Nonlinear Convex Analysis and Optimization
Vol. 3, No. 1, 2024, pp. 1–23
https://ncao.design.blog

On the Class of Wei–Yao–Liu Conjugate Gradient Methods for Vector Optimization

Jamilu Yahaya [a,c,1], Poom Kumam [a,2*], Sani Salisu [a,d,3], Auta Jonathan Timothy [e,f,4]

[a] Center of Excellence in Theoretical and Computational Science (TaCS-CoE) and KMUTTFixed Point, Research Laboratory, Room SCL 802 Fixed Point Laboratory Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand.
[c] Department of Mathematics, Faculty of Physical Sciences, Ahmadu Bello University Zaria, Kaduna, Nigeria.
[d] Department of Mathematics, Faculty of Natural and Applied Sciences, Sule Lamido University Kafin Hausa, Jigawa, Nigeria.
[e] Department of Mathematics and Statistics, Federal University Wukari, Taraba State, Nigeria.
[f] Mathematics Institute, African University of Science and Technology, Abuja, Nigeria.
[1] jamiluyahayamaths@gmail.com; [2] poom.kum@kmutt.ac.th; [3] sani.salisu@slu.edu.ng;
[4] jauta@aust.edu.ng
[*] Corresponding Author

ABSTRACT

Vector optimization problems (VOPs) are crucial research areas with widespread applications. The scalarization approach is commonly used to solve VOPs by transforming vector-valued functions into single-objective optimization. Despite its elegance, this method has the drawback of subjective weight selections. Alternatively, we propose five conjugate gradient (CG) methods designed for VOPs, where the set of Pareto-optimal points are obtained without weight selections, the methods are Wei-Yao-Liu (WYL) and four of its variants. Three of these methods lack sufficient descent conditions (SDC) in this context. However, we establish their global convergence using Wolfe line search. The remaining two methods fulfill SDC with the Wolfe line search, and their global convergence is further verified using the Wolfe line search. Importantly, our approach does not rely on regular restart or convexity assumptions associated with objective functions. We conduct numerical experiments to showcase the effectiveness of our methods, comparing them with the nonnegative PRP method. Through these experiments, we demonstrate the practical implementations of our proposed techniques.

Article History
Received 27 February, 2024
Revised 29 April, 2024
Accepted 30 April, 2024

Keywords:
Conjugate gradient method; Pareto-optimality; Sufficient descent condition; Vector optimization;

MSC
90C29; 90C52; 90C30; 90C26; 49M37

Please cite this article as: Yahaya et al., On the Class of Wei–Yao–Liu Conjugate Gradient Methods for Vector Optimization, Nonlinear Convex Anal. & Optim., Vol. 3 No. 1, 1–23. https://doi.org/10.58715/ncao.2024.3.1

1. Introduction

Lately, there has been significant interest in the effective use of CG methods to solve vector optimization problems (VOPs), as outlined in [40]. These methods have garnered attention for their simplicity and minimal memory requirements, demonstrating notable effectiveness, [22, 21, 25, 54, 53, 55].

Before exploring VOPs, let us consider some well-known CG parameters related to the natural unconstrained optimization problem, which focuses on minimizing $\bar{f} : \mathbb{R}^n \longrightarrow \mathbb{R}$. The parameters include the β_k of Polak-Ribiére–Polyak (PRP) [42], Hestenes-Stiefel (HS) [26] and Liu-Storey (LS) [35]. Other well-known CG methods can be found in [2, 13, 12, 7, 24]. In most cases, the convergence of the CG method based on these parameters is achieved only if the search direction attains a decent property or sufficient descent condition.

Another important method which is a modification of PRP method is the Wei-Yao-Liu (WYL) CG method [52], several other methods were developed due to the introduction of WYL CG method [56, 51, 29, 47].

In the following, we consider an unconstrained vector optimization problem of the form

$$\text{Minimize}_Q \ F(z), \tag{1.1}$$

where $F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is in C^1 (continuously differentiable function), $z \in \mathbb{R}^n$, and $Q \subset \mathbb{R}^m$ is closed, convex and pointed cone with nonempty interior. The partial order defined in $\mathbb{R}^m$, $\preccurlyeq_Q$, generated by Q is $a \preccurlyeq_Q b \iff b - a \in Q$, and $\prec_Q$, generated by $int(Q)$ is $a \prec_Q b \iff b - a \in int(Q)$. If $Q = \mathbb{R}^m_+$, then problem (1.1) is considered to be multiobjective optimization problem, and if $Q = \mathbb{R}_+$ then it reduces to single-objective optimization.

