1)}$ chosen by

$$\rho_k^{(1)} = \begin{cases} \langle v_k, y_{k-1}^v \rangle, & \text{if} \langle v_k, y_{k-1}^v \rangle > 0, \\ \| v_k \| \| y_{k-1}^v \|, & \text{otherwise}, \end{cases} \tag{19}$$

where y_{k-1} and y_{k-1}^v are the aggregated curvature surrogates defined previously.

As suggested in [2], to enforce the spectral bounds in (12) we apply a scaled modified Cholesky step to H_k.

[H]

Scaled modified Cholesky (size 2×2)

Input: diagonal scaling $D_k = \text{diag}(\| v_k \|, \| s_{k-1} \|)$, thresholds $0 < c_1 \leq c_2$

Steps:

1. Build H_k via (18)–(19).
2. Form the scaled matrix $\widehat{H}_k := D_k^{-1} H_k D_k^{-1}$.
3. Compute a lower–triangular $L \in \mathbb{R}^{2\times 2}$:

$$L_{11} = \begin{cases} \sqrt{\widehat{H}_{k,11}}, & \widehat{H}_{k,11} > c_1, \\ \sqrt{c_2}, & \text{otherwise}, \end{cases} \qquad L_{21} = \frac{\widehat{H}_{k,21}}{L_{11}},$$

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

4. Set $\widehat{H}_k := LL^{\mathsf{T}}$ and finally $H_k := D_k \, \widehat{H}_k \, D_k$.

10.48047/jocaaa.2025.34.11.17

SMBB algorithm (subspace BB for MOPs). [H]

Subspace Minimization Barzilai–Borwein for Multiobjective Problems

Input: $x_0 \in \mathbb{R}^n$, parameters $0 < c_1 \leq c_2,\ 0 < \sigma_1 \leq \sigma_2 < 1$

Initialization: pick x_{-1} near x_0

1. **for** $k = 0,1,2,\ldots$ **do**

   (a) Update the BB spectral vector α^k componentwise as in (2) (for all $i \in [m]$).

   (b) Solve the BB descent subproblem (2) to obtain v_k and its dual weights λ^k.

   (c) **if** $v_k = 0$ **then** return x_k (Pareto critical point).

   (d) **if** $k = 0$ **then** set $d_k := v_k,\ \bar{\lambda}^k := \lambda^k,\ \bar{\alpha}^k := \alpha^k$.

   (e) **else**

      i. Build H_k via Algorithm .

      ii. Update the reduced-space spectral parameters $\bar{\alpha}^k$ using the rule in (8)–(10).

      iii. Solve the 2-D subproblem (4) (and its dual) to get (μ_k, ν_k) and $\bar{\lambda}^k$.

      iv. Set $d_k := \mu_k v_k + \nu_k s_{k-1}$, where $s_{k-1} = x_k - x_{k-1}$.

   (f) Compute a stepsize t_k satisfying the Wolfe conditions (9)–(10).

   (g) Update $x_{k+1} := x_k + t_k d_k$.

2. **end for**

Convergence Analysis

This section establishes the convergence properties of Algorithm 4. The algorithm either terminates in a finite number of iterations at a Pareto critical point or generates an infinite sequence of noncritical iterates. In the following analysis, we focus on the latter case, assuming that an infinite sequence $\{x_k\}$ of noncritical points is produced.

1. Global Convergence

We first demonstrate that every accumulation point of the generated sequence is Pareto critical, without assuming convexity.

Suppose Assumption 3.1 holds and let $\{x_k\}$ be the sequence produced by Algorithm 4. Then $\{x_k\}$ has at least one accumulation point, and every such point $x^* \in L_F(x_0)$ is a Pareto

10.48047/jocaaa.2025.34.11.17

critical point.

Proof. From condition (9), each objective sequence $\{F_i(x_k)\}$ is nonincreasing, and

$$F_i(x_{k+1}) - F_i(x_k) \leq \alpha_{\min}\, \sigma_1\, t_k\, D_{\overline{\alpha}^k}(x_k, d_k). \tag{20}$$

Hence, $\{x_k\} \subset L_F(x_0)$, which is compact by Assumption 3.1, implying that $\{x_k\}$ admits an accumulation point x^*. The continuity of F guarantees that $\{F(x_k)\}$ is bounded. Because each component sequence is monotone and bounded, $\{F(x_k)\}$ is Cauchy and convergent to some $F^* = F(x^*)$.

