AIMS Mathematics, 11(9): 30904–30943.
DOI:10.3934/math.20261224
Received: 26 July 2026
Revised: 08 September 2026
Accepted: 17 September 2026
Published: 21 September 2026

https://www.aimspress.com/journal/Math

*Research article*

# A restarted three-term vector conjugate gradient method with a diagonal variable metric for nonconvex unconstrained vector optimization

**Yuchang Zhang**[1], **Jing Gao**[1,*] **and Chongyang He**[2]

[1] School of Mathematics and Statistics, Beihua University, Jilin 132013, China
[2] School of Engineering, RMIT University, Melbourne, VIC 3000, Australia

* **Correspondence:** Email: gaojingjy@163.com.

**Abstract:** This paper proposes a restarted three-term vector conjugate gradient method with a diagonal variable metric for nonconvex unconstrained vector optimization. The method incorporates a regularized diagonal Barzilai–Borwein-type variable metric into the vector steepest descent subproblem. A restart mechanism and a safeguarding correction strategy are introduced to ensure that the generated search directions satisfy a uniform sufficient descent condition. Under standard assumptions and a vector Wolfe line search, the global subsequential convergence of the method is established, and the sequence generated by the algorithm is shown to have at least one $K$-critical accumulation point. Numerical experiments demonstrate that the proposed method achieves good computational efficiency and numerical stability. The results also indicate that the diagonal variable metric can effectively improve the overall performance of the algorithm.

**Keywords:** nonconvex vector optimization; diagonal variable metric; Barzilai–Borwein-type update; restarted three-term structure; sufficient descent; global convergence
**Mathematics Subject Classification:** 65K05, 90C26, 90C29

## 1. Introduction

Vector optimization problems arise when several interrelated or conflicting objective functions must be handled simultaneously. They have found broad applications in engineering design [1], healthcare planning [2], aerospace mission planning [3], space exploration [4], facility location and resource allocation [5], management science [6], bilevel programming [7, 8], and statistics [9]. Recent advances in adaptive dynamic programming [10], event-triggered optimal control [11], integral reinforcement learning [12], and game-based optimal control [13] have demonstrated effective optimization strategies for complex nonlinear systems, which may also provide useful insights for extending vector optimization methods to dynamic and multiobjective control settings. Unlike scalar

optimization problems, vector optimization problems generally do not admit a single point that minimizes all objective components simultaneously. Their optimality must therefore be characterized through partial orders induced by ordering cones and the corresponding notions of efficiency. When the ordering cone is the nonnegative orthant, cone efficiency reduces to the usual notion of Pareto efficiency. For continuously differentiable vector optimization problems, efficient solutions satisfy appropriate first-order necessary conditions. Numerical methods are therefore commonly designed to locate or approximate critical points satisfying these conditions.

We consider the following nonconvex unconstrained vector optimization problem:

$$\min_K F(x), \qquad x \in \mathbb{R}^n, \tag{1.1}$$

where $F\colon \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable vector-valued function with component form $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T$ and $K \subset \mathbb{R}^m$ is a closed, convex, and pointed cone with a nonempty interior that induces a partial order in the objective space. When $K = \mathbb{R}^m_+$, problem (1.1) becomes a standard unconstrained multiobjective optimization problem. This paper focuses on descent methods based on cone-order descent criteria for computing $K$-critical points of nonconvex unconstrained vector optimization problems. Multi-start runs can then be used to investigate the distribution of approximate $K$-critical points in the objective space.

Existing approaches to vector optimization can be broadly divided into scalarization methods [14–16] and descent methods based on the underlying partial order. Scalarization methods transform a vector optimization problem into one or more scalar optimization problems by means of weights, parameters, or auxiliary functions. They can therefore take advantage of well-developed algorithms for scalar optimization. Their numerical results, however, often depend strongly on the chosen scalarization parameters. For nonconvex problems, solving a single scalarized problem is generally insufficient to provide an adequate representation of the efficient set. By contrast, descent methods directly use first-order information from the individual objective functions to construct a common descent direction. They also provide a clear first-order characterization of optimality and a natural framework for convergence analysis, and have consequently attracted considerable attention.

The vector steepest descent method [17] is one of the most fundamental descent methods for vector optimization. At each iteration, it computes a common descent direction by solving a strongly convex subproblem. Its numerical performance may nevertheless be affected by differences in variable scaling and local curvature. Building on this approach, Newton-type [18], projected-gradient [19], proximal [20, 21], and conjugate gradient methods have been extended to the vector optimization setting. Vector conjugate gradient methods incorporate information from previous search directions into the direction update. They have a simple structure and require little storage, which makes them potentially attractive for high-dimensional problems. To preserve descent for nonconvex problems, existing methods commonly employ parameter truncation, restart rules, or direction-correction mechanisms. These techniques are usually combined with vector Wolfe line searches to establish global convergence results.

Lucambio Pérez and Prudente developed a nonlinear conjugate gradient framework [22] for vector optimization and established corresponding global convergence results under a vector Wolfe line search [23]. The practical implementation of vector Wolfe line searches was subsequently investigated in greater detail. On this foundation, vector extensions of the Hager–Zhang [24], Liu–Storey [25], spectral conjugate-gradient [26], and Polak–Ribière–Polyak (PRP) methods [27] were developed.

These methods mainly aim to improve the direction update, provide stronger descent guarantees, and enhance numerical stability.

Spectral-gradient and variable-metric techniques [28] can extract local scaling information from first-order differences between consecutive iterates at a relatively low computational cost. The Barzilai–Borwein method constructs a spectral scaling factor from secant information and has demonstrated strong computational performance in scalar optimization. When extended to vector optimization, such scaling can improve the vector steepest descent direction through either a scalar parameter or a positive definite metric. Compared with a single spectral parameter, a diagonal variable metric can adjust the weights associated with different coordinate directions separately, thereby improving the ability of an algorithm to accommodate differences in variable scales.

Combining historical search directions with a diagonal variable metric raises two important issues. First, a direction that is descending at $x_{k-1}$ need not remain descending at the new iterate $x_k$. Directly reusing the previous search direction may therefore weaken or even destroy the common descent property. Second, because the diagonal metric changes during the iteration, the corresponding metric steepest descent subproblem also changes. It is therefore necessary to control the stability of the metric update and to understand how this update affects both the steepest descent direction and the criticality measure. These considerations call for a carefully controlled mechanism for incorporating historical directions, a stable metric update, and a convergence analysis that accounts for both features.

Motivated by these observations, we propose a restarted three-term vector conjugate gradient method with a diagonal variable metric, referred to as DVM-R3TCG. At each iteration, the method first solves a vector steepest descent subproblem under a uniformly positive definite diagonal metric. The diagonal scaling is then updated using first-order differences along a fixed normalized dual direction. The restart term and the historical term are constructed according to the actual descent quality of the previous search direction at the current iterate. When the historical information may jeopardize descent, the method switches to a safeguarded correction branch. A final direction-selection mechanism then ensures that the accepted search direction satisfies a uniform sufficient descent condition.

The main contributions of this paper are summarised as follows.

(1) We propose the DVM-R3TCG algorithm, whose main algorithmic novelty lies in three new constructions for nonconvex unconstrained vector optimization. First, a regularized diagonal variable-metric update is incorporated into the metric steepest $K$-descent subproblem. Second, an adaptive restart three-term direction is constructed to retain or discard the historical direction according to its actual descent quality at the current iterate. Finally, safeguarding correction and direction-selection mechanisms are designed to guarantee uniform sufficient descent. These mechanisms are unified within a single search-direction generation framework, constituting the main algorithmic novelty of DVM-R3TCG over existing vector conjugate gradient methods.
(2) Unlike existing vector conjugate gradient methods that often rely on restrictions, truncations, or additional control conditions to ensure descent, DVM-R3TCG directly guarantees uniform sufficient descent through its safeguarding mechanism. This yields global subsequential convergence and the existence of at least one $K$-critical accumulation point. Under the additional condition $\alpha_k \geq \underline{\alpha} > 0$, we further prove that $\|p_k\| \to 0$ and derive conditional iteration-complexity bounds, which remain uncommon for this class, namely, $T_\tau^p = O(\tau^{-2})$ and $T_\varepsilon^\theta = O(\varepsilon^{-1})$.
(3) Systematic numerical experiments validate the computational efficiency and robustness of DVM-

R3TCG. Ablation experiments separately reveal the practical contribution of the diagonal variable-metric module to the algorithmic performance, while sensitivity analysis of the key parameters verifies the parameter stability and further demonstrates the practical applicability of the method.

The remainder of this paper is organized as follows. Section 2 introduces the basic concepts of vector optimization together with the preliminary results needed for the subsequent analysis. The proposed method is presented in Section 3, where the sufficient descent property of the search directions is also established. The global subsequential convergence analysis is developed in Section 4. Numerical results are reported in Section 5, and the paper is concluded in Section 6.

## 2. Preliminaries

This section presents the basic concepts and properties required for the construction of the algorithm and the convergence analysis [22,24]. We use $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ to denote the Euclidean inner product and its induced norm, respectively. The corresponding spectral norm is used for matrices, while $\|\cdot\|_F$ denotes the Frobenius norm. For a continuously differentiable mapping $F\colon \mathbb{R}^n \to \mathbb{R}^m$, we denote the Jacobian matrix of $F$ at $x$ by $JF(x)$.

### *2.1. Cone order, criticality, and the descent function*

Let $K \subset \mathbb{R}^m$ be a closed, convex, and pointed cone with a nonempty interior. The partial order and strict partial order induced by $K$ are defined by

$$a \preceq_K b \quad \Longleftrightarrow \quad b - a \in K, \qquad a \prec_K b \quad \Longleftrightarrow \quad b - a \in \operatorname{int}(K).$$

When $K = \mathbb{R}^m_+$, these relations reduce to the usual componentwise order.

For problem (1.1), a point $x^*$ is called a $K$-efficient solution if there is no $x \in \mathbb{R}^n$ such that

$$F(x) \preceq_K F(x^*), \qquad F(x) \neq F(x^*).$$

When $K = \mathbb{R}^m_+$, a $K$-efficient solution is precisely a Pareto efficient solution in the usual sense.

A point $x$ is called $K$-critical if

$$\operatorname{Im}(JF(x)) \cap (-\operatorname{int}(K)) = \varnothing.$$

If $x$ is not $K$-critical, then there exists a direction $d \in \mathbb{R}^n$ such that $JF(x)d \in -\operatorname{int}(K)$. In this case, $d$ is called a $K$-descent direction for $F$ at $x$.

From the above relations, it follows that for continuously differentiable vector optimization problems, every $K$-efficient point is $K$-critical; in particular, when $K = \mathbb{R}^m_+$, every Pareto efficient point is $K$-critical. For nonconvex problems, however, $K$-criticality is only a first-order necessary condition and does not guarantee global $K$-efficiency.

The dual cone of $K$ is defined by

$$K^* := \{\lambda \in \mathbb{R}^m \mid \langle \lambda, y\rangle \geq 0 \text{ for all } y \in K\}.$$

To ensure that the descent criterion and the line-search conditions are computable, we further assume that $K^*$ is finitely generated. Let

$$Z = \left\{\lambda^1, \lambda^2, \ldots, \lambda^q\right\} \subset K^* \setminus \{0\}, \qquad \|\lambda^i\| = 1, \quad i = 1, \ldots, q$$

be a set of normalized generators satisfying

$$K^* = \operatorname{cone}(Z) = \left\{ \sum_{i=1}^{q} \alpha_i \lambda^i \;\middle|\; \alpha_i \geq 0, \quad i = 1, \ldots, q \right\}.$$

This assumption is equivalent to requiring $K^*$ to be a polyhedral cone. The normalization of the generators eliminates the effect of arbitrary scaling on the descent function and the algorithmic parameters. When $K = \mathbb{R}^m_+$, we have $K^* = \mathbb{R}^m_+$ and may take

$$Z = \{e_1, e_2, \ldots, e_m\},$$

where $e_i$ is the $i$th unit vector in $\mathbb{R}^m$.

Define

$$\phi(y) := \max_{\lambda \in Z} \langle \lambda, y \rangle.$$

The properties of the dual cone and the bipolar cone yield

$$y \in -K \quad \Longleftrightarrow \quad \phi(y) \leq 0, \qquad y \in -\operatorname{int}(K) \quad \Longleftrightarrow \quad \phi(y) < 0.$$

We further define the cone-order descent function by

$$H(x, d) := \phi\big(JF(x)d\big) = \max_{\lambda \in Z} \langle \lambda, JF(x)d \rangle. \tag{2.1}$$

Therefore, $d$ is a $K$-descent direction at $x$ if and only if $H(x, d) < 0$. Moreover, $x$ is $K$-critical if and only if

$$H(x, d) \geq 0, \quad \text{for all } d \in \mathbb{R}^n.$$

**Lemma 2.1.** *The function $H$ has the following properties.*

(1) For all $x, d_1, d_2 \in \mathbb{R}^n$ and all $a, b \geq 0$,

$$H(x, ad_1 + bd_2) \leq aH(x, d_1) + bH(x, d_2).$$

In particular, for all $a \geq 0$,

$$H(x, ad) = aH(x, d).$$

(2) For all $x, y, d \in \mathbb{R}^n$,

$$|H(x, d) - H(y, d)| \leq \|JF(x) - JF(y)\| \, \|d\|.$$

If $JF$ is Lipschitz continuous on $\Omega$ with constant $L_J > 0$, then

$$|H(x, d) - H(y, d)| \leq L_J \|x - y\| \, \|d\|, \qquad x, y \in \Omega. \tag{2.2}$$

*Proof.* The subadditivity and positive homogeneity of $\phi$ give

$$\begin{aligned} H(x, ad_1 + bd_2) &= \phi(aJF(x)d_1 + bJF(x)d_2) \\ &\le a\phi(JF(x)d_1) + b\phi(JF(x)d_2). \end{aligned}$$

Hence, the first assertion follows.

Using the elementary estimate for a finite maximum and $\|\lambda\| = 1$, we obtain

$$\begin{aligned} |H(x, d) - H(y, d)| &\le \max_{\lambda \in Z} \left|\langle \lambda, (JF(x) - JF(y))d\rangle\right| \\ &\le \|JF(x) - JF(y)\| \, \|d\|. \end{aligned}$$

Combining this estimate with the Lipschitz continuity of $JF$ yields (2.2). □

### 2.2. Regularized diagonal variable metric

Let

$$U_k = \mathrm{Diag}(u_k), \qquad u_k = (u_{k,1}, u_{k,2}, \ldots, u_{k,n})^T.$$

Given $s_k, y_k \in \mathbb{R}^n$, a regularization parameter $\omega > 0$, and constants $0 < m_L \le 1 \le m_U < +\infty$, we define $U_{k+1}$ as the solution of the following regularized diagonal secant-fitting problem:

$$\begin{aligned} \min_{U = \mathrm{Diag}(u)} \quad & \|Us_k - y_k\|^2 + \omega \|U - U_k\|_F^2 \\ \text{subject to} \quad & m_L I \le U \le m_U I. \end{aligned} \tag{2.3}$$

Problem (2.3) can be decomposed into $n$ independent one-dimensional strongly convex problems. Let $\Pi_{[m_L, m_U]}$ denote the projection onto the interval $[m_L, m_U]$. Then,

$$\widetilde{u}_{k+1,i} = \frac{s_{k,i} y_{k,i} + \omega u_{k,i}}{s_{k,i}^2 + \omega}, \qquad u_{k+1,i} = \Pi_{[m_L, m_U]}(\widetilde{u}_{k+1,i}), \qquad i = 1, \ldots, n. \tag{2.4}$$

The regularization term is used to suppress abrupt changes in the metric matrix, while the interval projection ensures that the metric matrix remains uniformly positive definite.

To construct $y_k$, fix a normalized dual direction satisfying

$$\lambda^0 \in \mathrm{int}(K^*), \qquad \|\lambda^0\| = 1,$$

and define

$$g(x) := JF(x)^T \lambda^0, \qquad s_k := x_{k+1} - x_k, \qquad y_k := g(x_{k+1}) - g(x_k). \tag{2.5}$$

When $K = \mathbb{R}^m_+$, we may take

$$\lambda^0 = \frac{1}{\sqrt{m}}(1, 1, \ldots, 1)^T.$$

The vector $\lambda^0$ is used only to extract diagonal scaling information. The descent property of a search direction is still determined by $H(x, d)$.