Several applications in industry and finance are considered instances of VOP, where multiple objective functions are optimized concurrently. Consequently, it becomes imperative to determine a set of optimal points for VOP [9, 17, 16, 23, 27, 31, 34, 48]. Due to the absence of a total order in $\mathbb{R}^m$, where $m \geq 2$, the solution of VOP entails a collection of non-dominated points, commonly known as Pareto optimal or efficient points. The difficulty lies in pinpointing the solutions that achieve the most favorable balance.

One approach to addressing VOPs involves scalarization techniques, which transform single-objective optimization problems into parameterized forms to generate Pareto-optimal points. The selection of these parameters is done by a decision-maker as they are not predefined. However, this decision-making process can present significant challenges or even be infeasible for certain problems [37, 33]. Consequently, to mitigate these limitations, some descent-based algorithms have been proposed as alternative solution methods for VOPs, as highlighted in the works of [11, 4]. Subsequently, numerous other studies have pursued similar avenues, exploring comparable approaches. For further details, refer to the survey on descent methods in multi-objective optimization (MOO) presented in [18], along with the references [1, 3, 14, 44].

In [40], conjugate parameters from [13, 42, 26, 6, 7] are explored for VOPs, with numerical implementations and analysis conducted. Notably, among these methods, the nonnegative PRP and HS demonstrated superior performance across various test problems even though they could not achieve SDC. Conversely, CD and DY methods exhibited greater efficiency than FR.

Goncalves and Prudente [22] later extended the Hager-Zhang (HZ) CG method for VOPs, although without guaranteeing descent conditions in the search direction, even with an exact line search. To tackle this, they proposed a self-adjusting HZ method utilizing a sufficiently accurate Wolfe line search, ensuring the descent property. Further research in this realm includes the LS CG method and its variants [21], the first hybrid CG methods for VOPs [54], modified CG methods [53, 55], the extension of spectral CG method [25] and alternative extension of the HZ CG method [28]. Other CG methods studied for MOO can found in [5].

To study the possible extension of the WYL CG method to vector setting, we propose five CG methods designed for solving VOPs. The first three methods are the nonnegative Wei-Yao-Liu (WYL) and its HS and LS types. Although these three methods lose their descent property in the vector setting, we establish their global convergence by employing a sufficiently accurate Wolfe line search. On the other hand, we modified the WYL of the HS and LS types and established two new methods that achieve SDC with Wolfe line search; global convergence is also established using Wolfe line search. We provide numerical implementations to demonstrate the efficiency and robustness of the proposed methods by comparing them with the nonnegative PRP method.

The paper is structured as follows: Section 2 introduces fundamental concepts and preliminary results related to VOPs. Section 3 examines the convergence properties of the proposed methods. Section 4 presents and discusses the numerical results. Finally, in Section 5, we have the concluding remarks.

2. Preliminaries

In this section, we present some basic notions and results of VOP used in this paper. For some notable preliminaries, see the references [11, 38, 40].

The aim in vector optimization is to minimize a finite set of objective functions simultaneously. Rarely does a single point minimize all objective functions at once. In this setting, an alternative notion of optimality is needed. The concept of *Pareto-optimality* and *weak Pareto-optimality* are utilized instead.

Definition 2.1. [18] A point $\bar{z} \in \mathbb{R}^n$ is Pareto-optimal or efficient if and only if there does not exists $z \in \mathbb{R}^n$ such that $F(z) \preccurlyeq_Q F(\bar{z})$ and $F(z) \neq F(\bar{z})$.
A point $\bar{z} \in \mathbb{R}^n$ is weak Pareto-optimal or weak efficient if and only if there does not exists $z \in \mathbb{R}^n$ such that $F(z) \prec_Q F(\bar{z})$.

Note that when $\bar{z} \in \mathbb{R}^n$ represents a Pareto-optimal point, it is also qualifies as a weak Pareto-optimal point. However, the reverse statement is often not true, [18].