Summing (20) over k gives

$$-\sum_{k=0}^{\infty} \alpha_{\min}\sigma_1 t_k D_{\overline{\alpha}^k}(x_k, d_k) \;\leq\; F_i(x_0) - F_i^* \;<\; \infty.$$

Substituting inequality (13) yields

$$\sum_{k=0}^{\infty} t_k \;\|\, d_k \,\|^2 \;<\; \infty,$$

implying

$$\lim_{k\to\infty} t_k d_k = 0. \tag{21}$$

From the curvature condition (10), we obtain

$$(\sigma_2 - 1)\, D_{\overline{\alpha}^k}(x_k, d_k) \;\leq\; D_{\overline{\alpha}^k}(x_k + t_k d_k, d_k) - D_{\overline{\alpha}^k}(x_k, d_k).$$

Taking limits and using (21) together with the continuity of ∇F_i gives

$$\lim_{k\to\infty} D_{\overline{\alpha}^k}(x_k, d_k) = 0.$$

By inequality (14), this implies $\lim_{k\to\infty} v_k = 0$, and hence, by [4, Lemma~5(d)], the limit x^* is Pareto critical.

2. Linear Convergence

We now establish a global linear rate under a standard error bound assumption.

The mapping F satisfies a *global error bound* if there exists a constant $\kappa > 0$ such that

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

where

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

10.48047/jocaaa.2025.34.11.17

Since $\| v_k \|$ and $\| d_k^{SD} \|$ (the steepest-descent step) are equivalent, Definition 4.2 coincides with the multiobjective Polyak–Łojasiewicz inequality of [3]. In particular, strong convexity of all F_i implies this property.

We first establish a uniform lower bound for the step size t_k.

For each $i \in [m]$, ∇F_i is Lipschitz continuous with constant L_i.

Under Assumption 4.2, if t_k satisfies the Wolfe conditions, then

$$t_k \geq t_{\min} := \frac{(1-\sigma_2)\, c_1\, \alpha_{\min}}{2\, L_{\max}}, \qquad L_{\max} := \max_{i\in[m]} L_i. \tag{22}$$

Proof. Using (10) and the Lipschitz continuity of ∇F_i, we have

$$(\sigma_2 - 1)\, D_{\bar{\alpha}^k}(x_k, d_k) \leq D_{\bar{\alpha}^k}(x_k + t_k d_k, d_k) - D_{\bar{\alpha}^k}(x_k, d_k)$$

$$\leq \max_{i\in[m]} \langle \frac{\nabla F_i(x_k + t_k d_k) - \nabla F_i(x_k)}{\bar{\alpha}_i^k}, d_k \rangle \leq \max_{i\in[m]} \frac{L_i}{\bar{\alpha}_i^k}\, t_k \| d_k \|^2 \leq \frac{L_{\max}}{\alpha_{\min}}\, t_k \| d_k \|^2.$$

Substituting the lower bound (13) for $D_{\bar{\alpha}^k}(x_k, d_k)$ yields the inequality (22).

We now prove the main rate result.

Suppose that F satisfies the global error bound (Definition 4.2) and Assumption 4.2 holds. Let $\{x_k\}$ be the sequence generated by Algorithm 4. Then

$$u_0(x_{k+1}) \leq (1 - r)\, u_0(x_k),$$

where

$$r := \frac{\sigma_1\, t_{\min}\, \alpha_{\min}^3}{c_2\, \kappa\, \alpha_{\max}^2}. \tag{23}$$

Proof. Combining the descent inequality (20) with (14) gives

$$F_i(x_{k+1}) - F_i(x_k) \leq \alpha_{\min} \sigma_1 t_k D_{\bar{\alpha}^k}(x_k, d_k) \leq -\frac{\alpha_{\min} \sigma_1 t_{\min}}{c_2} \min_{i\in[m]} \left(\frac{\alpha_i^k}{\bar{\alpha}_i^k}\right)^2 \| v_k \|^2 \leq$$

$$-\frac{\sigma_1 t_{\min}\, \alpha_{\min}^3}{c_2 \alpha_{\max}^2} \| v_k \|^2.$$

Using the global error bound $u_0(x_k) \leq \kappa \| v_k \|^2$ and defining r as in (23), we obtain

$$F_i(x_{k+1}) - F_i(x_k) \leq -r\, u_0(x_k).$$

Taking $\min_{i\in[m]}$ and $\sup_{x\in\mathbb{R}^n}$ on both sides yields

$$u_0(x_{k+1}) \leq (1 - r)\, u_0(x_k),$$

completing the proof.