**Lemma 2.2.** *Suppose that $U_0 = I$ and that $U_k$ is updated according to* (2.3)–(2.4)*. Then, for every $k \geq 0$, the following inequalities hold:*

$$m_L I \preceq U_k \preceq m_U I, \qquad m_L \|d\|^2 \leq d^T U_k d \leq m_U \|d\|^2. \tag{2.6}$$

If $JF$ is Lipschitz continuous with constant $L_J$ on a set containing $x_k$ and $x_{k+1}$, then

$$\|U_{k+1} - U_k\|_F \leq C_U \|s_k\|^2, \qquad C_U := \frac{L_J + m_U}{\omega}. \tag{2.7}$$

*Proof.* Since $U_0 = I$ and $m_L \leq 1 \leq m_U$, the conclusion holds for $k = 0$. It follows from (2.4) that

$$u_{k+1,i} \in [m_L, m_U], \qquad i = 1, \ldots, n.$$

Therefore, the first pair of matrix inequalities follows by induction, while the quadratic-form estimates follow directly from the lower and upper eigenvalue bounds of $U_k$.

We next prove (2.7). Since $u_{k,i} \in [m_L, m_U]$ and the interval projection is nonexpansive, we have

$$|u_{k+1,i} - u_{k,i}| \leq |\widetilde{u}_{k+1,i} - u_{k,i}|.$$

By (2.4),

$$\widetilde{u}_{k+1,i} - u_{k,i} = \frac{s_{k,i}(y_{k,i} - u_{k,i} s_{k,i})}{s_{k,i}^2 + \omega}.$$

Let

$$D_k := \mathrm{Diag}(s_{k,1}^2 + \omega, \ldots, s_{k,n}^2 + \omega).$$

Using the fact that the Frobenius norm of a diagonal matrix equals the Euclidean norm of its diagonal vector, we obtain

$$\begin{aligned} \|U_{k+1} - U_k\|_F &\leq \left\| D_k^{-1} \,\mathrm{Diag}(s_k)(y_k - U_k s_k) \right\| \\ &\leq \frac{\|s_k\|}{\omega} \left( \|y_k\| + \|U_k s_k\| \right). \end{aligned}$$

By (2.5), $\|\lambda^0\| = 1$, and the Lipschitz continuity of $JF$, we have

$$\begin{aligned} \|y_k\| &= \left\| \left( JF(x_{k+1}) - JF(x_k) \right)^T \lambda^0 \right\| \\ &\leq \|JF(x_{k+1}) - JF(x_k)\| \, \|\lambda^0\| \\ &\leq L_J \|s_k\|. \end{aligned}$$

Combining this estimate with $\|U_k s_k\| \leq m_U \|s_k\|$ gives (2.7). □

### *2.3. Metric steepest K-descent direction*

Given a symmetric positive definite matrix $U$, define the metric steepest $K$-descent direction by

$$v(x; U) := \arg\min_{d \in \mathbb{R}^n} \left\{ H(x, d) + \frac{1}{2} d^T U d \right\}, \tag{2.8}$$

and define the corresponding criticality function by

$$\theta(x; U) := H(x, v(x; U)) + \frac{1}{2} v(x; U)^T U v(x; U). \tag{2.9}$$

Since $H(x,\cdot)$ is a finite convex function and $U \succ 0$, the objective function in (2.8) is strongly convex and coercive. Hence, its minimizer exists and is unique.

When $Z$ is finite, problem (2.8) is equivalent to the following strongly convex quadratic program:

$$\begin{aligned} \min_{d\in\mathbb{R}^n,\, t\in\mathbb{R}} \quad & t + \frac{1}{2}d^T U d \\ \text{subject to} \quad & \langle \lambda, JF(x)d\rangle \le t, \qquad \lambda \in Z. \end{aligned}$$

In particular, when $K = \mathbb{R}^m_+$, the constraints can be written as

$$[JF(x)d]_i \le t, \qquad i = 1,\dots,m.$$

This formulation is used to solve the metric steepest descent subproblem numerically.

**Lemma 2.3.** *Let $U \succ 0$ and let $v = v(x;U)$. Then the following conclusions hold.*

(1) The point $x$ is $K$-critical if and only if $v = 0$, which is also equivalent to $\theta(x;U) = 0$.
(2) For every $x$, the following exact identities hold:

$$H(x,v) = -v^T U v, \qquad \theta(x;U) = -\frac{1}{2}v^T U v. \tag{2.10}$$

(3) If $x$ is not $K$-critical, then $v \neq 0$ and

$$H(x,v) < 0, \qquad \theta(x;U) < 0.$$

(4) The mapping $(x,U) \longmapsto v(x;U)$ and the function $(x,U) \longmapsto \theta(x;U)$ are continuous on the set where $U \succ 0$.

*Proof.* Define

$$\Phi(d;x,U) := H(x,d) + \frac{1}{2}d^T U d.$$

Suppose first that $x$ is $K$-critical. Then $H(x,d) \ge 0$ for every $d \in \mathbb{R}^n$, and hence,

$$\Phi(d;x,U) \ge \frac{1}{2}d^T U d \ge 0 = \Phi(0;x,U).$$

Since the objective function is strongly convex, its minimizer is unique. Therefore, $v = 0$.

Conversely, suppose that $v = 0$. For every $d \in \mathbb{R}^n$ and every $t > 0$, we have

$$0 = \Phi(0;x,U) \le \Phi(td;x,U) = tH(x,d) + \frac{t^2}{2}d^T U d.$$

Dividing both sides by $t > 0$ and letting $t \downarrow 0$ gives

$$H(x,d) \ge 0, \quad \text{for all } d \in \mathbb{R}^n.$$

Therefore, $x$ is $K$-critical.

We next prove (2.10). If $v = 0$, the conclusion is immediate. Suppose that $v \neq 0$ and consider the univariate function

$$q(t) := \Phi(tv;x,U) = tH(x,v) + \frac{t^2}{2}v^T U v, \qquad t \ge 0.$$

Since $v$ is the minimizer of $\Phi(\cdot; x, U)$, the point $t = 1$ is an interior minimizer of $q$ on $[0, +\infty)$. Hence,

$$q'(1) = H(x, v) + v^T U v = 0.$$

It follows that

$$H(x, v) = -v^T U v.$$

Substituting this identity into (2.9) gives

$$\theta(x; U) = -\frac{1}{2} v^T U v.$$

Consequently, $\theta(x; U) = 0$ if and only if $v = 0$. When $v \neq 0$, the positive definiteness of $U$ yields

$$H(x, v) < 0, \qquad \theta(x; U) < 0.$$

It remains to prove continuity. Suppose that $(x_j, U_j) \longrightarrow (x, U)$, $v_j := v(x_j; U_j)$. Since $U \succ 0$, there exists a constant $c > 0$ such that, for all sufficiently large $j$, $U_j \geq cI$. Moreover,

$$\Phi(v_j; x_j, U_j) \leq \Phi(0; x_j, U_j) = 0.$$

Since $\{JF(x_j)\}$ is bounded, we obtain

$$\frac{c}{2}\|v_j\|^2 \leq -H(x_j, v_j) \leq \|JF(x_j)\| \, \|v_j\|.$$

Therefore, the sequence $\{v_j\}$ is bounded.

Choose an arbitrary convergent subsequence satisfying $v_{j_\ell} \longrightarrow \bar{v}$. By the minimizing property, for every $d \in \mathbb{R}^n$,

$$\Phi(v_{j_\ell}; x_{j_\ell}, U_{j_\ell}) \leq \Phi(d; x_{j_\ell}, U_{j_\ell}).$$

Letting $\ell \to \infty$ and using the continuity of $H$, we obtain

$$\Phi(\bar{v}; x, U) \leq \Phi(d; x, U), \quad \text{for all } d \in \mathbb{R}^n.$$

Therefore,

$$\bar{v} = v(x; U).$$

Since the minimizer is unique, the entire sequence $\{v_j\}$ converges to $v(x; U)$. Hence, $v(x; U)$ is continuous. The continuity of $\theta(x; U)$ follows immediately from (2.9). □

It follows from (2.6) and (2.10) that whenever $m_L I \preceq U \preceq m_U I$, we have

$$m_L \|v(x; U)\|^2 \leq -H\big(x, v(x; U)\big) \leq m_U \|v(x; U)\|^2. \tag{2.11}$$

Therefore, $\|v(x; U)\|$ can be used as a criticality measure under uniformly bounded metric conditions. We next present a stability result required for the convergence analysis. Define

$$\mathcal{U} := \left\{ U \in \mathbb{R}^{n \times n} \;\middle|\; U = U^T, \quad m_L I \preceq U \preceq m_U I \right\}.$$

**Lemma 2.4.** *Let $C \subset \Omega$ be compact, and suppose that $JF$ is Lipschitz continuous on $\Omega$ with constant $L_J$. Then there exists a constant $L_\theta > 0$ such that, for all $x, y \in C$ and $U, V \in \mathcal{U}$,*

$$|\theta(x; U) - \theta(y; V)| \leq L_\theta \left(\|x - y\| + \|U - V\|_F\right). \tag{2.12}$$

*Proof.* By Lemma 2.3, the mapping $v(x; U)$ is continuous on the compact set $C \times \mathcal{U}$. Hence, there exists a constant $M_v > 0$ such that

$$\|v(x; U)\| \leq M_v, \qquad (x, U) \in C \times \mathcal{U}.$$

Let

$$p = v(x; U), \qquad q = v(y; V).$$

By the minimizing properties of $p$ and $q$, we have

$$\begin{aligned}\theta(x; U) - \theta(y; V) &\leq H(x, q) - H(y, q) + \frac{1}{2} q^T (U - V) q \\ &\leq L_J M_v \|x - y\| + \frac{1}{2} M_v^2 \|U - V\|_F.\end{aligned}$$

Interchanging $(x, U)$ and $(y, V)$ gives the estimate in the opposite direction. Therefore,

$$|\theta(x; U) - \theta(y; V)| \leq L_J M_v \|x - y\| + \frac{1}{2} M_v^2 \|U - V\|_F.$$

Choosing an appropriate constant $L_\theta > 0$ yields (2.12). □

### *2.4. Vector Wolfe line search*

Choose parameters satisfying $0 < \rho < \sigma < 1$. Select $e \in \text{int}(K)$ and scale it appropriately so that

$$0 < c_\lambda := \langle \lambda, e \rangle \leq 1, \qquad \lambda \in Z.$$

Since $Z$ is finite and $e \in \text{int}(K)$, such a scaling can always be achieved.

Let $d_k$ be a $K$-descent direction at $x_k$, which means that $H(x_k, d_k) < 0$. If a step length $\alpha_k > 0$ satisfies

$$F(x_k + \alpha_k d_k) \preceq_K F(x_k) + \rho \alpha_k H(x_k, d_k) e, \tag{2.13a}$$

$$H(x_k + \alpha_k d_k, d_k) \geq \sigma H(x_k, d_k), \tag{2.13b}$$

then $\alpha_k$ is said to satisfy the standard vector Wolfe conditions. Condition (2.13a) is the sufficient decrease condition, while condition (2.13b) is the curvature condition.

If (2.13a) holds and

$$|H(x_k + \alpha_k d_k, d_k)| \leq -\sigma H(x_k, d_k), \tag{2.14}$$

then $\alpha_k$ is said to satisfy the strong vector Wolfe conditions. Since $H(x_k, d_k) < 0$, condition (2.14) is equivalent to

$$\sigma H(x_k, d_k) \leq H(x_k + \alpha_k d_k, d_k) \leq -\sigma H(x_k, d_k).$$

Therefore, the strong vector Wolfe conditions imply the standard vector Wolfe curvature condition.

**Proposition 2.1.** *Suppose that the initial level set*

$$\mathcal{L} := \{z \in \mathbb{R}^n \mid F(z) \preceq_K F(x_0)\}$$

is bounded below with respect to $K$. More precisely, suppose that there exists $F^\ell \in \mathbb{R}^m$ such that

$$F^\ell \preceq_K F(z), \qquad z \in \mathcal{L}.$$

If $F$ is continuously differentiable, $x \in \mathcal{L}$, and $H(x, d) < 0$, then there exists $\alpha > 0$ satisfying the standard vector Wolfe conditions.

*Proof.* Define

$$h(t) := H(x + td, d), \qquad h_0 := H(x, d) < 0.$$

The function $h$ is continuous, and

$$h(0) = h_0 < \sigma h_0.$$

We first prove that there exists $\bar{\alpha} > 0$ such that

$$h(\bar{\alpha}) \geq \sigma h_0.$$

Suppose, to the contrary, that

$$h(t) < \sigma h_0$$

holds for every $t \geq 0$. Then, for every $\lambda \in Z$ and every $\alpha > 0$,

$$\begin{aligned} \langle \lambda, F(x + \alpha d) - F(x) \rangle &= \int_0^\alpha \langle \lambda, JF(x + td)d \rangle \, dt \\ &\leq \int_0^\alpha h(t) \, dt \\ &< \alpha \sigma h_0 < 0. \end{aligned}$$

Since the above inequality holds for all $\lambda \in Z$ and $Z$ generates $K^*$, it follows that

$$F(x + \alpha d) \prec_K F(x) \preceq_K F(x_0).$$

Consequently,

$$x + \alpha d \in \mathcal{L}, \qquad \alpha > 0.$$

For each fixed $\lambda \in Z$, we also have

$$\langle \lambda, F(x + \alpha d) \rangle < \langle \lambda, F(x) \rangle + \alpha \sigma h_0 \longrightarrow -\infty.$$

However, the inequality $F^\ell \preceq_K F(x + \alpha d)$ implies that

$$\langle \lambda, F(x + \alpha d) \rangle \geq \left\langle \lambda, F^\ell \right\rangle,$$

which gives a contradiction. Therefore, the required value $\bar{\alpha}$ must exist.

Set

$$\bar{\alpha} := \inf \{t > 0 \mid h(t) \geq \sigma h_0\}.$$

By the continuity of $h$, we have

$$h(\bar{\alpha}) = \sigma h_0, \qquad h(t) < \sigma h_0, \qquad 0 \le t < \bar{\alpha}.$$

For every $\lambda \in Z$, it follows that

$$\begin{aligned}
&\langle \lambda, F(x + \bar{\alpha}d) - F(x) - \rho\bar{\alpha}h_0 e\rangle \\
&\quad = \int_0^{\bar{\alpha}} \langle \lambda, JF(x + td)d\rangle \, dt - \rho\bar{\alpha}h_0\langle \lambda, e\rangle \\
&\quad \le \bar{\alpha}\sigma h_0 - \rho\bar{\alpha}h_0 c_\lambda \\
&\quad = \bar{\alpha}h_0 (\sigma - \rho c_\lambda) \le 0.
\end{aligned}$$

Here, we have used

$$0 < c_\lambda \le 1, \qquad 0 < \rho < \sigma < 1, \qquad h_0 < 0.$$

Since $Z$ generates $K^*$, we obtain

$$F(x + \bar{\alpha}d) \preceq_K F(x) + \rho\bar{\alpha}H(x, d)e.$$

At the same time,

$$H(x + \bar{\alpha}d, d) = \sigma H(x, d).$$

Therefore, $\bar{\alpha}$ satisfies the standard vector Wolfe conditions. □

## 3. Algorithm description

This section presents the restarted three-term vector conjugate gradient method with a diagonal variable metric, abbreviated as DVM-R3TCG. We also establish the well-definedness of the algorithmic parameters and the uniform sufficient descent property of the search directions. The term "three-term" means that the candidate direction consists of the baseline term, the restart correction term, and the historical direction term, namely

$$\widetilde{d_k} = p_k - r_k p_k + |\beta_k| d_{k-1}.$$

The three terms provide current descent information, adjust the degree of restart, and exploit historical search information, respectively. They are not required to correspond to three linearly independent directions.

### *3.1. Restarted three-term candidate direction*

Inspired by three-term constructions in scalar nonlinear conjugate gradient methods [29], we develop a corresponding direction structure for vector optimization using the cone-order descent function $H$.

At iteration $k$, given the current iterate $x_k$ and the diagonal variable-metric matrix $U_k$, define

$$p_k := v(x_k; U_k), \qquad \theta_k := \theta(x_k; U_k), \qquad a_k := -H(x_k, p_k). \tag{3.1}$$

By Lemma 2.3, if $x_k$ is not $K$-critical, then $p_k \neq 0$ and

$$a_k = p_k^T U_k p_k = -2\theta_k \geq m_L \|p_k\|^2 > 0.$$

The initial search direction is chosen as

$$d_0 = p_0.$$

In what follows, assume that $k \geq 1$ and that $d_{k-1}$ is a $K$-descent direction at $x_{k-1}$.