Now, let us look at some properties related to Q: the positive polar cone of Q is given as

$$Q^* := \{p \in \mathbb{R}^m \mid \langle p, z\rangle \geq 0, \ \forall\ z \in Q\}.$$

Note that since Q is closed and convex. Then, $Q = Q^{**}$. If $C \subseteq Q^* \backslash \{0\}$ is compact, then Q^* is defined as the conic hull of a convex hull of C :

$$Q^* = cone(conv(C)). \tag{2.1}$$

Again,

$$-Q = \{z \in \mathbb{R}^m \mid \langle z, p\rangle \leq 0, \ \forall\ p \in Q^*\}, \quad -int(Q) = \{z \in \mathbb{R}^m \mid \langle z, p\rangle < 0, \ \forall\ p \in Q^* \backslash \{0\}\}.$$

For a given point z, the term Image$(JF(z))$ represents the image on $\mathbb{R}^m$ generated by $JF(z)$. A necessary requirement for Q-optimality of $\bar{z} \in \mathbb{R}^n$ is given as

$$-int(Q) \cap Im(JF(\bar{z})) = \emptyset. \tag{2.2}$$

If the condition (2.2) is satisfied, we called $\bar{z} \in \mathbb{R}^n$ as *stationary or Q-critical point*. On the contrary, if $\bar{z} \in \mathbb{R}^n$ does not meet the criteria for stationary or Q-critical point, then there exists $h \in \mathbb{R}^n$ such that $JF(\bar{z})h \in -int(Q)$. This signifies that h is a Q-descent direction for F at the point $\bar{z}$. In other words, we have $s > 0$ for which $F(\bar{z} + \bar{r}h) \prec_Q F(\bar{z})$, for all $0 < \bar{r} < s$, see e.g., [38] for a full discussion on this.

Now, for a given Q (closed, convex and pointed cone with nonempty interior), the set

$$C = \{p \in Q^* \mid ||p|| = 1\}, \tag{2.3}$$

satisfies (2.1). Subsequently, we consider C to be as defined in equation (2.3).

Let us define $\theta : \mathbb{R}^m \to \mathbb{R}$ as

$$\theta(z) := \sup\{\langle z, p\rangle \mid p \in C\}. \tag{2.4}$$

The map θ is well-defined by the compactness of C. Again, define $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$\phi(z, d) := \theta(JF(z)d) = \sup\{\langle JF(z)d, \ p\rangle \mid p \in C\}. \tag{2.5}$$

For a given point z, we represent Jacobian of F by $JF(z)$.

Again, let us define the steepest descent direction and the optimal value: $u : \mathbb{R}^n \to \mathbb{R}^n$ and $v : \mathbb{R}^n \to \mathbb{R}$, respectively by

$$u(z) := \operatorname{argmin}\left\{\phi(z, d) + \frac{\|d\|^2}{2} \mid d \in \mathbb{R}^n\right\} \tag{2.6}$$

and

$$v(z) := \phi(z, u(z)) + \frac{\|u(z)\|^2}{2}. \tag{2.7}$$

Given that the real-valued function $\phi(z, \cdot)$ is convex and $d \mapsto \frac{\|d\|^2}{2}$ is strictly convex, then $u(z)$ exists and is unique. The function u allows us to develop the concept of the steepest descent direction in the vector minimization setting. It is worth noting that in scalar optimization, we have $\phi(z, d) = \langle \nabla F(z), d\rangle$, $u(z) = -\nabla F(z)$ and $v(z) = -\frac{\|\nabla F(z)\|^2}{2}$.

Now, we can describe a CG method as

$$z_{k+1} = z_k + \alpha_k d_k, \qquad k \geq 1, \tag{2.8}$$

where $\alpha_k > 0$ is the step size or step length which is obtainable through a line search technique, and d_k is the search direction defined by

$$d_k := \begin{cases} u(z_k), & k = 1, \\ u(z_k) + \beta_k d_{k-1}, & k \geq 2. \end{cases} \tag{2.9}$$

The algorithmic parameter β_k comes in numerous types; below are some of the possible options:

$$\beta_k^{FR} := \frac{\phi(z_k, u(z_k))}{\phi(z_{k-1}, u(z_{k-1}))}, \qquad \beta_k^{CD} := \frac{\phi(z_k, u(z_k))}{\phi(z_{k-1}, d_{k-1})}, \tag{2.10}$$

$$\beta_k^{DY} := \frac{-\phi(z_k, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}, \quad \beta_k^{PRP} := \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, u(z_{k-1}))}, \quad (2.11)$$

$$\beta_k^{HS} := \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}, \quad \beta_k^{LS} := \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})}, \quad (2.12)$$

are the Fletcher-Reeves (FR), Conjugate Descent (CD), Dai-Yuan (DY), Polak-Ribiére-Polyak (PRP), Hestenes-Stiefel (HS), and Liu-Storey (LS), respectively.