Numerical Experiments

10.48047/jocaaa.2025.34.11.17

We now assess the practical behavior of the proposed SMBBMO scheme on a collection of benchmark instances and compare it against two state-of-the-art baselines: the Barzilai–Borwein descent method for multiobjective problems (BBDMO) [6] and the Barzilai–Borwein quasi-Newton variant (BBQNMO) [?]. All codes were written in Python 3.7 and run on a laptop equipped with an Intel Core i7-11390H CPU (3.40 GHz) and 16 GB RAM.

Unless stated otherwise, the spectral truncation parameters for all three methods were fixed to

$$\alpha_{\min} = 10^{-3}, \qquad \alpha_{\max} = 10^{3}.$$

A Wolfe backtracking line search (as in Algorithm 3 of [?]) was used with

$$\sigma_1 = 10^{-4}, \qquad \sigma_2 = 0.1.$$

To guarantee termination, each run stops if

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

with machine precision $\varepsilon = 2^{-52} \approx 2.22 \times 10^{-16}$. Here, for BBDMO and SMBBMO we take $\theta(x) = -\frac{1}{2} \parallel v(x) \parallel^2$, while for BBQNMO we use $\theta(x) = -\frac{1}{2} \parallel d(x) \parallel^2_{B(x)}$. We also cap the iteration count at 500. For fairness, the same initial point is used across methods for each trial; starting points are sampled uniformly at random within the stated box constraints. Dual subproblems are solved efficiently via the Frank–Wolfe procedure. For each problem, we report averages over 200 independent runs of: (i) iteration count, (ii) function evaluations, and (iii) CPU time (ms).

Standard Test Suite

Table 1 summarizes the test instances. The second and third columns list the decision dimension n and the number of objectives m, while (x_L, x_U) specifies the lower and upper bounds.

Table 1: Benchmark instances used in the experiments.

10.48047/jocaaa.2025.34.11.17

Problem	n	m	x_L	x_U
DD1	5	2	$(-20, \dots, -20)$	$(20, \dots, 20)$ [7]
Deb	2	2	(0.1,0.1)	(1,1) [8]
Far1	2	2	$(-1, -1)$	(1,1) [9]
FDS	5	3	$(-2, \dots, -2)$	$(2, \dots, 2)$ [10]
FF1	2	2	$(-1, -1)$	(1,1) [11]
Hil1	2	2	(0,0)	(1,1) [12]
Imbalance1	2	2	$(-2, -2)$	(2,2) [13]
Imbalance2	2	2	$(-2, -2)$	(2,2) [14]
LE1	2	2	$(-5, -5)$	(10,10) [15]
PNR	2	2	$(-2, -2)$	(2,2) [16]
VU1	2	2	$(-3, -3)$	(3,3) [17]
WIT1	2	2	$(-2, -2)$	(2,2) [18]
WIT2	2	2	$(-2, -2)$	(2,2) [19]
WIT3	2	2	$(-2, -2)$	(2,2) [20]
WIT4	2	2	$(-2, -2)$	(2,2) [21]

10.48047/jocaaa.2025.34.11.17

WIT5	2	2	$(-2,-2)$	(2,2) [22]
WIT6	2	2	$(-2,-2)$	(2,2) [23]

Table 2 reports the averaged performance metrics. Two main tendencies emerge. First, BBQNMO and SMBBMO typically reduce the iteration count relative to BBDMO, indicating better exploitation of local curvature. Second, SMBBMO often matches or surpasses BBQNMO when $n = 2$, suggesting that the two-dimensional subspace step is particularly effective in low dimension. On the other hand, SMBBMO can be slower in wall-clock time than the two baselines; this is consistent with solving two nested subproblems per iteration and with the relatively benign conditioning of these benchmarks.

Table 2: Averaged iterations (iter), function evaluations (feval), and CPU time (ms) over 200 runs.