To measure the possible non-descent effect of the historical direction $d_{k-1}$ at the current point, define

$$S_k := \max\{H(x_k, d_{k-1}), 0\}. \tag{3.2}$$

When $S_k = 0$, the historical direction has no positive adverse effect on the descent property at the current point. When $S_k > 0$, the influence of the historical direction must be controlled.

Define the scaling parameter and the numerator of the historical coefficient by

$$\delta_k := \left(\frac{a_k}{a_{k-1}}\right)^{1/2}, \qquad N_k := a_k + \frac{\delta_k}{2}\left[H(x_k, p_{k-1}) + H(x_{k-1}, p_{k-1})\right]. \tag{3.3}$$

Furthermore, let

$$W_k := \max\{T_{k,1}, T_{k,2}, T_{k,3}, T_{k,4}\}, \tag{3.4}$$

where

$$\begin{aligned} T_{k,1} &:= \eta\sqrt{a_k a_{k-1}}, \\ T_{k,2} &:= a_{k-1}, \\ T_{k,3} &:= H(x_k, d_{k-1}) - H(x_{k-1}, d_{k-1}), \\ T_{k,4} &:= -H(x_{k-1}, d_{k-1}), \end{aligned} \qquad \eta > 0.$$

Define the historical coefficient and the restart parameter by

$$\beta_k := \frac{N_k}{W_k}, \qquad r_k := \zeta_k \frac{S_k}{W_k}, \qquad 0 \leq \zeta_k \leq \bar{\zeta} < 1. \tag{3.5}$$

Here, $W_k$ prevents degeneration of the denominator and suppresses excessive growth of the parameters, while $r_k$ adjusts the restart intensity according to the descent quality of the historical direction at the current point.

**Lemma 3.1.** *Suppose that neither $x_{k-1}$ nor $x_k$ is $K$-critical and that $d_{k-1}$ is a $K$-descent direction at $x_{k-1}$. Then all the parameters in* (3.3)–(3.5) *are well defined, and*

$$W_k > 0, \qquad 0 \leq r_k \leq \bar{\zeta} < 1.$$

*Proof.* By Lemma 2.3 and the descent property of $d_{k-1}$, we have

$$a_k > 0, \qquad a_{k-1} > 0, \qquad -H(x_{k-1}, d_{k-1}) > 0.$$

Therefore,

$$T_{k,1} > 0, \qquad T_{k,2} > 0, \qquad T_{k,4} > 0,$$

and hence, $W_k > 0$. Since $a_k > 0$ and $a_{k-1} > 0$, the parameters $\delta_k, \beta_k$, and $r_k$ are all well defined.

If $H(x_k, d_{k-1}) \leq 0$, then (3.2) gives $S_k = 0$, and consequently $r_k = 0$.

If $H(x_k, d_{k-1}) > 0$, then $S_k = H(x_k, d_{k-1})$. Since $H(x_{k-1}, d_{k-1}) < 0$, we obtain

$$\begin{aligned} T_{k,3} &= H(x_k, d_{k-1}) - H(x_{k-1}, d_{k-1}) \\ &> H(x_k, d_{k-1}) = S_k. \end{aligned}$$

Thus, $W_k > S_k$, and therefore

$$0 \leq r_k = \zeta_k \frac{S_k}{W_k} < \zeta_k \leq \bar{\zeta} < 1.$$

□

Choose parameters satisfying

$$0 < \widehat{\mu} < 1, \qquad \widehat{\mu} + \bar{\zeta} < 1.$$

If

$$|\beta_k| S_k \leq \widehat{\mu} a_k, \tag{3.6}$$

then the possible non-descent effect of the historical direction remains sufficiently controlled by the current baseline descent quantity, and the three-term direction retaining the historical information is used.

If (3.6) does not hold, define the safeguarded direction by

$$q_k := \begin{cases} d_{k-1}, & H(x_k, d_{k-1}) < 0, \\ p_k, & H(x_k, d_{k-1}) \geq 0. \end{cases} \tag{3.7}$$

It follows directly from the definition that $H(x_k, q_k) < 0$ in both cases. Given parameters $0 < \xi < 1$ and $\psi > 0$, with $\beta_k$ and $r_k$ defined by (3.5) and $q_k$ defined by (3.7), we define the candidate direction as

$$\widetilde{d_k} := \begin{cases} p_k - r_k p_k + |\beta_k| d_{k-1}, & \text{if (3.6) holds,} \\ p_k + \xi \dfrac{a_k}{\max\{-H(x_k, q_k), \psi \|q_k\|\}} q_k, & \text{otherwise.} \end{cases} \tag{3.8}$$

The first case retains the baseline term, the restart correction term, and the historical direction term. The second case uses the known descent direction $q_k$ to apply a safeguarded correction to the baseline direction $p_k$.

From the algorithmic design, compared with the common practice of taking the restart direction directly as $p_k$, we adopt a restart mechanism along the ray generated by the current metric steepest $K$-descent direction $p_k$. When condition (3.6) does not hold, we have $S_k > 0$, and hence $H(x_k, d_{k-1}) > 0$. By (3.7), $q_k = p_k$. In this case, the second branch of (3.8) reduces to

$$\widetilde{d_k} = c_k p_k, \qquad c_k = 1 + \xi \frac{a_k}{\max\{a_k, \psi \|p_k\|\}} > 1.$$

Therefore, the historical direction $d_{k-1}$ is completely discarded, and the $c_k$-multiple of the current metric steepest $K$-descent direction $p_k$ is selected as the restart direction.

**Lemma 3.2.** *If $x_k$ is not $K$-critical, then the candidate direction defined by* (3.8) *satisfies*

$$H(x_k, \widetilde{d}_k) < 0.$$

*Proof.* Since $x_k$ is not $K$-critical, we have

$$a_k = -H(x_k, p_k) > 0.$$

Suppose first that (3.6) holds. By the sublinearity of $H(x_k, \cdot)$,

$$H(x_k, \widetilde{d}_k) \leq (1 - r_k)H(x_k, p_k) + |\beta_k|H(x_k, d_{k-1}).$$

If $H(x_k, d_{k-1}) \leq 0$, then $r_k < 1$ and $H(x_k, p_k) < 0$ immediately give

$$H(x_k, \widetilde{d}_k) < 0.$$

If $H(x_k, d_{k-1}) > 0$, then $S_k = H(x_k, d_{k-1})$. Combining (3.6) with $r_k \leq \bar{\zeta}$ gives

$$\begin{aligned} H(x_k, \widetilde{d}_k) &\leq -(1 - r_k)a_k + |\beta_k|S_k \\ &\leq -(1 - r_k - \widehat{\mu})a_k \\ &\leq -(1 - \bar{\zeta} - \widehat{\mu})a_k < 0. \end{aligned}$$

Now, suppose that (3.6) does not hold. Let

$$M_k := \max\left\{-H(x_k, q_k), \psi\|q_k\|\right\}.$$

Since $H(x_k, q_k) < 0$, we have $M_k > 0$. Using sublinearity again yields

$$\begin{aligned} H(x_k, \widetilde{d}_k) &\leq H(x_k, p_k) + \xi\frac{a_k}{M_k}H(x_k, q_k) \\ &< H(x_k, p_k) < 0. \end{aligned}$$

Therefore, $\widetilde{d}_k$ is a $K$-descent direction at $x_k$ in both cases. □

### *3.2. Uniform sufficient descent direction*

Although the candidate direction has the strict descent property, its degree of descent need not admit a uniform lower bound. A final safeguarded selection mechanism is therefore introduced.

Choose

$$\mu > \frac{1}{2}, \qquad \chi := 1 - \frac{1}{2\mu} \in (0, 1).$$

By Lemma 3.2, we have $\widetilde{d}_k \neq 0$. Define the safeguarded correction parameter by

$$\bar{\beta}_k := \frac{H(x_k, p_k)}{2\mu \max\left\{H(x_k, -\widetilde{d}_k), \psi\|\widetilde{d}_k\|\right\}}. \tag{3.9}$$

The denominator in (3.9) is positive, while $H(x_k, p_k) < 0$. Hence,

$$\bar{\beta}_k < 0.$$

Using the candidate direction $\widetilde{d}_k$ and the safeguarded correction parameter $\bar{\beta}_k$ defined above, we define the final search direction as

$$d_k := \begin{cases} \widetilde{d}_k, & H(x_k, \widetilde{d}_k) \le \chi H(x_k, p_k), \\ p_k + \bar{\beta}_k \widetilde{d}_k, & \text{otherwise.} \end{cases} \tag{3.10}$$

The first case accepts the candidate direction directly. Otherwise, the safeguarded correction is applied to the candidate direction. Since $\bar{\beta}_k < 0$, the correction term $\bar{\beta}_k \widetilde{d}_k$ is a positive multiple of $-\widetilde{d}_k$, whose magnitude is controlled by $\max\{H(x_k, -\widetilde{d}_k), \psi\|\widetilde{d}_k\|\}$ in (3.9). The resulting sufficient descent property is established in Proposition 3.1.

**Proposition 3.1.** *If $x_k$ is not K-critical, then the search direction defined by* (3.10) *satisfies*

$$H(x_k, d_k) \le \chi H(x_k, p_k) < 0. \tag{3.11}$$

*Proof.* When $k = 0$, we have $d_0 = p_0$. Since $H(x_0, p_0) < 0$ and $0 < \chi < 1$, it follows that

$$H(x_0, d_0) = H(x_0, p_0) \le \chi H(x_0, p_0) < 0.$$

Assume henceforth that $k \ge 1$.

If the first case in (3.10) holds, then (3.11) follows directly from the definition.

Finally, consider the remaining case. In this case,

$$H(x_k, \widetilde{d}_k) > \chi H(x_k, p_k), \qquad H(x_k, -\widetilde{d}_k) > 0, \qquad \bar{\beta}_k < 0.$$

Observe that

$$d_k = p_k + (-\bar{\beta}_k)(-\widetilde{d}_k), \qquad -\bar{\beta}_k > 0.$$

Using the sublinearity of $H(x_k, \cdot)$, we obtain

$$H(x_k, d_k) \le H(x_k, p_k) + (-\bar{\beta}_k) H(x_k, -\widetilde{d}_k).$$

By (3.9),

$$\begin{aligned} (-\bar{\beta}_k) H(x_k, -\widetilde{d}_k) &= \frac{-H(x_k, p_k) H(x_k, -\widetilde{d}_k)}{2\mu \max\left\{H(x_k, -\widetilde{d}_k), \psi\|\widetilde{d}_k\|\right\}} \\ &\le -\frac{H(x_k, p_k)}{2\mu}. \end{aligned}$$

Therefore,

$$\begin{aligned} H(x_k, d_k) &\le H(x_k, p_k) - \frac{H(x_k, p_k)}{2\mu} \\ &= \left(1 - \frac{1}{2\mu}\right) H(x_k, p_k) \\ &= \chi H(x_k, p_k) < 0. \end{aligned}$$

□

By Proposition 3.1, the direction $d_k$ generated at every nonterminating iteration is a $K$-descent direction at $x_k$ and satisfies a uniform sufficient descent bound that is independent of the iteration index. Combining this result with (3.1) also gives

$$-H(x_k, d_k) \geq \chi a_k \geq \chi m_L \|p_k\|^2. \tag{3.12}$$

Estimate (3.12) provides an important foundation for establishing the global convergence results in the next section.

### *3.3. The DVM-R3TCG algorithm*

**Algorithm 3.1. The DVM-R3TCG algorithm**

**Step 0.** Choose an initial point $x_0 \in \mathbb{R}^n$, set $U_0 = I$, and choose a stopping tolerance $\varepsilon \geq 0$. Select parameters satisfying $0 < \rho < \sigma < 1$, $\mu > \frac{1}{2}$, $0 < \widehat{\mu} < 1$, $0 < \xi < 1$, $\eta > 0$, $\psi > 0$, $0 \leq \zeta_k \leq \bar{\zeta} < 1$, $\widehat{\mu} + \bar{\zeta} < 1$, $\omega > 0$, and $0 < m_L \leq 1 \leq m_U$. Take the normalized set $Z$, the vector $e \in \text{int}(K)$, and $\lambda^0 \in \text{int}(K^*)$ defined in Section 2. Set $k := 0$.

**Step 1.** Compute $p_k = v(x_k; U_k)$ and $\theta_k = \theta(x_k; U_k)$. If $|\theta_k| \leq \varepsilon$, output $x_k$ and stop. Otherwise, proceed to Step 2.

**Step 2.** If $k = 0$, set $d_0 = p_0$. If $k \geq 1$, compute $S_k$, $\delta_k$, $N_k$, $W_k$, $\beta_k$, and $r_k$ according to (3.2)–(3.5). Then compute the candidate direction $\widetilde{d}_k$ and the final search direction $d_k$ according to (3.7)–(3.10).

**Step 3.** Use the standard vector Wolfe line search to determine $\alpha_k > 0$ and set

$$x_{k+1} = x_k + \alpha_k d_k. \tag{3.13}$$

**Step 4.** Compute $s_k$ and $y_k$ according to (2.5). Then update $U_{k+1}$ using the regularized diagonal variable-metric formula given in Section 2.

**Step 5.** Set $k := k + 1$ and return to Step 1.

**Remark 3.1.** *By Lemma 2.3, $\theta_k = 0$ if and only if $p_k = 0$, which is also equivalent to $x_k$ being $K$-critical. Since $\theta_k \leq 0$, for $\varepsilon > 0$, the condition $|\theta_k| \leq \varepsilon$ is equivalent to $\theta_k \geq -\varepsilon$. Thus, the stopping criterion in Algorithm 3.1 provides a natural approximate criticality condition for the numerical implementation.*

## 4. Global convergence

This section investigates the global convergence properties of the DVM-R3TCG method. We consider the case in which the algorithm generates an infinite sequence of iterates $\{x_k\}$, and we assume that the line search at each iteration returns a step length satisfying the standard vector Wolfe conditions.

We make the following assumptions to establish the convergence results.

**Assumption 4.1.** The initial level set

$$\mathcal{L} := \{x \in \mathbb{R}^n \mid F(x) \preceq_K F(x_0)\}$$

is bounded, and $F$ is bounded below on $\mathcal{L}$ with respect to the cone order. More precisely, there exists $F^\ell \in \mathbb{R}^m$ such that

$$F^\ell \preceq_K F(x), \qquad x \in \mathcal{L}.$$

**Assumption 4.2.** There exist an open set $\Omega \subset \mathbb{R}^n$ containing $\mathcal{L}$ and a constant $L_J > 0$ such that

$$\|JF(x) - JF(y)\| \leq L_J\|x - y\|, \qquad x, y \in \Omega.$$

By Proposition 3.1, every search direction generated at a nonterminating iteration is a $K$-descent direction. Together with Proposition 2.1, this shows that a standard vector Wolfe step length always exists under Assumptions 4.1–4.2.

**Lemma 4.1.** *Under Assumptions 4.1–4.2, all iterates belong to $\mathcal{L}$. Moreover, the sequences $\{JF(x_k)\}$, $\{p_k\}$, $\{\theta_k\}$, and $\{a_k\}$ are bounded, where $a_k := -H(x_k, p_k)$.*

*Proof.* The Wolfe sufficient decrease condition and $H(x_k, d_k) < 0$ give

$$F(x_{k+1}) \preceq_K F(x_k) + \rho\alpha_k H(x_k, d_k)e \preceq_K F(x_k).$$

Induction therefore yields

$$F(x_k) \preceq_K F(x_0), \quad \text{for all } k \geq 0,$$

and hence, $x_k \in \mathcal{L}$ for all $k \geq 0$.

Since $F$ is continuous and $K$ is closed, the set $\mathcal{L}$ is closed. Assumption 4.1 also states that $\mathcal{L}$ is bounded, so $\mathcal{L}$ is compact. Assumption 4.2 then implies that $JF$ is bounded on $\mathcal{L}$. Consequently, the sequence $\{JF(x_k)\}$ is bounded.