In the convergence analysis of CG methods, it is required that the search direction d to be Q-descent direction for F at z, that is

$$\phi(z, d) < 0. \quad (2.13)$$

A point z is Q-critical point for F if

$$\phi(z, d) \geq 0, \quad (2.14)$$

for all $d \in \mathbb{R}^n$. A direction d is said to satisfies *sufficient descent condition* (SDC) at z if

$$\phi(z, d) \leq c\phi(z, u(z)), \quad (2.15)$$

for some $c > 0$.

Lemma 2.2. *[11]. Suppose $F : \mathbb{R}^n \to \mathbb{R}^m$ is in C^1. Then, the statements below hold:*

(a) $\phi(z, z' + \alpha d) \leq \phi(z, z') + \alpha\phi(z, d)$, *for* $z, z', d \in \mathbb{R}^n$ *and* $\alpha \geq 0$;

(b) The mapping $(z, d) \longmapsto \phi(z, d)$ *is continuous;*

(c) $|\phi(z, d) - \phi(z', d)| \leq \|JF(z) - JF(z')\|\|d\|$, *for* $z, z', d \in \mathbb{R}^n$;

(d) Let $\|JF(z) - JF(z')\| \leq L\|z - z'\|$, *then* $|\phi(z, d) - \phi(z', d)| \leq L\|d\|\|z - z'\|$.

Consider the following convex quadratic problem

$$\begin{aligned} &\text{Minimize } \alpha + \frac{1}{2}\|u\|^2, \\ &\text{subject to } [JF(z)u]_i \leq \alpha, \quad i = 1, 2, \cdots, m, \end{aligned} \quad (2.16)$$

with linear inequality constraints, see for instance, [15]. We say that the step size, $\alpha > 0$ can be obtained through an exact line search at a point x along the direction d if

$$\phi(z + \alpha d, d) = 0. \quad (2.17)$$

We now give the vector Wolfe conditions that was introduced by Lucambio Pérez and Prudente [39].

Definition 2.3. [40] Let $d \in \mathbb{R}^n$ be a Q-descent direction and $e \in Q$, then we have

$$0 < \langle p, \ e \rangle \leq 1, \quad (2.18)$$

for all $p \in C$. Now, $\alpha > 0$ satisfies the *standard Wolfe condition* (WWC) if

$$F(z + \alpha d) \preccurlyeq_Q F(z) + \rho\alpha\phi(z, d)e$$

$$\phi(z+\alpha d, d) \geq \sigma \phi(z, d), \tag{2.19}$$

where $0 < \rho < \sigma < 1$. Furthermore, $\alpha > 0$ satisfies the *strong Wolfe condition* (SWC) if

$$F(z+\alpha d) \preccurlyeq_Q F(z) + \rho \alpha \phi(z, d) e$$
$$|\phi(z+\alpha d, d)| \leq \sigma |\phi(z, d)|. \tag{2.20}$$

It is interesting to know that the vector $e \in Q$ given in (2.18), always exists. Specifically, for multiobjective optimization, we define e as $[1, \cdots, 1]^T \in \mathbb{R}^m$, Q as $\mathbb{R}^m_+$, and C as $\{e_1, e_2, \cdots, e_m\} \subset \mathbb{R}^m$.

Let us now conclude this section with the following important results.

Lemma 2.4. *[11]. (a) let z be a Q-critical for F, then $u(z) = 0$ and $v(z) = 0$. (b) suppose z is not Q-critical for F, then $u(z) \neq 0$, $v(z) < 0$, $\phi(z, u(z)) < -\frac{\|u(z)\|^2}{2} < 0$ and $u(z)$ Q-descent direction for F at z. (c) The u and v are continuous maps.*

3. Algorithm and Its Convergence Analysis

This section presents the methods and the general prototype of the algorithm, along with the analysis that leads to the SDC property and the global convergence of these methods.

Assumption 3.1. Suppose that the cone Q is finitely generated and there exists an open set Δ for which the $\mathcal{L} := \{z \mid F(z) \preccurlyeq_Q F(z_1)\} \subset \Delta$, where $z_1 \in \mathbb{R}^n$ and there exists $L > 0$ such that $\|JF(z) - JF(z')\| \leq L\|z - z'\|$ for all $z, z' \in \Delta$.

Assumption 3.2. The level set $\mathcal{L} := \{z \mid F(z) \preccurlyeq_Q F(z_1)\}$ is bounded.