	BBDMO			BBQNMO			SMBBMO		
(lr)2-4(lr)5-7 (lr)8-10 Problem	ter	eval	ime	ter	eval	ime	ter	eval	ime
DD1	.77	.91	.36	.82	6.09	.87	.38	.08	.60
Deb	.53	.59	.96	.17	.51	.40	.28	.81	.04
Far1	2.07	2.56	.18	.94	6.11	.74	5.24	5.96	.89
FDS	.12	.35	.60	.54	.77	.90	.83	.23	.87
FF1									

10.48047/jocaaa.2025.34.11.17

	.08	.30	.63	.37	.12	.90	.50	.83	.13
Hil1	.19	.96	.46	.85	.26	.13	.34	0.91	.41
Imbalance1	.55	.48	.40	.46	.33	.62	.00	.86	.62
Imbalance2	.00	.00	.27	.00	.00	.29	.00	.00	.21
LE1	.61	.77	.58	.78	.93	.90	.57	.85	.11
PNR	.30	.58	.88	.38	.40	.73	.17	.28	.89
VU1	3.68	3.73	.86	.73	2.41	.70	1.49	6.47	.32
WIT1	.95	.04	.42	.77	.23	.59	.54	.91	.70
WIT2	.27	.37	.48	.09	.23	.68	.81	.99	.76
WIT3	.17	.26	.59	.87	.97	.80	.52	.77	.02
WIT4	.33	.38	.58	.08	.15	.84	.59	.85	.00
WIT5	.43	.45	.50	.36	.40	.72	.94	.04	.83
WIT6	.00	.00	.22	.00	.00	.24	.00	.00	.23

Discussion.

10.48047/jocaaa.2025.34.11.17

On most problems, both BBQNMO and the proposed SMBBMO reduce the iteration count versus BBDMO, consistent with the use of richer curvature information. The strongest relative gains for SMBBMO are visible on low-dimensional tasks ($n = 2$), where the two-dimensional subspace exactly matches the ambient space. Conversely, the elapsed time for SMBBMO can be higher due to solving the BB subproblem and the reduced-space subproblem at each iteration, an overhead that is amplified when the underlying instances are well conditioned.

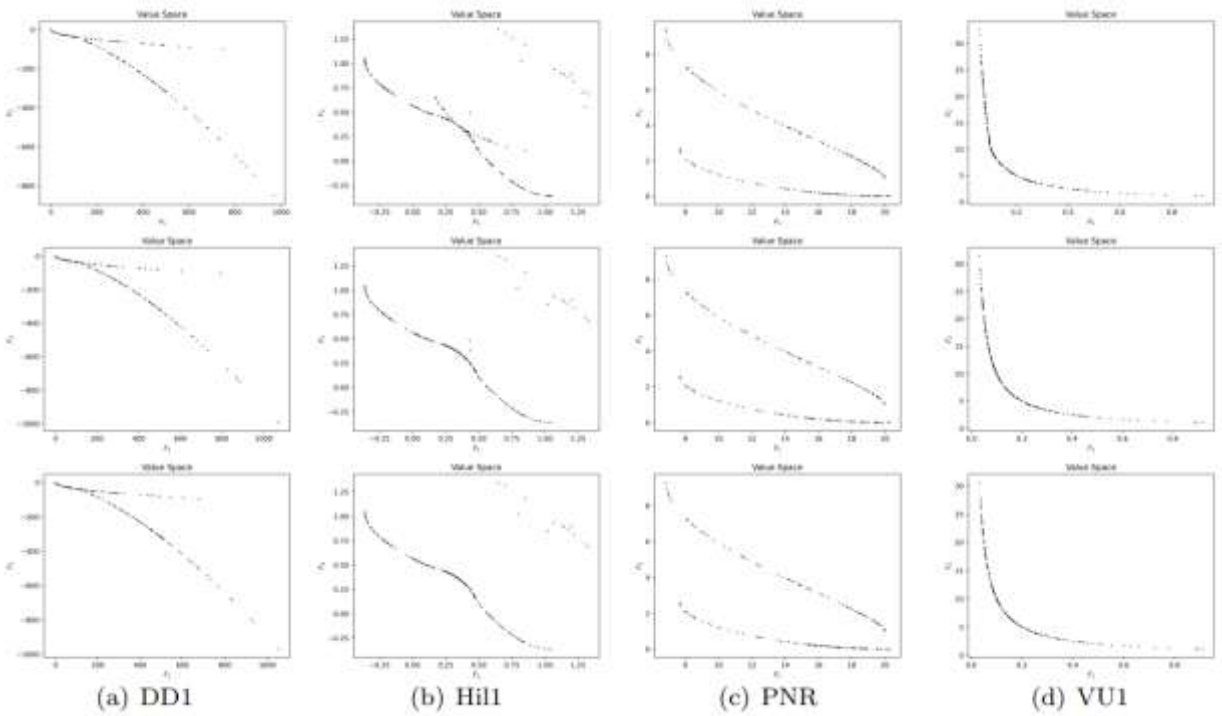

Fig. 1: Numerical results in value space obtained by BBDMO (top), BBQNMO (middle) and SMBBMO for problems DD1, Hil1, PNR, and VU1.