By Lemma 2.2,

$$m_L I \leq U_k \leq m_U I, \quad \text{for all } k \geq 0.$$

Thus, $\{U_k\} \subset \mathcal{U}$, where $\mathcal{U}$ is compact. The continuity of the mapping $(x, U) \longmapsto v(x; U)$ and the function $(x, U) \longmapsto \theta(x; U)$ established in Lemma 2.3 implies that the sequences $\{p_k\}$ and $\{\theta_k\}$ are bounded. Finally, the identity

$$a_k = -2\theta_k = p_k^T U_k p_k$$

implies that $\{a_k\}$ is bounded. □

Define

$$c_e := \min_{\lambda \in Z} \langle \lambda, e \rangle > 0.$$

**Theorem 4.1.** *Under Assumptions 4.1–4.2, if the step length at every iteration satisfies the standard vector Wolfe conditions, then*

$$\sum_{k=0}^{\infty} \frac{H(x_k, d_k)^2}{\|d_k\|^2} < +\infty. \tag{4.1}$$

*Proof.* By the Wolfe sufficient decrease condition, for each fixed $\lambda \in Z$, we have

$$\begin{aligned} \langle \lambda, F(x_{k+1}) \rangle &\leq \langle \lambda, F(x_k) \rangle + \rho\alpha_k H(x_k, d_k)\langle \lambda, e \rangle \\ &\leq \langle \lambda, F(x_k) \rangle + \rho c_e \alpha_k H(x_k, d_k). \end{aligned}$$

The last inequality follows from $H(x_k, d_k) < 0$ and $\langle \lambda, e \rangle \geq c_e$. Therefore,

$$\rho c_e \alpha_k [-H(x_k, d_k)] \leq \langle \lambda, F(x_k) - F(x_{k+1}) \rangle.$$

Summing this inequality from $k = 0$ to $N$ gives

$$\rho c_e \sum_{k=0}^{N} \alpha_k[-H(x_k, d_k)] \leq \langle \lambda, F(x_0) - F(x_{N+1}) \rangle.$$

By Assumption 4.1, there exists $F^\ell$ such that $F^\ell \preceq_K F(x_{N+1})$, which implies

$$\langle \lambda, F(x_{N+1}) \rangle \geq \langle \lambda, F^\ell \rangle.$$

Hence, the right-hand side is uniformly bounded above. Letting $N \to \infty$ yields

$$\sum_{k=0}^{\infty} \alpha_k[-H(x_k, d_k)] < +\infty.$$

The Wolfe curvature condition gives $H(x_{k+1}, d_k) \geq \sigma H(x_k, d_k)$. It follows that

$$(1 - \sigma)[-H(x_k, d_k)] \leq H(x_{k+1}, d_k) - H(x_k, d_k).$$

By Lemma 2.1 and Assumption 4.2,

$$\begin{aligned} H(x_{k+1}, d_k) - H(x_k, d_k) &\leq \|JF(x_{k+1}) - JF(x_k)\| \, \|d_k\| \\ &\leq L_J \|x_{k+1} - x_k\| \, \|d_k\| \\ &= L_J \alpha_k \|d_k\|^2. \end{aligned}$$

Therefore,

$$\alpha_k \geq \frac{(1 - \sigma)[-H(x_k, d_k)]}{L_J \|d_k\|^2}.$$

Substituting this estimate into the preceding summable series gives

$$\sum_{k=0}^{\infty} \frac{H(x_k, d_k)^2}{\|d_k\|^2} < +\infty.$$

□

We next prove the main convergence result. Under the contradiction hypothesis that the criticality measure remains bounded away from zero, Lemma 4.2 first shows that the historical coefficient satisfies $|\beta_k| \leq C_\beta \|s_{k-1}\|$. Using this estimate, Lemma 4.3 gives the recursive representation $d_k = c_k + b_k d_{k-1}$, where $\{c_k\}$ is bounded and $b_k \leq C_\beta \|s_{k-1}\|$. Lemma 4.4 then combines this representation with Theorem 4.1 to obtain $\sum_{k=k_\gamma}^{\infty} 1/\|d_k\|^2 < +\infty$ and the square summability of the successive changes in the normalized search directions $z_k = d_k/\|d_k\|$. The latter implies that the normalized search directions become nearly aligned over any fixed block of sufficiently late iterations. The boundedness of the level set then provides a uniform bound on the accumulated step lengths over such a block, which in turn yields a uniform bound on the corresponding sum of the recursive coefficients $b_k$. Combining this bound with the recursion in Lemma 4.3 shows that $\{d_k\}$ is eventually bounded, contradicting the first summability result in Lemma 4.4. This is the basic idea underlying the convergence proof of Theorem 4.2.

We now make the above argument rigorous. Suppose, to the contrary, that there exist a constant $\gamma > 0$ and an integer $k_\gamma$ such that

$$\|p_k\| \geq \gamma, \quad \text{for all } k \geq k_\gamma. \tag{4.2}$$

It follows from (2.11) and Lemma 4.1 that there exist constants satisfying

$$0 < a_{\min} \leq a_{\max} < +\infty$$

such that

$$a_{\min} \leq a_k \leq a_{\max}, \quad \text{for all } k \geq k_\gamma.$$

In particular, one may take

$$a_{\min} = m_L \gamma^2.$$

**Lemma 4.2.** *If* (4.2) *holds, then there exists a constant* $C_\beta > 0$ *such that, for all sufficiently large k,*

$$|\beta_k| \leq C_\beta \|s_{k-1}\|, \qquad s_{k-1} := x_k - x_{k-1}. \tag{4.3}$$

*Proof.* Since $a_k = -2\theta(x_k; U_k)$, Lemma 2.4 and (2.7) give

$$\begin{aligned} |a_k - a_{k-1}| &\leq 2L_\theta \left(\|s_{k-1}\| + \|U_k - U_{k-1}\|_F\right) \\ &\leq 2L_\theta \left(\|s_{k-1}\| + C_U \|s_{k-1}\|^2\right). \end{aligned}$$

Since $\{x_k\} \subset \mathcal{L}$, the sequence $\{s_k\}$ is bounded. Therefore, there exists a constant $C_a > 0$ such that

$$|a_k - a_{k-1}| \leq C_a \|s_{k-1}\|.$$

Let

$$E_k := H(x_k, p_{k-1}) - H(x_{k-1}, p_{k-1}).$$

By Lemma 2.1, Assumption 4.2, and the boundedness of $\{p_k\}$, there exists a constant $C_E > 0$ such that

$$|E_k| \leq L_J \|s_{k-1}\| \, \|p_{k-1}\| \leq C_E \|s_{k-1}\|.$$

It follows from (3.3) and $H(x_{k-1}, p_{k-1}) = -a_{k-1}$ that

$$N_k = a_k - \delta_k a_{k-1} + \frac{\delta_k}{2} E_k.$$

Moreover,

$$\delta_k = \sqrt{\frac{a_k}{a_{k-1}}},$$

and hence,

$$\begin{aligned} |a_k - \delta_k a_{k-1}| &= \left|a_k - \sqrt{a_k a_{k-1}}\right| \\ &= \frac{\sqrt{a_k}}{\sqrt{a_k} + \sqrt{a_{k-1}}} |a_k - a_{k-1}| \\ &\leq |a_k - a_{k-1}|. \end{aligned}$$

The uniform upper and lower bounds on $\{a_k\}$ imply that $\{\delta_k\}$ is bounded. Therefore, there exists a constant $C_N > 0$ such that

$$|N_k| \leq C_N \|s_{k-1}\|.$$

It also follows from (3.4) that

$$W_k \geq T_{k,2} = a_{k-1} \geq a_{\min}.$$

Consequently,

$$|\beta_k| = \frac{|N_k|}{W_k} \leq \frac{C_N}{a_{\min}} \|s_{k-1}\|.$$

□

**Lemma 4.3.** *If* (4.2) *holds, then there exist constants $C_d > 0$ and $C_\beta > 0$, a bounded vector sequence $\{c_k\}$, and a nonnegative sequence $\{b_k\}$ such that, for all sufficiently large k,*

$$d_k = c_k + b_k d_{k-1}, \qquad \|c_k\| \leq C_d, \qquad 0 \leq b_k \leq C_\beta \|s_{k-1}\|. \tag{4.4}$$

*Proof.* Suppose first that the control condition (3.6) holds and the candidate direction is accepted as the final search direction. Then,

$$d_k = (1 - r_k)p_k + |\beta_k| d_{k-1}.$$

Set

$$c_k = (1 - r_k)p_k, \qquad b_k = |\beta_k|.$$

Since $0 \leq r_k < 1$, the sequence $\{p_k\}$ is bounded, and Lemma 4.2 holds, relation (4.4) follows.

We next consider the remaining branches. In each of these cases, set

$$b_k = 0, \qquad c_k = d_k.$$

It remains only to prove that $\{d_k\}$ is bounded.

Suppose first that the control condition (3.6) does not hold. Then

$$|\beta_k| S_k > \widehat{\mu} a_k > 0.$$

Thus, $S_k > 0$, which implies

$$H(x_k, d_{k-1}) > 0.$$

It follows from (3.7) that $q_k = p_k$. The safeguarded candidate direction therefore becomes

$$\widetilde{d_k} = \left(1 + \xi \frac{a_k}{\max\{a_k, \psi \|p_k\|\}}\right) p_k.$$

The coefficient in parentheses does not exceed $1 + \xi$, so $\{\widetilde{d_k}\}$ is bounded.

If the candidate direction is accepted as the final search direction, then

$$d_k = \widetilde{d_k},$$

and hence, $\{d_k\}$ is bounded.

Otherwise, the final direction uses the safeguarded correction

$$d_k = p_k + \bar{\beta}_k \widetilde{d}_k.$$

Then, (3.9) gives

$$|\bar{\beta}_k| \, \|\widetilde{d}_k\| \le \frac{a_k}{2\mu\psi}.$$

Since both $\{a_k\}$ and $\{p_k\}$ are bounded, the sequence $\{d_k\}$ is also bounded.

Finally, suppose that the control condition (3.6) holds but the candidate direction is not accepted as the final search direction. Then

$$d_k = p_k + \bar{\beta}_k \widetilde{d}_k.$$

Again, (3.9) gives

$$|\bar{\beta}_k| \, \|\widetilde{d}_k\| \le \frac{a_k}{2\mu\psi}.$$

Since both $\{a_k\}$ and $\{p_k\}$ are bounded, the sequence $\{d_k\}$ is bounded.

Therefore, in all remaining cases, we set $b_k = 0$ and $c_k = d_k$. The preceding estimates show that the sequence $\{c_k\}$ is bounded. Thus, (4.4) follows. □

Define

$$z_k := \frac{d_k}{\|d_k\|}.$$

**Lemma 4.4.** *If* (4.2) *holds, then*

$$\sum_{k=k_\gamma}^{\infty} \frac{1}{\|d_k\|^2} < +\infty, \qquad \sum_{k=k_\gamma+1}^{\infty} \|z_k - z_{k-1}\|^2 < +\infty. \tag{4.5}$$

*Proof.* By Proposition 3.1, (2.11), and (4.2),

$$\begin{aligned} -H(x_k, d_k) &\ge \chi[-H(x_k, p_k)] \\ &= \chi a_k \\ &\ge \chi m_L \gamma^2. \end{aligned}$$

Combining this estimate with Theorem 4.1 gives

$$\sum_{k=k_\gamma}^{\infty} \frac{1}{\|d_k\|^2} < +\infty.$$

We next prove the second conclusion. It follows from (4.4) that

$$d_k = c_k + b_k d_{k-1}, \qquad b_k \ge 0.$$

When $b_k > 0$, the standard estimate for normalized vectors gives

$$\left\| \frac{d_k}{\|d_k\|} - \frac{d_{k-1}}{\|d_{k-1}\|} \right\| \le \frac{2\|c_k\|}{\|d_k\|}.$$

When $b_k = 0$, we have $d_k = c_k$, and the same inequality remains valid. Therefore,

$$\|z_k - z_{k-1}\| \le \frac{2C_d}{\|d_k\|}.$$

Squaring both sides, summing over $k$, and using the first conclusion yields (4.5). □

**Theorem 4.2.** *Under Assumptions 4.1–4.2, suppose that the algorithm generates an infinite sequence of iterates and that every step length satisfies the standard vector Wolfe conditions. Then,*

$$\liminf_{k\to\infty} \|p_k\| = 0. \tag{4.6}$$

*Proof.* Suppose, to the contrary, that the conclusion does not hold. Then there exists a constant $\gamma > 0$ such that (4.2) holds. Consequently, Lemmas 4.2–4.4 are applicable.

Let the diameter of the level set $\mathcal{L}$ be

$$D_{\mathcal{L}} := \sup\{\|x - y\| \mid x, y \in \mathcal{L}\} < +\infty.$$

Fix an integer $\Delta \geq 1$. It follows from (4.5) that, for $j = 0, 1, \ldots, \Delta - 1$,

$$\begin{aligned} \|z_{k+j} - z_k\| &\leq \sum_{i=k+1}^{k+j} \|z_i - z_{i-1}\| \\ &\leq \sqrt{\Delta}\left(\sum_{i=k+1}^{k+\Delta-1} \|z_i - z_{i-1}\|^2\right)^{1/2} \longrightarrow 0 \qquad (k \to \infty). \end{aligned}$$

Therefore, when $k$ is sufficiently large,

$$\langle z_{k+j}, z_k\rangle \geq \frac{1}{2}, \qquad j = 0, 1, \ldots, \Delta - 1.$$

Since

$$s_i = x_{i+1} - x_i = \alpha_i d_i = \|s_i\| z_i, \qquad \alpha_i > 0,$$

we obtain

$$\begin{aligned} D_{\mathcal{L}} &\geq \|x_{k+\Delta} - x_k\| \\ &= \left\|\sum_{j=0}^{\Delta-1} \|s_{k+j}\| z_{k+j}\right\| \\ &\geq \left\langle \sum_{j=0}^{\Delta-1} \|s_{k+j}\| z_{k+j}, z_k \right\rangle \\ &\geq \frac{1}{2}\sum_{j=0}^{\Delta-1} \|s_{k+j}\|. \end{aligned}$$

Hence,

$$\sum_{j=0}^{\Delta-1} \|s_{k+j}\| \leq 2D_{\mathcal{L}}.$$

It follows from (4.4) that

$$\sum_{i=k+1}^{k+\Delta} b_i \leq C_\beta \sum_{j=0}^{\Delta-1} \|s_{k+j}\| \leq 2C_\beta D_{\mathcal{L}}.$$

Set

$$B := 2C_\beta D_{\mathcal{L}}.$$

Expanding the recursion (4.4) over $\Delta$ successive steps gives

$$\|d_{k+\Delta}\| \le A_\Delta + \left(\prod_{i=k+1}^{k+\Delta} b_i\right)\|d_k\|,$$

where $A_\Delta > 0$ is independent of $k$. The arithmetic–geometric mean inequality gives

$$\prod_{i=k+1}^{k+\Delta} b_i \le \left(\frac{1}{\Delta}\sum_{i=k+1}^{k+\Delta} b_i\right)^\Delta \le \left(\frac{B}{\Delta}\right)^\Delta.$$

Choose $\Delta$ sufficiently large so that

$$q := \left(\frac{B}{\Delta}\right)^\Delta < 1.$$

Then,

$$\|d_{k+\Delta}\| \le A_\Delta + q\|d_k\|.$$

Repeatedly applying this inequality along each subsequence with the same remainder modulo $\Delta$ shows that $\{\|d_k\|\}$ is eventually uniformly bounded. This would imply

$$\sum_{k=k_\gamma}^{\infty} \frac{1}{\|d_k\|^2} = +\infty,$$

which contradicts (4.5). Therefore, the assumption is false, and

$$\liminf_{k\to\infty} \|p_k\| = 0. \qquad \square$$

**Theorem 4.3.** *Under Assumptions 4.1–4.2, suppose that the algorithm generates an infinite sequence of iterates. Then $\{x_k\}$ has at least one $K$-critical accumulation point.*

*Proof.* By Theorem 4.2, there exists a subsequence $\{k_j\}$ such that

$$\|p_{k_j}\| \longrightarrow 0.$$

Since both $\mathcal{L}$ and $\mathcal{U}$ are compact, a further subsequence, still denoted by $\{k_j\}$, can be selected so that

$$x_{k_j} \longrightarrow x^*, \qquad U_{k_j} \longrightarrow U^* \in \mathcal{U}.$$

Since

$$p_{k_j} = v(x_{k_j}; U_{k_j}),$$

the continuity of $v(x; U)$ established in Lemma 2.3 gives

$$v(x^*; U^*) = \lim_{j\to\infty} p_{k_j} = 0.$$

Lemma 2.3 therefore shows that $x^*$ is $K$-critical. $\square$

**Corollary 4.4.** *Under Assumptions 4.1–4.2, for every fixed $\varepsilon > 0$, the DVM-R3TCG method reaches $|\theta_k| \le \varepsilon$ after finitely many iterations.*

*Proof.* Suppose that the algorithm does not terminate. Then $|\theta_k| > \varepsilon$ holds for every $k$. By Lemmas 2.2–2.3,

$$|\theta_k| = \frac{1}{2} p_k^T U_k p_k \leq \frac{m_U}{2} \|p_k\|^2,$$

and hence,

$$\|p_k\| > \sqrt{\frac{2\varepsilon}{m_U}}, \qquad \forall k.$$

Therefore,

$$\liminf_{k\to\infty} \|p_k\| \geq \sqrt{\frac{2\varepsilon}{m_U}} > 0.$$

This contradicts Theorem 4.2. Therefore, the algorithm must reach $|\theta_k| \leq \varepsilon$ after finitely many iterations. □

The preceding results show that when $\varepsilon = 0$, if the algorithm terminates after finitely many iterations, the termination point is $K$-critical; otherwise, the algorithm generates an infinite sequence satisfying $\liminf_{k\to\infty} \|p_k\| = 0$, and the sequence has at least one $K$-critical accumulation point. For any $\varepsilon > 0$, Corollary 4.4 shows that the algorithm reaches $|\theta_k| \leq \varepsilon$ and terminates after finitely many iterations.