Note that, by Assumption 3.2 we have that for any $\{z_k\}$ in $\mathcal{L}$, there exists $\bar{M} > 0$ s.t

$$\|z_k\| \leq \bar{M}, \tag{3.1}$$

for all k. Therefore, we have from Lemma 2.2(d) that there exists $\gamma > 0$ s.t

$$\|JF(z_k)\| \leq \gamma, \tag{3.2}$$

for all k. Also, by the boundedness of $\{\phi(z_k, u(z_k))\}$ and Lemma 2.4(b), there exists $\delta > 0$ s.t

$$\|u(z_k)\| \leq \delta, \tag{3.3}$$

for all k. From (2.5), Lemma 2.4 (b), (3.2) and (3.3), we have

$$0 < -\phi(z_k, u(z_k)) \leq -\langle JF(z_k)u(z_k), q\rangle \leq \|JF(z_k)\|\|u(z_k)\| \leq \delta\gamma, \tag{3.4}$$

with $\|q\| = 1$.

We state the general prototype of the considered CG algorithm for VOPs.

Algorithm 1:

Step 0: Let $z_1 \in \mathbb{R}^n$ be given and initialize $k \longleftarrow 1$.
Step 1: Compute $u(z_k)$ and $v(z_k)$ as in (2.6) and (2.7), respectively.
Step 2: Compute $\alpha_k > 0$ using condition (2.20).
Step 3: If $v(z_k) = 0$, then stop. Otherwise, compute

$$\begin{aligned} d_k &= u(z_k) \quad \textit{for} \qquad\quad k = 1, \\ d_k &= u(z_k) + \beta_k d_{k-1} \quad \textit{for} \quad k \geq 2, \end{aligned} \tag{3.5}$$

where β_k is the considered conjugate parameter.
Step 4: Set $z_{k+1} = z_k + \alpha_k d_k$, for $k \longleftarrow k+1$ and go to **Step 1.**

We define the Wei-Yao-Liu (WYL) β_k parameter as follows:

$$\beta_k^{WYL} := \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, u(z_{k-1}))}. \tag{3.6}$$

In light of this, we propose the following variants of the WYL and their modified versions:

$$\beta_k^{WHS} := \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}. \tag{3.7}$$

$$\beta_k^{WLS} := \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})}. \tag{3.8}$$

$$\beta_k^{WHS^*} := \frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}. \tag{3.9}$$

$$\beta_k^{WLS^*} := \frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})}. \tag{3.10}$$

Remark 3.3. It is important to note that all the CG parameters in (3.6)–(3.8) are well-defined based on Lemma 2.4(b) and the conditions that: (i) d_k is a Q-descent direction of F at z_k, and (ii) α_k satisfies condition (2.20). Moreover, in our subsequent analysis, it is important to note that we only consider the values $\max\{\beta_k, 0\}$ for each parameter in (3.6)–(3.10), provided that $\phi(z_{k-1}, u(z_k)) > 0$. If $\phi(z_{k-1}, u(z_k)) \leq 0$, we set $\max\{\beta_k, 0\} := 0$. Thus, based on the considered formulation, the β_k in (3.6)–(3.8) are nonnegative, making $\max\{\beta_k, 0\} = \beta_k$ for all $k \geq 2$.

The well-known property $(*)$, originally introduced by Gilbert and Nocedal [19] to analyze the global convergence of PRP and HS in scalar, its vector extension was subsequently provided by Lucambio Pérez and Prudente [40]. The property is stated as follows:

Property $(*)$ [40] Consider Algorithm 1 and suppose that

$$0 < \bar{\delta} \leq \|u(z_k)\|, \tag{3.11}$$

for all $k \geq 2$. Using the assumption above, we get a property $(*)$ if there exist some constants $q > 1$ and $\lambda > 0$ for all k:

$$|\beta_k| \leq q,$$

and

$$\|s_{k-1}\| \leq \lambda \Longrightarrow |\beta_k| \leq \frac{1}{2q},$$

where $s_{k-1} = z_k - z_{k-1}$.

The following lemma follows from Theorem 5.10 in [40].