Quadratic, Ill–Conditioned Benchmarks

To stress–test the methods under poor conditioning, we consider pairs of quadratic objectives of the form

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

with $A_i \in \mathbb{S}^n_{++}$. Each matrix is synthesized via a spectral decomposition $A_i = H_i D_i H_i^{\top}$, where

10.48047/jocaaa.2025.34.11.17

H_i is a random orthogonal matrix and $D_i = \mathrm{Diag}(d_1^i, \dots, d_n^i)$, chosen so that the condition number obeys

$$\kappa_i = \frac{\max_j d_j^i}{\min_j d_j^i}.$$

Table 3 details the test instances: the decision dimension n, the pair of condition numbers (κ_1, κ_2), and the box $[x_L, x_U]$ used to sample initial points.

Table 3: Quadratic test instances and conditioning.

Problem	n	(κ_1, κ_2)	(x_L, x_U)
QPa	10	(10,10)	$10[-1, \dots, -1]$, $10[1, \dots, 1]$
QPb	10	$(10^2, 10^2)$	$10[-1, \dots, -1]$, $10[1, \dots, 1]$
QPc	100	$(10^2, 10^2)$	$100[-1, \dots, -1]$, $100[1, \dots, 1]$
QPd	100	$(10^3, 10^3)$	$100[-1, \dots, -1]$, $100[1, \dots, 1]$
QPe	500	$(10^3, 10^3)$	$500[-1, \dots, -1]$, $500[1, \dots, 1]$
QPf	500	$(10^4, 10^4)$	$500[-1, \dots, -1]$, $500[1, \dots, 1]$
QPg	1000	$(10^4, 10^4)$	$1000[-1, \dots, -1$, $1000[1, \dots, 1]$
QPh	1000	$(10^5, 10^5)$	$1000[-1, \dots, -1$, $1000[1, \dots, 1]$

Table 4 reports averages over 200 trials for iteration count, function evaluations, and

10.48047/jocaaa.2025.34.11.17

CPU time (ms). The trends mirror those seen on the standard suite: as conditioning worsens and the dimension grows, curvature–aware methods markedly reduce the number of iterations relative to BBDMO. SMBBMO remains competitive with BBQNMO in iteration counts, while its runtime reflects the extra cost of solving two reduced problems per step.

Table 4: Averaged performance on quadratic ill–conditioned problems (200 runs).

(lr)2-4(lr)5-7(lr)8-10 Problem	BBDMO			BBQNMO			SMBBMO		
	iter	feval	time	iter	feval	time	iter	feval	time
QPa	2.06	13.44	1.38	.55	13.77	1.89	.96	0.65	2.67
QPb	2.24	67.46	5.04	0.16	38.92	4.32	0.67	1.60	5.92
QPc	3.39	82.49	8.47	4.59	65.80	10.52	6.20	2.68	10.39
QPd	80.45	356.16	31.72	2.81	88.31	13.32	8.78	1.57	17.80
QPe	84.43	343.49	111.07	4.94	110.92	308.70	1.60	7.49	47.68
QPf	36.72	1168.17	432.60	16.48	279.83	1286.07	21.55	03.98	80.80
QPg	20.00	909.17	1164.93	57.15	511.41	8483.27	54.84	89.42	468.84
QPh	00.00	2856.25	3513.05	62.81	1106.15	15049.72	75.41	85.56	1542.79