The above conclusions can be strengthened under an additional regularity condition on the step lengths. If there exists a constant $\underline{\alpha} > 0$ such that $\alpha_k \geq \underline{\alpha}$, then the summability relation established in the proof of Theorem 4.1 together with (3.12) yields $\|p_k\| \to 0$. If, in addition, $L$ contains a unique $K$-critical point $x^*$, then the compactness of $L$ and the continuity of $v(x; U)$ in Lemma 2.3 imply $x_k \to x^*$.

Under the same additional condition, we next estimate the number of iterations required to attain prescribed criticality tolerances.

**Theorem 4.5.** *Under Assumptions 4.1–4.2, suppose that the step lengths satisfy the standard vector Wolfe conditions. Assume further that there exists a constant $\underline{\alpha} > 0$ such that $\alpha_k \geq \underline{\alpha}$ at every nonterminating iteration.*

For a prescribed norm-based criticality tolerance $\tau > 0$, let $T_\tau^p = \min\{k \geq 0 \mid \|p_k\| \leq \tau\}$. Then

$$T_\tau^p = O(\tau^{-2}).$$

Furthermore, for the stopping criterion $|\theta_k| \leq \varepsilon$ used in Algorithm 3.1, let $T_\varepsilon^\theta = \min\{k \geq 0 \mid |\theta_k| \leq \varepsilon\}$. Then,

$$T_\varepsilon^\theta = O(\varepsilon^{-1}).$$

*Proof.* Combining the summability relation established in the proof of Theorem 4.1 with (3.12), and using $\alpha_k \geq \underline{\alpha}$, we obtain

$$\sum_{k=0}^{N} \|p_k\|^2 \leq \frac{C_F}{\rho c_e \underline{\alpha} \chi m_L} = C_p,$$

where $C_F > 0$ is independent of $N$. Hence,

$$\min_{0\leq k\leq N} \|p_k\| \leq \sqrt{\frac{C_p}{N+1}}.$$

Therefore, whenever $N + 1 \geq C_p/\tau^2$, there exists an index $k \leq N$ such that $\|p_k\| \leq \tau$. Consequently,

$$T_\tau^p = O(\tau^{-2}).$$

Since $a_k = -2\theta_k = 2|\theta_k|$, the same argument gives

$$\sum_{k=0}^{N} |\theta_k| \leq \frac{C_F}{2\rho c_e \underline{\alpha}\underline{\chi}} = C_\theta.$$

It follows that

$$\min_{0\leq k\leq N} |\theta_k| \leq \frac{C_\theta}{N+1}.$$

Therefore,

$$T_\varepsilon^\theta = O(\varepsilon^{-1}).$$ □

## 5. Numerical experiments

This section investigates the computational performance of DVM-R3TCG through numerical experiments. We first compare the proposed method with the vector version of the nonnegative Polak–Ribière–Polyak method (VPRP+) [22], the modified Dai–Yuan conjugate gradient method (YDY) [30], the conjugate descent–Dai–Yuan hybrid conjugate gradient method (CD-DY) [31], and the vector steepest descent method (SD) [17]. The efficiency and robustness of the five methods are evaluated using Dolan–Moré performance profiles. We then examine the practical role of the diagonal variable-metric update through an ablation experiment in which this update is disabled. Parameter sensitivity, including the influence of the parameter $\mu$ and the initial trial step selection, is subsequently investigated. The impact of different stopping accuracies and the computational cost of the metric steepest-descent subproblem are also analysed. Finally, multi-start experiments are used to illustrate the objective-space behaviour of approximate $K$-critical points.

The test set consists of 36 classical problem instances covering convex and nonconvex problems, low- and high-dimensional problems, and biobjective and multiobjective problems. Several representative instances are selected for problems such as JOS1, MGH16, MGH27, TE8, and Toi10, whose dimensions or numbers of objectives can be varied. Throughout the experiments, we take

$$K = \mathbb{R}_+^m, \qquad Z = \{e_1, e_2, \ldots, e_m\},$$

where $e_i$ is the $i$th unit vector in $\mathbb{R}^m$.

Information on the test problems is given in Table 1. Here, $m$ and $n$ denote the number of objectives and the dimension of the decision variable, respectively. The vectors $lb$ and $ub$ specify the box from which the random starting points are generated,

$$\{x \in \mathbb{R}^n \mid lb \leq x \leq ub\}.$$

This box is used only to generate the starting points. The constraints are not explicitly handled during the iterations of the algorithms.

All algorithms were implemented in MATLAB R2022a and run in the same computing environment. For each test problem, 100 common starting points were generated uniformly at random from the box specified in Table 1. DVM-R3TCG, VPRP+, YDY, CD-DY, and SD were then run from the same starting points.

**Table 1.** Test problems.

| Problem | References | $m$ | $n$ | $lb^T$ | $ub^T$ |
|---|---|---|---|---|---|
| AP3 | [32] | 2 | 2 | $(-100, -100)$ | $(100, 100)$ |
| BK1 | [33] | 2 | 2 | $(-5, -5)$ | $(10, 10)$ |
| DGO1 | [33] | 2 | 1 | $-10$ | 13 |
| DGO2 | [33] | 2 | 1 | $-10$ | 13 |
| IKK1 | [33] | 3 | 2 | $(-50, -50)$ | $(50, 50)$ |
| JOS1-1 | [34] | 2 | 10 | $(0, \ldots, 0)$ | $(1, \ldots, 1)$ |
| JOS1-2 | [34] | 2 | 100 | $(0, \ldots, 0)$ | $(1, \ldots, 1)$ |
| JOS1-3 | [34] | 2 | 1000 | $(0, \ldots, 0)$ | $(1, \ldots, 1)$ |
| Lov5 | [35] | 2 | 3 | $(-2, -2, -2)$ | $(2, 2, 2)$ |
| MGH13 | [36] | 4 | 4 | $(-1, -1, -1, -1)$ | $(1, 1, 1, 1)$ |
| MGH16-1 | [36] | 10 | 4 | $(-25, -25, -25, -25)$ | $(25, 25, 25, 25)$ |
| MGH16-2 | [36] | 50 | 4 | $(-25, -25, -25, -25)$ | $(25, 25, 25, 25)$ |
| MGH16-3 | [36] | 100 | 4 | $(-25, -25, -25, -25)$ | $(25, 25, 25, 25)$ |
| MGH26 | [36] | 4 | 4 | $(-1, -1, -1, -1)$ | $(1, 1, 1, 1)$ |
| MGH27-1 | [36] | 50 | 10 | $(-1, \ldots, -1)$ | $(1, \ldots, 1)$ |
| MGH27-2 | [36] | 100 | 100 | $(-1, \ldots, -1)$ | $(1, \ldots, 1)$ |
| MGH27-3 | [36] | 1000 | 1000 | $(-1, \ldots, -1)$ | $(1, \ldots, 1)$ |
| MGH33 | [36] | 6 | 10 | $(-1, \ldots, -1)$ | $(1, \ldots, 1)$ |
| MLF1 | [33] | 2 | 1 | 0 | 20 |
| MOP1 | [33] | 2 | 1 | $-10^5$ | $10^5$ |
| MOP2 | [33] | 2 | 2 | $(-4, -4)$ | $(4, 4)$ |
| MOP3 | [33] | 2 | 2 | $(-\pi, -\pi)$ | $(\pi, \pi)$ |
| MOP5 | [33] | 3 | 2 | $(-30, -30)$ | $(30, 30)$ |
| MOP7 | [33] | 3 | 2 | $(-400, -400)$ | $(400, 400)$ |
| SK1 | [33] | 2 | 1 | $-100$ | 100 |
| SLCDT2 | [37] | 3 | 10 | $(-100, \ldots, -100)$ | $(100, \ldots, 100)$ |
| SP1 | [33] | 2 | 2 | $(-100, -100)$ | $(100, 100)$ |
| SSFYY2 | [33] | 2 | 1 | $-100$ | 100 |
| TE4 | [38] | 2 | 10 | $(-10, \ldots, -10)$ | $(10, \ldots, 10)$ |
| TE8-1 | [38] | 3 | 15 | $(0, \ldots, 0)$ | $(10, \ldots, 10)$ |
| TE8-2 | [38] | 3 | 30 | $(0, \ldots, 0)$ | $(1, \ldots, 1)$ |
| TE8-3 | [38] | 3 | 50 | $(0, \ldots, 0)$ | $(1, \ldots, 1)$ |
| Toi8 | [39] | 3 | 3 | $(-1, -1, -1)$ | $(1, 1, 1)$ |
| Toi10-1 | [39] | 9 | 10 | $(-2, \ldots, -2)$ | $(2, \ldots, 2)$ |
| Toi10-2 | [39] | 49 | 50 | $(-2, \ldots, -2)$ | $(2, \ldots, 2)$ |
| Toi10-3 | [39] | 99 | 100 | $(-2, \ldots, -2)$ | $(2, \ldots, 2)$ |

The parameters of DVM-R3TCG were chosen as $\mu = 0.6$, $\widehat{\mu} = 0.6$, $\eta = 10^{-4}$, $\zeta_k = 0.02$, $\xi = 0.5$, $\psi = 10^{-4}$, $\omega = 10^{-6}$, $m_L = 10^{-8}$, and $m_U = 10^8$. A run was terminated when $\theta_k \geq -5\sqrt{\varepsilon_{\text{mach}}}$ or when the number of iterations reached 5000, where $\varepsilon_{\text{mach}} = 2^{-52} \approx 2.22 \times 10^{-16}$. For a fair comparison, the five methods used the same objective scaling, stopping criterion, and line-search parameters. The

comparison methods were implemented according to their original references. VPRP+ [22], CD-DY [31], and SD [17] require no additional method-specific parameters. For YDY [30], we set $\mu_1 = 0.01$ and $\mu_2 = 0.1$, consistent with the numerical settings reported in [30].

The metric steepest descent direction $p_k = v(x_k; U_k)$ and the corresponding criticality value were computed using the MATLAB built-in function `quadprog`.

The step length was determined by the strong vector Wolfe conditions with $\rho = 10^{-4}$ and $\sigma = 0.1$. Since the strong vector Wolfe conditions imply the standard vector Wolfe conditions, this line-search strategy is compatible with the convergence analysis in Section 4.

The initial trial step length used in the line search was selected as follows. When $k = 0$, we used

$$\alpha_0^c = \min\left\{\max\left\{\frac{1}{\|p_0\|}, 1\right\}, \alpha_{\max}\right\}.$$

When $k \geq 1$, we used

$$\alpha_k^c = \min\left\{\max\left\{\frac{H(x_{k-1}, d_{k-1})}{H(x_k, d_k)}, 0.02\right\}, 100\right\},$$

where $\alpha_{\max} = 10^{10}$.

To reduce the effect of differences in the orders of magnitude of the objective components, the following positively scaled problem was used for each starting point $x_0$,

$$\min_{x\in\mathbb{R}^n} (\gamma_1 f_1(x), \gamma_2 f_2(x), \ldots, \gamma_m f_m(x))^T,$$

where

$$\gamma_j = \frac{1}{\max\left\{1, \|\nabla f_j(x_0)\|_\infty\right\}}, \qquad j = 1, \ldots, m.$$

When $K = \mathbb{R}^m_+$, positive scaling of the objective components does not change the Pareto order of the original problem. All algorithms used the same scaling factors for the same test problem and starting point.

Dolan–Moré performance profiles were used to compare the overall efficiency and robustness of the methods. Let $\mathcal{P}$ denote the set of test problems and let $\mathcal{A}$ denote the set of algorithms. The quantity $t_{p,a}$ denotes the median performance of algorithm $a \in \mathcal{A}$ on problem $p \in \mathcal{P}$ with respect to a given measure. Define

$$r_{p,a} = \frac{t_{p,a}}{\min\left\{t_{p,\bar{a}} \mid \bar{a} \in \mathcal{A}\right\}},$$

and

$$\rho_a(\tau) = \frac{1}{|\mathcal{P}|}\left|\left\{p \in \mathcal{P} \;\middle|\; r_{p,a} \leq \tau\right\}\right|.$$

If an algorithm has no successful run on a problem, the corresponding performance value is set to $+\infty$. The value $\rho_a(1)$ at the left end of the profile gives the proportion of problems for which the method attains the best or tied-best result. As $\tau$ increases, the height of the profile at the right end reflects the robustness of the method over the entire test set.

### *5.1. Efficiency and robustness analysis*

Table 2 presents the numerical results of the five methods on the test problems. The symbol "%" denotes the percentage of runs that successfully reached the stopping criterion. IT, NF, NG, NV, and CPU denote the median numbers of iterations, function evaluations, gradient evaluations, vector steepest descent evaluations, and the median CPU time, respectively. For problems with success rates below 100%, these median values were computed using only the successful runs. In the performance profiles, unsuccessful runs were assigned the value $+\infty$.