Lemma 3.4. *Consider Algorithm 1 and let Assumptions 3.1 and 3.2 hold, for all k, where:*

(a) β_k is nonnegative;
(b) d_k is a Q-descent direction of F at z_k;
(c) α_k satisfies condition (2.20);
(d) property $(*)$ holds.
Then,

$$\liminf_{k\to\infty} \|u(z_k)\| = 0.$$

Theorem 3.5. *Consider Algorithm 1 such that the sequence $\{z_k\}$ is generated with $\beta_k = \beta_k^{WYL}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. Suppose Assumptions 3.1 and 3.2 hold. If d_k is Q-descent direction of F at z_k and α_k satisfies condition (2.20). Then,*

$$\liminf_{k\to\infty} \|u(z_k)\| = 0. \tag{3.12}$$

Proof. It is observed from Lemma 3.4 that it is enough to show that WYL satisfies property $(*)$. To demonstrate this, we follow the approach outlined in [21], wherein we establish the existence of a nonnegative constant ϵ such that

$$|\beta_k| \leq \epsilon\|s_{k-1}\|, \quad \forall \quad k \geq 2. \tag{3.13}$$

Now, assume that (3.11) holds. Then, by (3.3) and (3.11), we have

$$0 < \bar{\delta} \leq \|u(z_k)\| \leq \delta, \quad \forall \quad k \geq 2. \tag{3.14}$$

Additionally, by (3.2) and Lemma 2.4 (b), we have

$$\frac{\bar{\delta}^2}{2} \leq -\phi(z_k, u(z_k)) \leq \delta\gamma. \tag{3.15}$$

We also see from (3.2) and (2.5) that there exists $\bar{p} \in C$ such that

$$|\phi(z_{k-1}, u(z_k))| = |\langle JF(z_{k-1})u(z_k), \bar{p}\rangle| \leq ||JF(z_{k-1})||||u(z_k)|| \leq \delta\gamma, \tag{3.16}$$

then by Assumption 3.1, Lemma 2.2(d), and (3.14), for all $k \geq 2$, we get

$$\left| -\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k)) \right| \leq L\|z_k - z_{k-1}\|\|u(z_k)\| \leq L\delta\|s_{k-1}\|, \tag{3.17}$$

where $\|s_{k-1}\| = \|z_k - z_{k-1}\|$.

Now, consider $\phi(z_{k-1}, u(z_k)) > 0$, then by Lemma 2.4 (b), we have

$$\begin{aligned}
\mid \beta_k \mid = \beta_k^{WYL} &= \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, u(z_{k-1}))} \\
&= \frac{-\|u(z_{k-1})\|\phi(z_k, u(z_k)) + \|u(z_k)\|\phi(z_{k-1}, u(z_k))}{-\|u(z_{k-1})\|\phi(z_{k-1}, u(z_{k-1}))} \\
&\le \frac{\delta}{\bar{\bar{\delta}}}\left(\frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, u(z_{k-1}))}\right) \\
&\le \frac{\delta}{\bar{\bar{\delta}}}\frac{\left|-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))\right|}{-\phi(z_{k-1}, u(z_{k-1}))}.
\end{aligned}$$

This, combined with (3.15) and (3.17), imply that

$$|\beta_k| \le \frac{2L\delta^2\|s_{k-1}\|}{\bar{\delta}^3},$$

where $\epsilon = \frac{2L\delta^2}{\bar{\delta}^3}$. Since $\beta_k = 0$ for the case when $\phi(z_{k-1}, u(z_k)) \le 0$. This implies that

$$|\beta_k| \le \frac{2L\delta^2\|s_{k-1}\|}{\bar{\delta}^3}, \quad \forall k \ge 2,$$

which completes the proof. ∎

Theorem 3.6. *Let Assumptions 3.1 and 3.2 hold. Consider Algorithm 1 such that the sequence $\{z_k\}$ is generated with $\beta_k = \beta_k^{WHS}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. If d_k satisfies the SDC and α_k satisfies condition (2.20). Then,*

$$\liminf_{k\to\infty} \|u(z_k)\| = 0. \tag{3.18}$$

Proof. The proof follows the same pattern as that of Theorem 3.5. Firstly, we observe that by (2.19) and (2.15), we have

$$\begin{aligned}
\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1}) &\ge -(1-\sigma)\phi(z_{k-1}, d_{k-1}) \\
&\ge -c(1-\sigma)\phi(z_{k-1}, u(z_{k-1})) > 0.
\end{aligned} \tag{3.19}$$