10.48047/jocaaa.2025.34.11.17

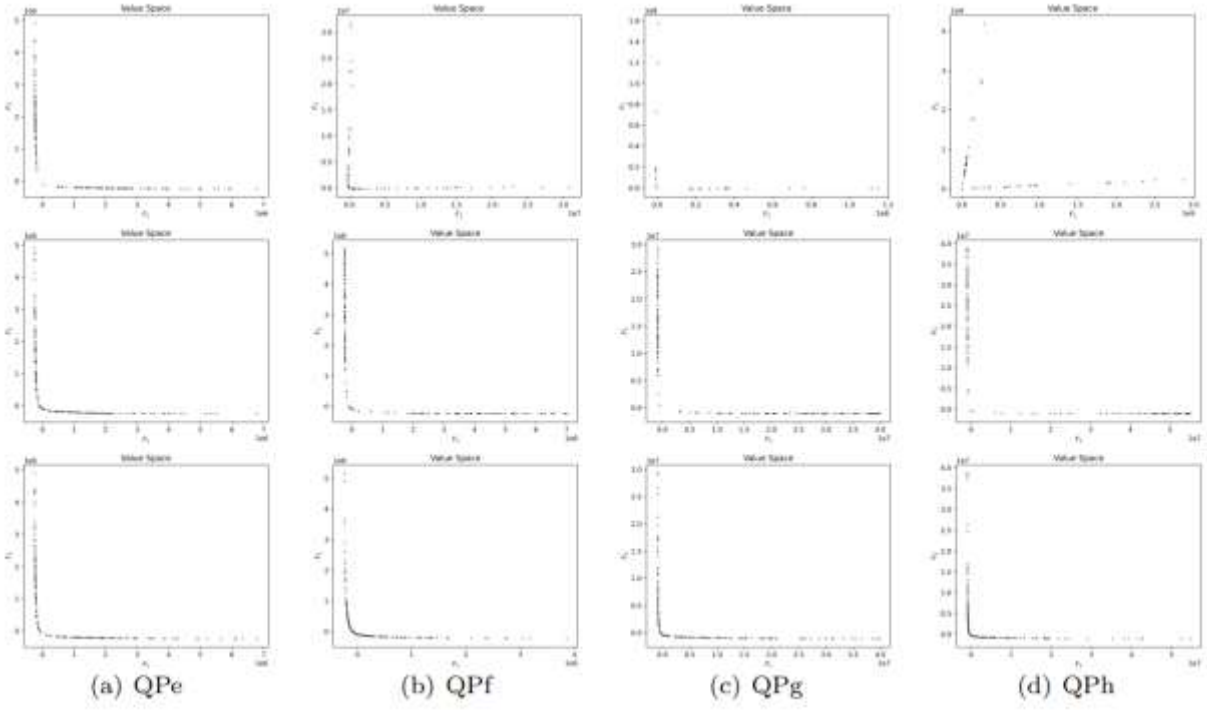

Fig. 2: Numerical results in value space obtained by BBDMO (top), BBQNMO (middle) and SMBBMO for problems QPe, QPf, QPg, and QPh.

Discussion of Table 4.

Table 4 reports averages over 200 trials of iterations (*iter*), objective evaluations (*feval*), and elapsed time in milliseconds (*time*). The first–order BBDMO, aided by the Barzilai–Borwein step scaling, handles moderately ill–conditioned instances (e.g., QPb–QPe) reasonably well, but fails to reach convergence within the 500–iteration limit on the most severely conditioned problems (QPf–QPh). In contrast, on the challenging large–scale, ill–conditioned set (QPe–QPh), SMBBMO frequently achieves lower runtime than BBQNMO, reflecting the efficiency of its two–dimensional subspace step in exploiting local curvature. Overall, the evidence indicates that SMBBMO converges faster than BBDMO while incurring a smaller computational cost than BBQNMO.

Conclusions

We proposed a subspace minimization Barzilai–Borwein framework for multiobjective optimization (SMBBMO). The method improves upon BBDMO in terms of convergence speed

10.48047/jocaaa.2025.34.11.17

and typically requires fewer computational resources than BBQNMO. A modified Cholesky safeguard on a scaled 2×2 model matrix ensures global convergence even in nonconvex settings. The numerical study, spanning both standard and ill–conditioned benchmarks at scale, shows that SMBBMO is a competitive and robust solver for large, poorly conditioned MOPs.

Outlook.

Two methodological extensions are particularly appealing:

• **Richer subspaces.** Incorporating additional historical information (e.g., multi–step memories as in [25, 26]) could yield more informative reduced spaces while keeping the subproblem dimension small.

• **Alternative local models.** Cubically regularized subspace schemes (in the spirit of SMCG with cubic terms [27]) can be adapted to the multiobjective case, potentially improving robustness on highly nonconvex landscapes.

Refrencess

[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.

10.48047/jocaaa.2025.34.11.17

[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. Gra˜na 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.

10.48047/jocaaa.2025.34.11.17

[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. Gra˜na Drummond, and A. M. Masuda. A conjugate directions-type procedure for quadratic multiobjective optimization. *Optimization*, 71(2):419–437, 2022.

[19] M. L. N. Gon¸calves, 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. Gon¸calves 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. Gra˜na 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.

10.48047/jocaaa.2025.34.11.17

[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.