**Table 2.** Numerical results for the five methods.

| Problem | DVM-R3TCG<br>IT/NF/NG/NV/CPU/% | VPRP+<br>IT/NF/NG/NV/CPU/% | YDY<br>IT/NF/NG/NV/CPU/% | CD-DY<br>IT/NF/NG/NV/CPU/% | SD<br>IT/NF/NG/NV/CPU/% |
|---|---|---|---|---|---|
| AP3 | 2/207/17/3/0.0120/96 | 4/845/50/5/0.0565/97 | 3/358/34/4/0.0394/95 | 3/356/34/4/0.0433/96 | 3/358/34/4/0.0440/95 |
| BK1 | 6/342/89/6/0.0148/100 | 5/508/110/6/0.0535/100 | 5/428/86/6/0.0278/100 | 7/418/113/8/0.1127/100 | 5/427/86/6/0.0350/100 |
| DGO1 | 2/73/24/3/0.0064/100 | 2/73/27/3/0.0337/100 | 2/73/24/3/0.0153/100 | 2/72/25/3/0.0324/100 | 2/73/24/3/0.0162/100 |
| DGO2 | 3/270/54/4/0.0114/100 | 4/570/72/5/0.0489/100 | 3/266/59/4/0.0117/100 | 3/259/54/4/0.0400/100 | 3/266/59/4/0.0130/100 |
| IKK1 | 3/232/50/4/0.0100/100 | 3/253/52/4/0.0225/100 | 3/240/44/4/0.0185/100 | 3/234/44/4/0.0150/100 | 3/236/44/4/0.0182/100 |
| JOS1-1 | 4/880/42/5/0.0244/100 | 4/2211/81/5/0.0652/100 | 4/2211/75/5/0.0527/100 | 3/1518/51/4/0.0296/100 | 4/2211/75/5/0.0416/100 |
| JOS1-2 | 5/887/51/6/0.0399/100 | 4/2211/81/5/0.0787/100 | 4/2211/75/5/0.0756/100 | 3/1475/51/4/0.0527/100 | 4/2211/75/5/0.0688/100 |
| JOS1-3 | 6/935/58/7/1.1549/100 | 5/2947/107/6/1.8243/100 | 5/2947/99/6/1.8904/100 | 4/2211/75/5/1.5718/100 | 5/2947/99/6/1.7001/100 |
| Lov5 | 2/50/19/3/0.0104/100 | 2/50/21/3/0.0321/100 | 2/50/19/3/0.0257/100 | 2/50/19/3/0.0281/100 | 2/50/19/3/0.0237/100 |
| MGH13 | 4/149/84/5/0.0185/100 | 4/438/248/5/0.0236/100 | 4/153/88/5/0.0155/100 | 4/153/88/5/0.0224/100 | 4/153/88/5/0.0199/100 |
| MGH16-1 | 8/619/356/9/0.0722/100 | 12/1238/752/12/0.1451/100 | 27/1242/752/28/0.1689/100 | 9/675/388/10/0.0715/100 | 23/1144/680/24/0.1419/100 |
| MGH16-2 | 12/959/564/13/0.2292/100 | 20/1555/917/22/0.4081/100 | 14/864/510/15/0.2440/100 | 27/1222/724/28/0.3180/95 | 17/1060/632/18/0.2916/100 |
| MGH16-3 | 12/979/562/13/0.5053/100 | 17/1481/886/18/1.1814/100 | 16/1066/654/17/0.6291/100 | 26/1367/789/26/0.8726/96 | 16/989/588/17/0.6125/100 |
| MGH26 | 2/4/3/3/0.0173/92 | 2/11/12/3/0.0196/100 | 2/6/6/3/0.0138/94 | 2/6/6/3/0.0145/94 | 2/6/6/3/0.0133/94 |
| MGH27-1 | 1/2/1/2/0.0153/100 | 1/2/2/2/0.0193/100 | 1/2/2/2/0.0174/100 | 1/2/2/2/0.0180/100 | 1/2/2/2/0.0153/100 |
| MGH27-2 | 1/2/1/2/0.0248/100 | 1/2/2/2/0.0297/100 | 1/2/2/2/0.0263/100 | 1/2/2/2/0.0338/100 | 1/2/2/2/0.0223/100 |
| MGH27-3 | 1/2/1/2/2.2274/100 | 1/2/2/2/2.6654/100 | 1/2/2/2/2.3209/100 | 1/2/2/2/3.5481/100 | 1/2/2/2/2.9638/100 |
| MGH33 | 2/65/37/4/0.0112/100 | 2/105/60/3/0.0174/100 | 2/84/49/4/0.0127/100 | 2/85/52/4/0.0174/100 | 2/84/49/4/0.0129/100 |
| MLF1 | 1/3/3/2/0.0086/100 | 1/3/3/2/0.0163/100 | 1/3/3/2/0.0069/100 | 1/3/3/2/0.0168/100 | 1/3/3/2/0.0144/100 |
| MOP1 | 4/1740/65/5/0.0189/100 | 4/1953/78/5/0.0575/100 | 4/1848/69/5/0.0386/100 | 3/1307/49/4/0.0428/100 | 4/1830/69/5/0.0515/100 |
| MOP2 | 1/3/3/2/0.0079/100 | 1/3/3/2/0.0155/100 | 1/3/3/2/0.0120/100 | 1/3/3/2/0.0150/100 | 1/3/3/2/0.0168/100 |
| MOP3 | 6/237/78/7/0.0298/100 | 7/370/145/8/0.0764/100 | 8/310/114/10/0.0950/100 | 10/374/133/11/0.1373/100 | 8/306/118/10/0.1276/100 |
| MOP5 | 1/4/4/2/0.0071/100 | 1/4/4/2/0.0251/100 | 1/7/4/2/0.0076/100 | 1/4/4/2/0.0036/100 | 1/4/4/2/0.0179/100 |
| MOP7 | 6/443/100/7/0.0417/100 | 6/560/102/7/0.1458/100 | 16/1318/248/18/0.0976/100 | 6/562/96/7/0.0338/100 | 15/1269/236/16/0.2665/100 |
| SK1 | 2/171/24/3/0.0093/100 | 3/203/36/4/0.0395/100 | 3/203/29/4/0.0247/100 | 3/203/29/4/0.0392/100 | 3/203/29/4/0.0349/100 |
| SLCDT2 | 2/183/27/3/0.0170/100 | 4/888/56/5/0.0916/100 | 3/788/47/4/0.0276/100 | 3/752/47/4/0.0198/100 | 3/785/47/4/0.0525/100 |
| SP1 | 3/266/43/4/0.0156/100 | 7/470/90/8/0.0996/100 | 10/378/104/11/0.0924/100 | 7/361/75/8/0.0973/100 | 11/386/105/12/0.1312/100 |
| SSFYY2 | 3/198/38/4/0.0124/100 | 3/206/43/4/0.0364/100 | 3/196/38/4/0.0097/100 | 3/195/36/4/0.0359/100 | 3/197/38/4/0.0342/100 |
| TE4 | 7/350/92/8/0.0315/100 | 8/530/122/9/0.0327/100 | 15/514/150/16/0.1689/100 | 13/1019/212/14/0.1902/100 | 15/521/150/16/0.1872/100 |
| TE8-1 | 7/444/129/8/0.0459/100 | 7/895/195/8/0.0761/100 | 6/692/142/7/0.0534/100 | 12/750/200/14/0.0976/100 | 6/690/144/7/0.0904/100 |
| TE8-2 | 15/866/242/16/0.0877/100 | 16/1064/269/17/0.3243/100 | 16/1011/230/16/0.1120/100 | 300/13818/3046/300/2.4492/100 | 16/1022/226/17/0.1868/100 |
| TE8-3 | 16/953/254/16/0.1100/100 | 16/1181/288/17/0.2896/100 | 16/1157/246/17/0.1226/100 | 332/16827/3530/332/2.2390/100 | 16/1150/243/17/0.2385/100 |
| Toi8 | 3/133/46/4/0.0227/100 | 4/452/76/5/0.1041/100 | 4/148/44/4/0.0200/100 | 4/142/44/5/0.0181/100 | 4/144/44/4/0.0544/100 |
| Toi10-1 | 14/1784/1006/15/0.1505/100 | 18/3056/1673/19/0.2271/95 | 220/11134/6460/221/1.0638/100 | 12/1707/965/13/0.0810/98 | 208/9256/5512/210/0.9559/100 |
| Toi10-2 | 24/3438/1938/26/0.5652/100 | 33/5308/3161/34/0.9541/93 | 428/14754/8745/429/5.0742/99 | 23/2996/1691/24/0.5377/95 | 414/14444/8389/415/5.8190/99 |
| Toi10-3 | 26/3348/1886/27/1.5554/100 | 31/5078/2881/32/2.5256/90 | 285/13982/8141/286/13.2655/99 | 27/3339/1905/28/1.6598/100 | 256/13774/8119/257/11.6158/99 |

Table 2 shows that DVM-R3TCG reaches the stopping criterion reliably on the vast majority of the test problems. Its average success rate is 99.67%. The method achieves a success rate of 100% on 34 problems, with AP3 and MGH26 being the only exceptions. The success rates on AP3 and MGH26 are 96% and 92%, respectively, indicating that the numerical behaviour of these individual problems is still affected by the starting points and the problem structure.

In terms of computational cost, DVM-R3TCG reaches the stopping criterion with fewer function evaluations, fewer gradient evaluations, or less CPU time on many problems. The proposed method demonstrates favourable overall efficiency on the MGH16 series, MOP3, SLCDT2, SP1, TE4, and TE8. In terms of CPU time, DVM-R3TCG achieves the smallest or tied-smallest value on 25 of the 36 test problems. Based on the geometric mean of the problem-wise CPU-time ratios, its relative CPU time is reduced by approximately 58.0%, 45.1%, 51.4%, and 57.1% compared with VPRP+, YDY, CD-DY, and SD, respectively. The lower CPU time is generally accompanied by fewer function

and gradient evaluations, indicating that the improvement is consistent with a reduction in the actual computational effort during the search. Therefore, for multi-start or repeated-solve vector optimization problems, the lower cost of an individual run can further reduce the overall computational burden.

For a clearer overall comparison of the five methods, Figure 1 shows performance profiles based on IT, NF, NG, NV, and CPU. Table 2 and Figure 1 serve different purposes. The table reports the specific median results for each test problem, whereas the performance profiles reflect the overall efficiency and robustness of the methods over the complete test set.

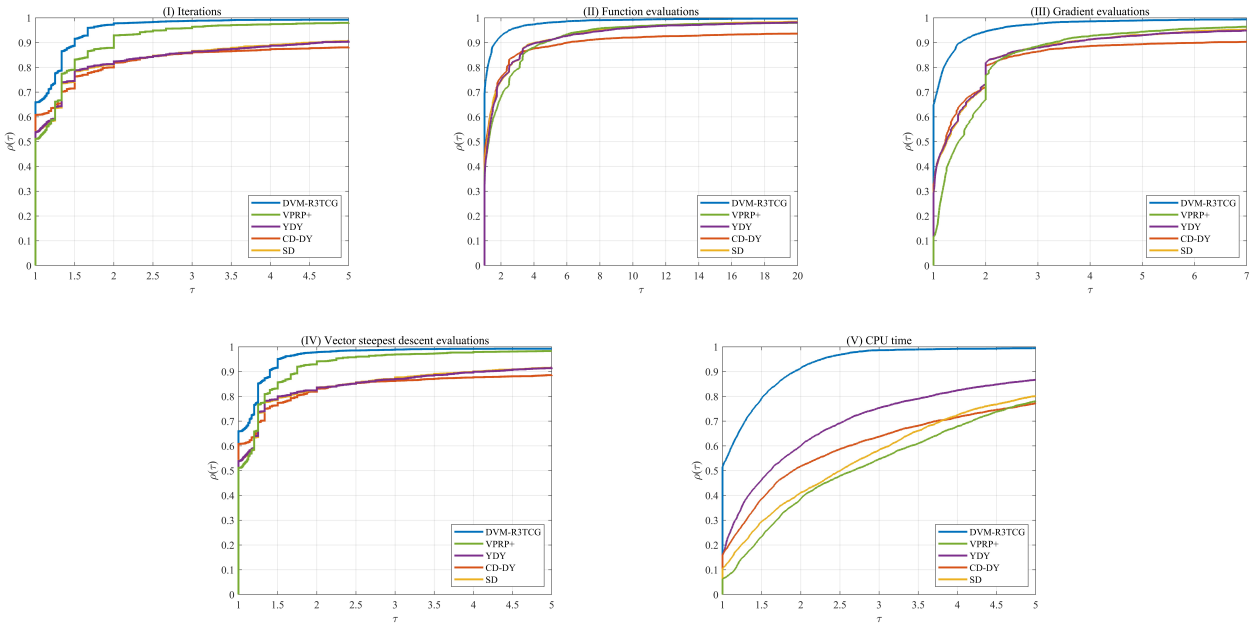


**Figure 1.** Performance profiles of the five methods over 100 starting points for each test problem.

Figure 1 shows that the performance profiles of DVM-R3TCG generally occupy favourable positions for all five measures. This indicates that the method attains the best or nearly the best performance on a large proportion of the test problems. In particular, the DVM-R3TCG profiles remain at relatively high levels as $\tau$ increases. This shows that the proposed method is not only efficient on individual problems but also exhibits stable solution performance over the entire test set.

Taken together, Table 2 and Figure 1 show that the advantages of DVM-R3TCG are not confined to a single performance measure. The method achieves a favourable overall balance in terms of the number of iterations, function and gradient evaluations, vector steepest descent subproblem evaluations, and CPU time.

To further investigate the role of the diagonal variable-metric module, we construct DVM-R3TCG-noDBB as an ablated variant, in which $U_k = I$ for all $k \geq 0$. This setting disables the diagonal variable-metric update in (2.3) and (2.4). Its search direction construction, safeguarded selection mechanism, line search, objective scaling, starting points, and stopping criterion are identical to those of the complete method. Table 3 reports the complete numerical results for DVM-R3TCG and DVM-R3TCG-noDBB.

Table 3 shows that both DVM-R3TCG and DVM-R3TCG-noDBB have high run success rates.

Their average success rates are 99.67% and 99.69%, respectively, and the difference is small. This indicates that the diagonal variable-metric module has no marked effect on whether the algorithm reaches the stopping criterion. Its main role is reflected in improved computational efficiency.

**Table 3.** Ablation results for the diagonal variable metric.

| Problem | DVM-R3TCG IT/NF/NG/NV/CPU/% | DVM-R3TCG-noDBB IT/NF/NG/NV/CPU/% | Problem | DVM-R3TCG IT/NF/NG/NV/CPU/% | DVM-R3TCG-noDBB IT/NF/NG/NV/CPU/% |
|---|---|---|---|---|---|
| AP3 | 2/207/17/3/0.0120/96 | 3/333/31/4/0.0216/95 | MLF1 | 1/3/3/2/0.0086/100 | 1/3/3/2/0.0081/100 |
| BK1 | 6/342/89/6/0.0148/100 | 8/394/116/9/0.0230/100 | MOP1 | 4/1740/65/5/0.0189/100 | 4/1869/69/5/0.0263/100 |
| DGO1 | 2/73/24/3/0.0064/100 | 2/88/26/3/0.0074/100 | MOP2 | 1/3/3/2/0.0079/100 | 1/3/3/2/0.0075/100 |
| DGO2 | 3/270/54/4/0.0114/100 | 3/280/60/4/0.0194/100 | MOP3 | 6/237/78/7/0.0298/100 | 9/300/114/10/0.0555/100 |
| IKK1 | 3/232/50/4/0.0100/100 | 3/234/48/4/0.0091/100 | MOP5 | 1/4/4/2/0.0071/100 | 1/4/4/2/0.0073/100 |
| JOS1-1 | 4/880/42/5/0.0244/100 | 4/2297/78/5/0.0293/100 | MOP7 | 6/443/100/7/0.0417/100 | 7/634/122/8/0.0495/100 |
| JOS1-2 | 5/887/51/6/0.0399/100 | 5/2992/103/6/0.0397/100 | SK1 | 2/171/24/3/0.0093/100 | 3/203/30/4/0.0196/100 |
| JOS1-3 | 6/935/58/7/1.1549/100 | 5/2992/103/6/0.9486/100 | SLCDT2 | 2/183/27/3/0.0170/100 | 3/756/55/4/0.0271/100 |
| Lov5 | 2/50/19/3/0.0104/100 | 2/50/19/3/0.0139/100 | SP1 | 3/266/43/4/0.0156/100 | 7/347/84/8/0.0468/100 |
| MGH13 | 4/149/84/5/0.0185/100 | 4/154/88/5/0.0167/100 | SSFYY2 | 3/198/38/4/0.0124/100 | 3/199/38/4/0.0196/100 |
| MGH16-1 | 8/619/356/9/0.0722/100 | 8/618/354/9/0.1129/100 | TE4 | 7/350/92/8/0.0315/100 | 19/606/194/20/0.1211/100 |
| MGH16-2 | 12/959/564/13/0.2292/100 | 12/959/564/13/0.2783/100 | TE8-1 | 7/444/129/8/0.0459/100 | 9/766/204/10/0.0542/100 |
| MGH16-3 | 12/979/562/13/0.5053/100 | 12/979/562/13/0.5354/100 | TE8-2 | 15/866/242/16/0.0877/100 | 18/954/330/19/0.1007/100 |
| MGH26 | 2/4/3/3/0.0173/92 | 2/6/4/3/0.0304/94 | TE8-3 | 16/953/254/16/0.1100/100 | 19/1008/344/20/0.1352/100 |
| MGH27-1 | 1/2/1/2/0.0153/100 | 1/2/1/2/0.0155/100 | Toi8 | 3/133/46/4/0.0227/100 | 3/155/54/4/0.0228/100 |
| MGH27-2 | 1/2/1/2/0.0248/100 | 1/2/1/2/0.0262/100 | Toi10-1 | 14/1784/1006/15/0.1505/100 | 12/1822/1022/14/0.1442/100 |
| MGH27-3 | 1/2/1/2/2.2274/100 | 1/2/1/2/2.2300/100 | Toi10-2 | 24/3438/1938/26/0.5652/100 | 28/3127/1774/29/0.5452/100 |
| MGH33 | 2/65/37/4/0.0112/100 | 2/89/50/4/0.0228/100 | Toi10-3 | 26/3348/1886/27/1.5554/100 | 28/3212/1824/28/1.5153/100 |

Among the 36 test problems, DVM-R3TCG requires fewer function evaluations on 24 problems, fewer gradient evaluations on 22 problems, and less CPU time on 27 problems. For IT and NV, the two methods obtain identical results on 21 problems in each case. This indicates that the diagonal variable metric does not significantly change the number of outer iterations or vector steepest descent subproblem evaluations on some problems. It can nevertheless reduce the number of function and gradient evaluations during the search process by improving the agreement between the search direction and the local coordinate scaling. This effect is particularly evident on BK1, JOS1-1, MOP3, MOP7, SLCDT2, SP1, TE4, and the TE8 series.