Now, consider $\phi(z_{k-1}, u(z_k)) > 0$, then we have

$$\begin{aligned}
\mid \beta_k \mid = \beta_k^{WHS} &= \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \\
&= \frac{1}{\|u(z_{k-1})\|}\left(\frac{-\|u(z_{k-1})\|\phi(z_k, u(z_k)) + \|u(z_k)\|\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}\right) \\
&\le \frac{\delta}{\bar{\bar{\delta}}}\left(\frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}\right) \\
&\le \frac{\delta}{\bar{\bar{\delta}}}\left|\frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}\right|.
\end{aligned}$$

Now Using (3.19), we get

$$|\beta_k| \leq \frac{\delta}{\bar{\bar{\delta}}} \frac{\Big|-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))\Big|}{-c(1-\sigma)\phi(z_{k-1}, u(z_{k-1}))}.$$

By (3.15) and (3.17), we have

$$|\beta_k| \leq \frac{\delta}{\bar{\bar{\delta}}} \frac{\Big|-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))\Big|}{-c(1-\sigma)\phi(z_{k-1}, u(z_{k-1}))} \leq \frac{2L\delta^2\|s_{k-1}\|}{c(1-\sigma)\bar{\delta}^3}.$$

Thus,

$$|\beta_k| \leq \frac{2L\delta^2\|s_{k-1}\|}{c(1-\sigma)\bar{\delta}^3}, \tag{3.20}$$

where $\epsilon = \frac{2L\delta^2}{c(1-\sigma)\bar{\delta}^3}$. Observe that, since $\beta_k = 0$ if $\phi(z_{k-1}, u(z_k)) \leq 0$, which guarantees that (3.20) holds for all $k \geq 2$. ∎

Theorem 3.7. *Let Assumptions 3.1 and 3.2 hold. Consider Algorithm 1 such that the sequence $\{z_k\}$ is generated with $\beta_k = \beta_k^{WLS}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. If d_k satisfies the SDC and α_k satisfies condition (2.20). Then,*

$$\liminf_{k\to\infty} \|u(z_k)\| = 0. \tag{3.21}$$

Proof. The proof follows the same pattern as that of Theorem 3.5. Firstly, by (2.15) and Lemma 2.4 (b), we have

$$-\phi(z_{k-1}, d_{k-1}) \geq -c\phi(z_{k-1}, u(z_{k-1})) > 0. \tag{3.22}$$

Now, consider $\phi(z_{k-1}, u(z_k)) > 0$, then we have

$$\begin{aligned} |\beta_k| = \beta_k^{WLS} &= \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})} \\ &\leq \frac{\delta}{\bar{\bar{\delta}}}\left(\frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})}\right) \\ &\leq \frac{\delta}{\bar{\bar{\delta}}} \frac{\Big|-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))\Big|}{-c\phi(z_{k-1}, u(z_{k-1}))}. \end{aligned}$$

By (3.15) and (3.17), we have

$$|\beta_k| \leq \frac{\delta}{\bar{\bar{\delta}}} \frac{\Big|-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))\Big|}{-c\phi(z_{k-1}, u(z_{k-1}))} \leq \frac{2L\delta^2\|s_{k-1}\|}{c\bar{\delta}^3},$$

which implies that

$$|\beta_k| \leq \frac{2L\delta^2\|s_{k-1}\|}{c\bar{\delta}^3}, \tag{3.23}$$

where $\epsilon = \frac{2L\delta^2}{c\bar{\delta}^3}$. Observe that, since $\beta_k = 0$ if $\phi(z_{k-1}, u(z_k)) \leq 0$, which guarantees that (3.23) holds for all $k \geq 2$. ∎

Next, we consider a modified version of WHS given as (3.9) and investigate its descent as well as convergence properties.

Lemma 3.8. *Consider Algorithm 1 such that the sequence* $\{z_k\}$ *is generated with* $\beta_k = \max\{\beta_k^{WHS^*}, 0\}$ *if* $\phi(z_{k-1}, u(z_k)) > 0$ *or* $\beta_k = 0$ *otherwise. Suppose* α_k *satisfy (2.20). Then* d_k *defined by (3.5) satisfies the SDC with* $c = \frac{1}{1+\sigma}$.