Figure 2 presents the performance profiles of the two methods based on IT, NF, NG, NV, and CPU.

Figure 2 shows that the profiles of DVM-R3TCG generally lie above those of DVM-R3TCG-noDBB for all five measures. The differences between the two methods are relatively small for IT and NV, whereas they are more pronounced for NF, NG, and CPU. In particular, the CPU profile of DVM-R3TCG maintains higher performance-profile values over a wider range. This observation is consistent with the problem-by-problem results in Table 3 and further indicates that the diagonal variable metric improves the overall efficiency of the algorithm mainly by reducing the numbers of function and gradient evaluations and the computing time.

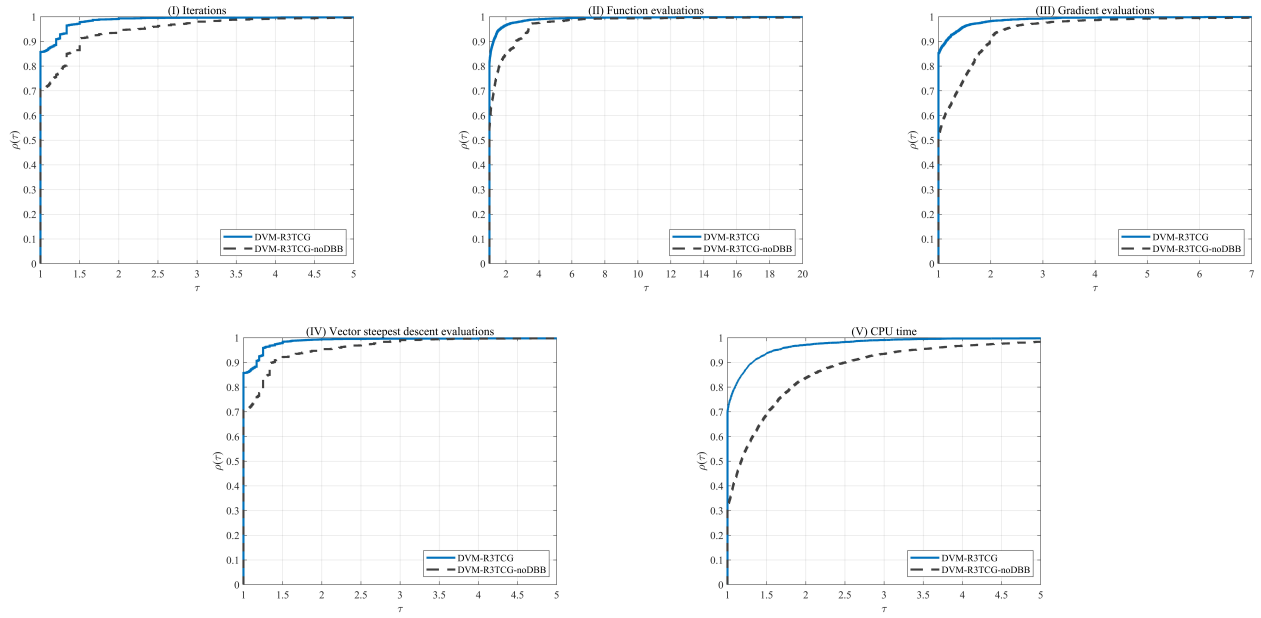


**Figure 2.** Performance profiles of DVM-R3TCG and DVM-R3TCG-noDBB over 100 starting points for each test problem.

### 5.2. *Sensitivity and computational-cost analysis*

To further examine the influence of the parameter $\mu$ on the numerical performance of DVM-R3TCG, we tested $\mu = 0.51$, $\mu = 0.60$, and $\mu = 10.00$ while keeping all the other parameter settings unchanged. For each test problem, the three parameter settings were applied to the same 100 randomly generated starting points. The value $\mu = 0.51$ is close to the lower bound imposed by $\mu > 1/2$, $\mu = 0.60$ is the value used in the main numerical experiments, and $\mu = 10.00$ represents a relatively large choice. Since $\chi = 1 - \frac{1}{2\mu}$, choosing $\mu$ close to $1/2$ makes the sufficient-descent safeguard less restrictive, whereas increasing $\mu$ strengthens the safeguard and leads to a more conservative direction selection.

Figure 3 shows that the performance profiles for the three values of $\mu$ are generally close in terms of IT, NF, NG, and NV, indicating that the choice of $\mu$ has only a limited influence on the main iteration process. Among the three settings, $\mu = 0.60$ shows a modest overall advantage. The difference is more pronounced for CPU time, where $\mu = 0.60$ provides the best overall computational efficiency. Although $\mu = 0.51$ gives a less restrictive sufficient-descent safeguard, it does not lead to a consistent improvement in numerical performance. Similarly, the more conservative choice $\mu = 10.00$ does not provide an overall advantage. These results indicate that $\mu = 0.60$ gives a favourable balance among the tested settings.

To further investigate the practical convergence behaviour with respect to $\mu$, we selected the representative problems JOS1-3, TE8-3, and MGH16-3 and plotted the criticality function $-\theta_k$ against the iteration number.

In Figure 4, the vertical axis is displayed on a logarithmic scale, and the horizontal dashed line indicates the stopping threshold used in the numerical computation. The criticality functions corresponding to the three values of $\mu$ exhibit very similar decreasing trends. Although some differences occur over individual stages of the iterations, no consistent acceleration or deceleration is observed as $\mu$ increases or decreases. This indicates that the choice of $\mu$ has only a limited effect on

the practical convergence speed of the method.

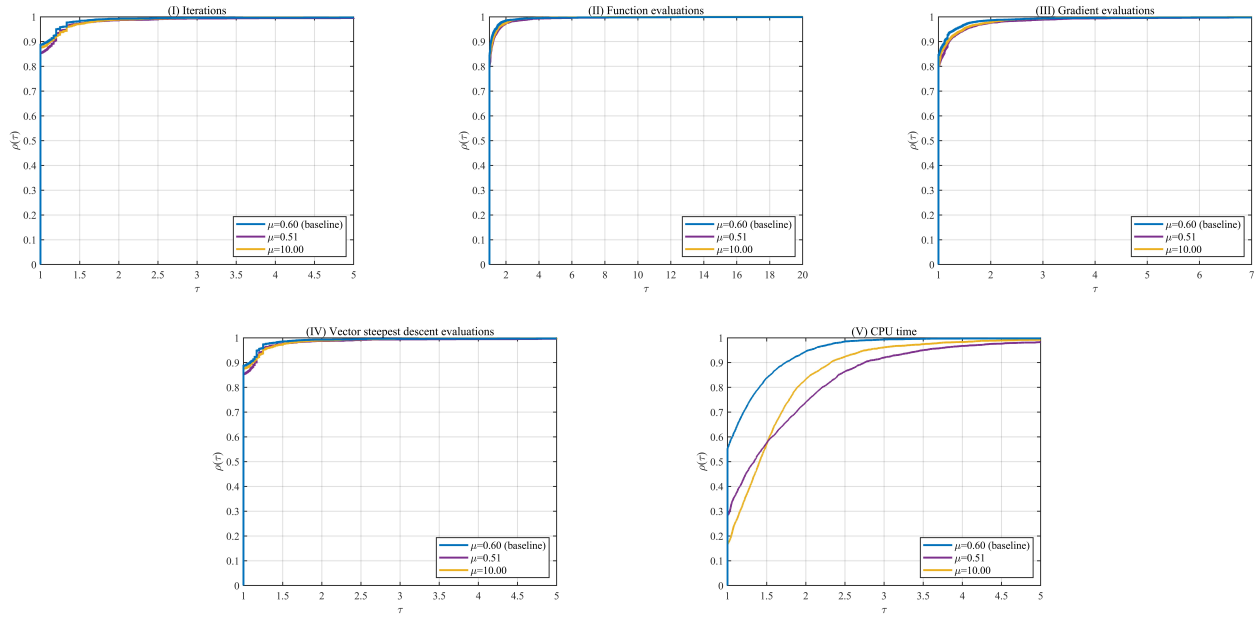


**Figure 3.** Performance profiles of DVM-R3TCG for different values of $\mu$.

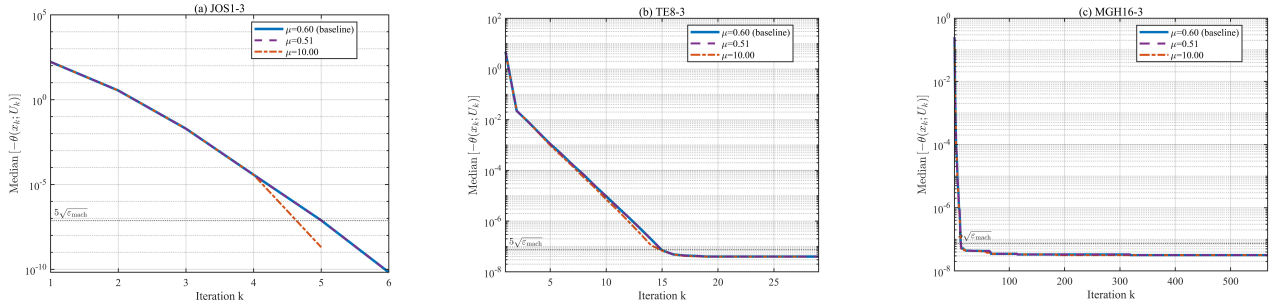


**Figure 4.** Variation of the criticality measure for DVM-R3TCG with different values of $\mu$.

Taken together, Figures 3 and 4 indicate that the overall performance of DVM-R3TCG is relatively insensitive to the choice of $\mu$, with the main difference appearing in CPU time. Among the tested settings, $\mu = 0.60$ provides the most favourable overall computational efficiency and is therefore retained as the default value of DVM-R3TCG.

To further examine the influence of the initial trial step on the numerical performance of DVM-R3TCG, we use the original rule

$$\alpha_0^c = \min\left\{\max\left\{\frac{1}{\|p_0\|}, 1\right\}, \alpha_{\max}\right\}$$

as the baseline, referred to as the *Original rule (baseline)* in Figure 5. We compare it with three alternative choices, namely the *fixed rule* $\alpha_0^c = 1$, the *norm-scaled rule* $\alpha_0^c = \min\{1/\|p_0\|, \alpha_{\max}\}$, and the *maximum rule* $\alpha_0^c = \alpha_{\max}$. Figure 5 presents the performance profiles of the four rules in terms of IT, NF, NG, NV, and CPU.

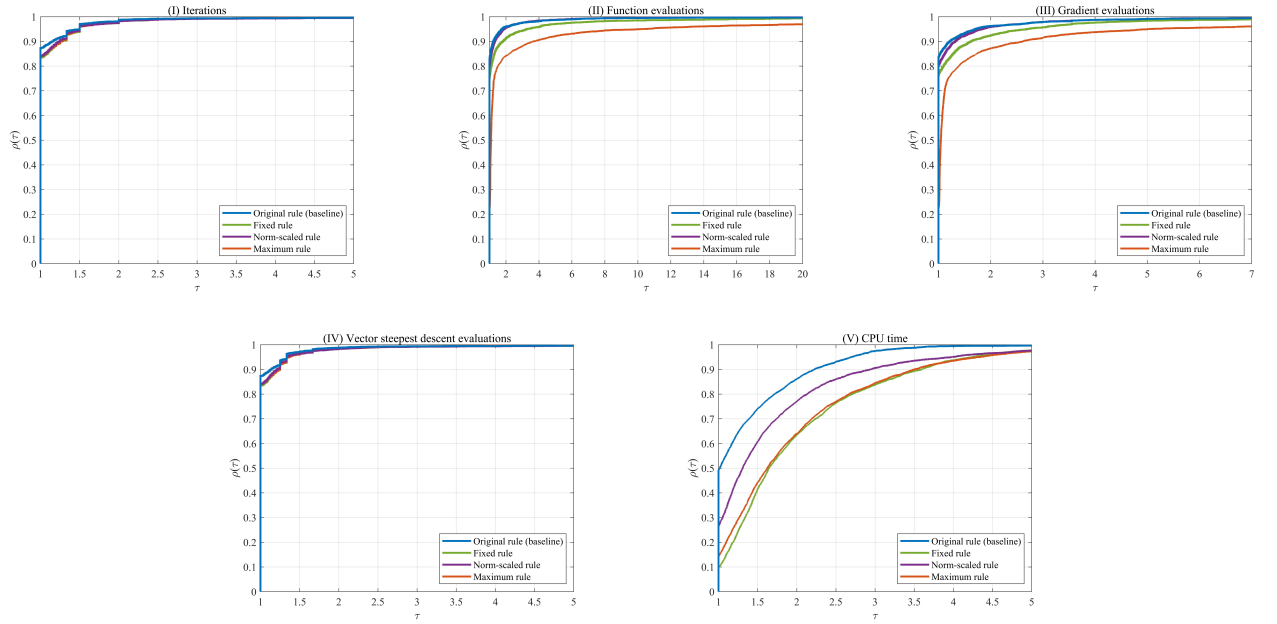


**Figure 5.** Performance profiles of DVM-R3TCG with different initial trial step rules.

Figure 5 shows that scaling the initial trial step according to $\|p_0\|$ is beneficial to the overall computational performance, while introducing the lower safeguard 1 further improves the performance of the norm-scaled rule. In contrast, directly using the large value $\alpha_{\max}$ leads to additional function and gradient evaluations associated with the line search, indicating that an excessively large initial trial step can increase the computational cost. The four rules show relatively small differences in IT and NV, which suggests that the initial trial step mainly affects the line-search process and has only a limited influence on the outer iterations. Overall, the original rule provides a favourable balance between scaling with the initial direction and safeguarding the trial step, and the numerical results support its use in the implementation.

To further examine the influence of the stopping accuracy on the numerical performance of DVM-R3TCG, we considered the termination condition $\theta_k \geq -c\sqrt{\varepsilon_{\text{mach}}}$ with $c \in \{1, 5, 10\}$, while keeping all other experimental settings unchanged. The value $c = 5$ is used in the main numerical experiments, whereas $c = 1$ and $c = 10$ correspond to stricter and looser stopping requirements, respectively. Table 4 reports the numbers of test problems for which each setting attains the smallest or tied-smallest values of IT, NF, NG, and NV.

**Table 4.** Sensitivity of DVM-R3TCG to the stopping parameter $c$.

| $c$ | IT-best | NF-best | NG-best | NV-best |
|---|---|---|---|---|
| 1 | 22/36 | 9/36 | 10/36 | 20/36 |
| 5 | 31/36 | 14/36 | 16/36 | 28/36 |
| 10 | 36/36 | 36/36 | 36/36 | 36/36 |

Here, "best" denotes the number of test problems for which the corresponding measure is the smallest or tied for the smallest among the three settings. As expected, a looser stopping requirement reduces the computational effort, whereas a stricter requirement requires more evaluations to attain a

smaller criticality level. The purpose of this experiment is therefore not to identify an optimal value of $c$, but to illustrate the computational trade-off associated with the stopping accuracy. The value $c = 5$ provides an intermediate accuracy level and is adopted in the main experiments. Moreover, the largest iteration counts among all successful runs were 1040, 829, and 743 for $c = 1$, $c = 5$, and $c = 10$, respectively, all of which are well below 5000. Thus, the maximum iteration count of 5000 mainly serves as a safeguard against excessively long unsuccessful runs.

Furthermore, to investigate the influence of the metric steepest descent quadratic programming subproblem in (2.8) on the computational efficiency of DVM-R3TCG, we record the CPU time consumed by MATLAB's `quadprog` for solving this subproblem and calculate its proportion in the total running time. For each test problem, the quadratic programming (QP) contribution reported in Table 5 is defined as the median value of $T_{\mathrm{QP}}/T_{\mathrm{total}} \times 100\%$ over all successful runs, where $T_{\mathrm{QP}}$ denotes the cumulative CPU time spent on solving all metric steepest descent QP subproblems and $T_{\mathrm{total}}$ denotes the total running time of DVM-R3TCG in one run. Since the absolute CPU time can be affected by the hardware configuration, software environment, and system load, we report the relative computational contribution of the QP solver rather than the absolute running time.

**Table 5.** Computational contribution of the metric steepest descent QP subproblem.

| Problem | QP contribution (%) | Problem | QP contribution (%) | Problem | QP contribution (%) |
|---|---|---|---|---|---|
| AP3 | 74.97 | MGH16-3 | 29.84 | SK1 | 71.41 |
| BK1 | 78.67 | MGH26 | 78.22 | SLCDT2 | 65.10 |
| DGO1 | 73.23 | MGH27-1 | 80.89 | SP1 | 75.45 |
| DGO2 | 80.86 | MGH27-2 | 87.64 | SSFYY2 | 79.06 |
| IKK1 | 66.62 | MGH27-3 | 98.72 | TE4 | 81.50 |
| JOS1-1 | 55.22 | MGH33 | 60.02 | TE8-1 | 78.97 |
| JOS1-2 | 53.17 | MLF1 | 68.40 | TE8-2 | 80.15 |
| JOS1-3 | 90.73 | MOP1 | 77.33 | TE8-3 | 76.82 |
| Lov5 | 71.87 | MOP2 | 68.17 | Toi8 | 77.06 |
| MGH13 | 72.69 | MOP3 | 82.19 | Toi10-1 | 57.84 |
| MGH16-1 | 46.98 | MOP5 | 69.15 | Toi10-2 | 33.67 |
| MGH16-2 | 26.16 | MOP7 | 82.00 | Toi10-3 | 33.33 |

As shown in Table 5, the QP contribution ranges from 26.16% to 98.72% among all test problems, indicating that the metric steepest descent QP subproblem is an important component of the computational cost of DVM-R3TCG. However, this ratio does not increase monotonically with the problem size, since the total running time also includes objective function evaluations, gradient evaluations, Wolfe line searches, and search direction updates. Therefore, the relative computational cost of the QP subproblem depends on both the problem characteristics and the cost of the other algorithmic components.

### *5.3. Objective-space behaviour of approximate K-critical points*

In addition to computational efficiency and robustness, we further use multi-start experiments to examine the search behaviour of DVM-R3TCG in the objective space. It should be noted that Section 4 establishes the global subsequential convergence of the algorithm and proves that the

generated sequence has at least one $K$-critical accumulation point. Accordingly, the terminal objective vectors shown below are interpreted as the objective-space distribution of approximate $K$-critical points. Although some of these points lie along apparent trade-off boundaries, the present convergence theory does not guarantee that they are globally Pareto efficient.

We select six representative problems, namely BK1, DGO1, DGO2, JOS1-1, MOP3, and MOP7. For each problem, 300 starting points are generated uniformly at random from the box specified in Table 1, and DVM-R3TCG is run from each starting point. The trajectories of the objective values generated during the iterations and their terminal points are plotted in the objective space. The results are shown in Figure 6.

In Figure 6, the black line segments represent the iterative trajectories of the algorithm in the objective-value space, while the red points represent the objective values obtained when the algorithm terminates. For the biobjective problems, the horizontal and vertical axes represent $f_1(x)$ and $f_2(x)$, respectively. For the triobjective problem MOP7, $f_1(x)$, $f_2(x)$, and $f_3(x)$ are displayed in the three-dimensional objective space.

Figure 6 shows that, when the algorithm starts from different initial points, the objective values generally move toward lower regions of the objective space. On several problems, the terminal points form continuous or piecewise continuous boundary contours. These observations illustrate the objective-space behaviour of the approximate $K$-critical points obtained from multi-start runs.

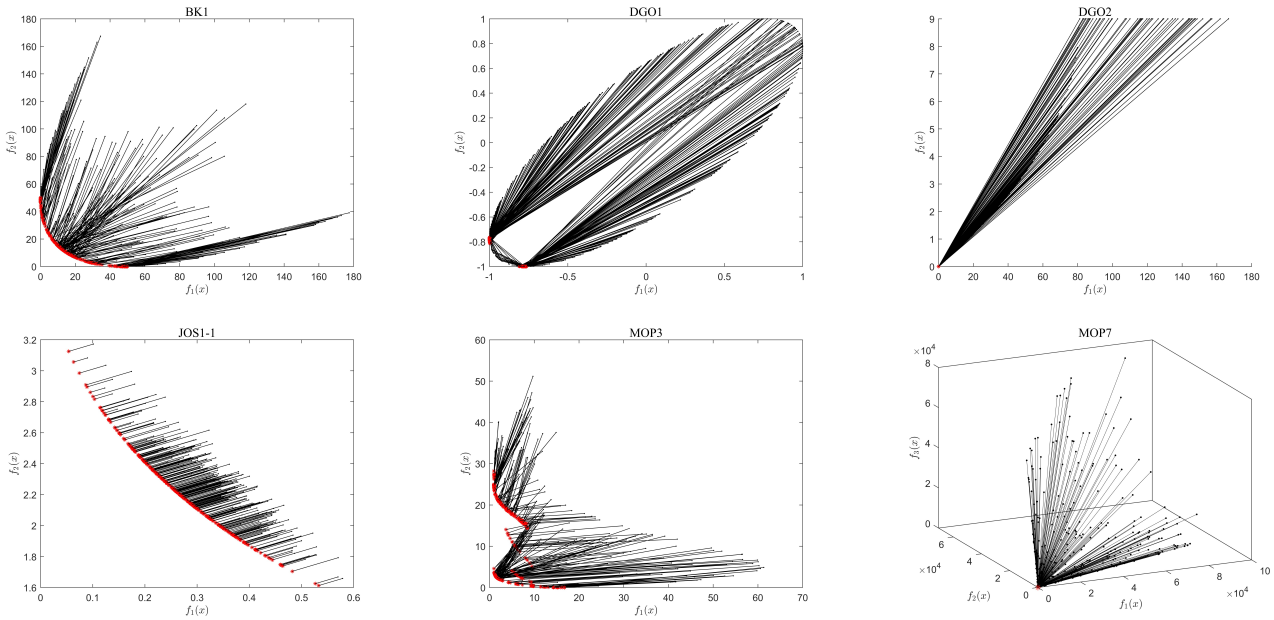


**Figure 6.** Objective-space trajectories and approximate $K$-critical points obtained by DVM-R3TCG from 300 random starting points.

## 6. Conclusions

This paper proposes a restarted three-term vector conjugate gradient method with a diagonal variable metric, referred to as DVM-R3TCG, for solving nonconvex unconstrained vector optimization problems. The method combines a diagonal variable metric, a restart mechanism, and a safeguarding

correction strategy so that the generated search directions satisfy a uniform sufficient descent condition. Under standard assumptions and a standard vector Wolfe line search, the global subsequential convergence of the method is established, and the generated sequence is shown to have at least one $K$-critical accumulation point. Numerical comparisons and ablation experiments demonstrate favourable computational efficiency and numerical stability of the proposed method. Multi-start experiments further illustrate the objective-space distribution of approximate $K$-critical points obtained from different initial points.

## Author contributions

Yuchang Zhang, Jing Gao, and Chongyang He: Methodology, Software, Visualization, Writing–original draft. All authors contributed equally to this work. All authors have read and approved the final version of the manuscript for publication.

## Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

The authors would like to thank the Natural Science Foundation of Jilin Province (YDZJ202501ZYTS653), the doctoral research project start-up fund of Beihua University, and the graduate innovation project of Beihua University (2025037).

## Conflict of interest

All authors declare no conflicts of interest in this paper.

## References

1. J. Jahn, A. Kirsch, C. Wagner, Optimization of rod antennas of mobile phones, *Math. Meth. Oper. Res.*, **59** (2004), 37–51. https://doi.org/10.1007/s001860300318
2. T. S. Hong, D. L. Craft, F. Carlsson, T. R. Bortfeld, Multicriteria optimization in intensity-modulated radiation therapy treatment planning for locally advanced cancer of the pancreatic head, *Int. J. Radiat. Oncol.*, **72** (2008), 1208–1214. https://doi.org/10.1016/j.ijrobp.2008.07.015
3. Y. Q. Wang, J. Zhu, H. P. Huang, F. Xiao, Bi-objective ant colony optimization for trajectory planning and task offloading in UAV-assisted MEC systems, *IEEE T. Mobile Comput.*, **23** (2024), 12360–12377. https://doi.org/10.1109/TMC.2024.3408603
4. M. Tavana, A subjective assessment of alternative mission architectures for the human exploration of Mars at NASA using multicriteria decision making, *Comput. Oper. Res.*, **31** (2004), 1147–1164. https://doi.org/10.1016/S0305-0548(03)00074-1

5. T. M. Leschine, H. Wallenius, W. A. Verdini, Interactive multiobjective analysis and assimilative capacity-based ocean disposal decisions, *Eur. J. Oper. Res.*, **56** (1992), 278–289. https://doi.org/10.1016/0377-2217(92)90228-2
6. D. J. White, Epsilon-dominating solutions in mean-variance portfolio analysis, *Eur. J. Oper. Res.*, **105** (1998), 457–466. https://doi.org/10.1016/S0377-2217(97)00056-8
7. G. Eichfelder, Multiobjective bilevel optimization, *Math. Program.*, **123** (2010), 419–449. https://doi.org/10.1007/s10107-008-0259-0
8. J. Fliege, L. N. Vicente, Multicriteria approach to bilevel optimization, *J. Optim. Theory Appl.*, **131** (2006), 209–225. https://doi.org/10.1007/s10957-006-9136-2
9. P. De, J. B. Ghosh, C. E. Wells, On the minimization of completion time variance with a bicriteria extension, *Oper. Res.*, **40** (1992), 1148–1155. https://doi.org/10.1287/opre.40.6.1148
10. C. B. Qin, K. L. Sun, A. Sun, D. H. Zhang, J. S. Zhang, G. H. Wang, et al., Dynamic event-triggered optimal safety control of interconnected nonlinear systems with discontinuous state constraints by adaptive dynamic programming, *Neurocomputing*, **669** (2026), 132467. https://doi.org/10.1016/j.neucom.2025.132467
11. C. B. Qin, M. Y. Pang, Z. W. Wang, S. Y. Hou, Q. Qiu, ADP-based dynamic event-triggered fault-tolerant control for nonlinear systems with actuator failures and full-state time-varying constraints, *Commun. Nonlinear Sci.*, **157** (2026), 109676. https://doi.org/10.1016/j.cnsns.2026.109676
12. Y. Y. Hu, J. Q. Chen, C. B. Qin, Event-triggered prescribed performance control of the multiplayer game nonlinear system via integral reinforcement learning, *Appl. Math. Comput.*, **511** (2026), 129716. https://doi.org/10.1016/j.amc.2025.129716
13. C. B. Qin, Z. W. Wang, S. Y. Hou, M. Y. Pang, G. H. Wang, Y. Wang, et al., Mixed zero-sum game based dynamic event-triggered optimal control of multi-input nonlinear system with safety constraints, *Commun. Nonlinear Sci.*, **152** (2026), 109376. https://doi.org/10.1016/j.cnsns.2025.109376
14. R. S. Burachik, C. Y. Kaya, M. M. Rizvi, A new scalarization technique and new algorithms to generate Pareto fronts, *SIAM J. Optim.*, **27** (2017), 1010–1034. https://doi.org/10.1137/16M1083967
15. G. Eichfelder, An adaptive scalarization method in multiobjective optimization, *SIAM J. Optim.*, **19** (2009), 1694–1718. https://doi.org/10.1137/060672029
16. A. M. Geoffrion, Proper efficiency and the theory of vector maximization, *J. Math. Anal. Appl.*, **22** (1968), 618–630. https://doi.org/10.1016/0022-247X(68)90201-1
17. J. Fliege, B. F. Svaiter, Steepest descent methods for multicriteria optimization, *Math. Meth. Oper. Res.*, **51** (2000), 479–494. https://doi.org/10.1007/s001860000043
18. J. Fliege, L. M. Graña Drummond, B. F. Svaiter, Newton's method for multiobjective optimization, *SIAM J. Optim.*, **20** (2009), 602–626. https://doi.org/10.1137/08071692X
19. E. H. Fukuda, L. M. Graña Drummond, Inexact projected gradient method for vector optimization, *Comput. Optim. Appl.*, **54** (2013), 473–493. https://doi.org/10.1007/s10589-012-9501-z
20. H. Bonnel, A. N. Iusem, B. F. Svaiter, Proximal methods in vector optimization, *SIAM J. Optim.*, **15** (2005), 953–970. https://doi.org/10.1137/S1052623403429093

21. L. C. Ceng, B. S. Mordukhovich, J. C. Yao, Hybrid approximate proximal method with auxiliary variational inequality for vector optimization, *J. Optim. Theory Appl.*, **146** (2010), 267–303. https://doi.org/10.1007/s10957-010-9667-4
22. L. R. Lucambio Pérez, L. F. Prudente, Nonlinear conjugate gradient methods for vector optimization, *SIAM J. Optim.*, **28** (2018), 2690–2720. https://doi.org/10.1137/17M1126588
23. L. R. Lucambio Pérez, L. F. Prudente, A Wolfe line search algorithm for vector optimization, *ACM T. Math. Software*, **45** (2019), 37. https://doi.org/10.1145/3342104
24. M. L. N. Gonçalves, L. F. Prudente, On the extension of the Hager–Zhang conjugate gradient method for vector optimization, *Comput. Optim. Appl.*, **76** (2020), 889–916. https://doi.org/10.1007/s10589-019-00146-1
25. M. L. N. Gonçalves, F. S. Lima, L. F. Prudente, A study of Liu–Storey conjugate gradient methods for vector optimization, *Appl. Math. Comput.*, **425** (2022), 127099. https://doi.org/10.1016/j.amc.2022.127099
26. Q. R. He, C. R. Chen, S. J. Li, Spectral conjugate gradient methods for vector optimization problems, *Comput. Optim. Appl.*, **86** (2023), 457–489. https://doi.org/10.1007/s10589-023-00508-w
27. J. W. Chen, Y. S. Bai, X. L. Qin, X. Q. Ou, X. L. Qin, A PRP type conjugate gradient method without truncation for nonconvex vector optimization, *J. Optim. Theory Appl.*, **204** (2025), 13. https://doi.org/10.1007/s10957-024-02571-7
28. Z. Yang, L. Ma, Adaptive step size rules for stochastic optimization in large-scale learning, *Stat. Comput.*, **33** (2023), 45. https://doi.org/10.1007/s11222-023-10218-2
29. W. W. Wang, J. Gao, An efficient three-term conjugate gradient algorithm with restart strategy and image restoration, *Sci. Rep.*, **15** (2025), 35451. https://doi.org/10.1038/s41598-025-19329-4
30. H. Pan, J. W. Chen, K. P. Liu, Y. Lv, A modified Dai–Yuan conjugate gradient method for vector optimization problems, *Optimization*, **2025** (2025), 1–24. https://doi.org/10.1080/02331934.2025.2547719
31. J. W. Peng, J. W. Zhang, J. C. Yao, A novel hybrid conjugate gradient method for multiobjective optimization problems, *Optimization*, **75** (2026), 773–792. https://doi.org/10.1080/02331934.2024.2373903
32. M. A. T. Ansary, G. Panda, A modified quasi-Newton method for vector optimization problem, *Optimization*, **64** (2015), 2289–2306. https://doi.org/10.1080/02331934.2014.947500
33. S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE T. Evolut. Comput.*, **10** (2006), 477–506. https://doi.org/10.1109/TEVC.2005.861417
34. Y. C. Jin, M. Olhofer, B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how, In: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO'01)*, San Francisco: Morgan Kaufmann Publishers Inc., 2001, 1042–1049. Available from: `https://dl.acm.org/doi/10.5555/2955239.2955427`.
35. A. Lovison, Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis, *SIAM J. Optim.*, **21** (2011), 463–490. https://doi.org/10.1137/100784746

36. J. J. Moré, B. S. Garbow, K. E. Hillstrom, Testing unconstrained optimization software, *ACM T. Math. Software*, **7** (1981), 17–41. https://doi.org/10.1145/355934.355936

37. O. Schütze, M. Laumanns, C. A. Coello Coello, M. Dellnitz, E.-G. Talbi, Convergence of stochastic search algorithms to finite-size Pareto set approximations, *J. Glob. Optim.*, **41** (2008), 559–577. https://doi.org/10.1007/s10898-007-9265-7

38. J. Thomann, G. Eichfelder, Numerical results for the multiobjective trust region algorithm MHT, *Data Brief*, **25** (2019), 104103. https://doi.org/10.1016/j.dib.2019.104103

39. Ph. L. Toint, Test problems for partially separable optimization and results for the routine PSPMIN, Technical Report 83/4, Department of Mathematics, University of Namur, Belgium, 1983. Available from: `https://perso.unamur.be/~phtoint/pubs/TR83-04.pdf`.

AIMS Press