Proof. Since $\beta_k \geq 0$, it follows from Lemma 2.2(a) and (3.5) that

$$\phi(z_k, d_k) \leq \phi(z_k, u(z_k)) + \beta_k \phi(z_k, d_{k-1}).$$

For the case when $\beta_k = 0$ or $\phi(z_k, d_{k-1}) \leq 0$, we get

$$\phi(z_k, d_k) \leq \phi(z_k, u(z_k)) \leq \frac{1}{1+\sigma}\phi(z_k, u(z_k)). \tag{3.24}$$

When $\beta_k = \beta_k^{WHS^*}$ and $\phi(z_k, d_{k-1}) > 0$, then $\phi(z_{k-1}, u(z_k)) > 0$ and consequently, we get

$$\begin{aligned}
\phi(z_k, d_k) &\leq \phi(z_k, u(z_k)) + \left(\frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}\right)\phi(z_k, d_{k-1}) \\
&\leq \phi(z_k, u(z_k)) + \frac{-\phi(z_k, u(z_k))\phi(z_k, d_{k-1})}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \\
&\leq \left(1 - \frac{\phi(z_k, d_{k-1})}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}\right)\phi(z_k, u(z_k)) \\
&\leq \left(\frac{\phi(z_{k-1}, d_{k-1})}{\phi(z_{k-1}, d_{k-1}) - \phi(z_k, d_{k-1})}\right)\phi(z_k, u(z_k)) \\
&\leq (\frac{1}{1-q_k})\phi(z_k, u(z_k)),
\end{aligned}$$

where $q_k = \frac{\phi(z_k, d_{k-1})}{\phi(z_{k-1}, d_{k-1})}$. Observe that by (2.20) we have $q_k \in [-\sigma, \sigma]$. Again, by Lemma 2.4(b) we have $\phi(z_k, u(z_k)) < 0$ for all k, then

$$\phi(z_k, d_k) \leq (\frac{1}{1-q_k})\phi(z_k, u(z_k)) \leq (\frac{1}{1+\sigma})\phi(z_k, u(z_k)). \tag{3.25}$$

This completes the proof. ∎

Theorem 3.9. *Let Assumptions 3.1 and 3.2 hold. Consider Algorithm 1 such that the sequence* $\{z_k\}$ *is generated with* $\beta_k = \max\{\beta_k^{WHS^*}, 0\}$ *if* $\phi(z_{k-1}, u(z_k)) > 0$ *or* $\beta_k = 0$ *otherwise. If* α_k *satisfies condition (2.20). Then,*

$$\liminf_{k\to\infty} \|u(z_k)\| = 0. \tag{3.26}$$

Proof. Just like in the case of Theorem 3.5 and considering the result of Lemma 3.8, we only need to establish a constant ϵ such that

$$|\beta_k| \leq \epsilon\|s_{k-1}\|, \quad \forall \ \ k \geq 2. \tag{3.27}$$

For the case when $\beta_k = 0$, the desired inequality (3.27) is satisfied for any ϵ. So, we only need to consider the case when $\beta_k = \beta_k^{WHS^*} > 0$. In this case, $\phi(z_{k-1}, u(z_k)) > 0$ and following (3.19), we have

$$\begin{aligned} \mid \beta_k \mid = \beta_k^{WHS^*} &= \frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \\ &\leq \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \\ &\leq \frac{2L\delta^2\|s_{k-1}\|}{c(1-\sigma)\bar{\delta}^3}, \end{aligned}$$

In a similar way to the proof of Theorem 3.6, we conclude that

$$\mid \beta_k \mid = \beta_k^{WHS^*} \leq \frac{2L\delta^2\|s_{k-1}\|}{c(1-\sigma)\bar{\delta}^3},$$

where $\epsilon = \frac{2L\delta^2}{c(1-\sigma)\bar{\delta}^3}$. ∎

Next, we consider the modified version of WLS given as (3.10) and investigate its descent property.

Lemma 3.10. *Consider Algorithm 1 such that the sequence $\{z_k\}$ is generated with $\beta_k = \max\{\beta_k^{WHS^*}, 0\}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. Suppose that α_k satisfies (2.20). Then d_k defined by (3.5) satisfies the SDC with $c = 1 - \sigma$.*

Proof. Following Lemma 2.2 (a), (3.5) and the fact that $\beta_k^{WLS^*} \geq 0$, we have

$$\phi(z_k, d_k) \leq \phi(z_k, u(z_k)) + \beta_k^{WLS^*}\phi(z_k, d_{k-1}). \tag{3.28}$$

If $\beta_k = 0$ or $\phi(z_k, d_{k-1}) \leq 0$, we get

$$\phi(z_k, d_k) \leq \phi(z_k, u(z_k)) \leq (1-\sigma)\phi(z_k, u(z_k)).$$

When $\beta_k = \beta_k^{WHS^*}$ and $\phi(z_k, d_{k-1}) > 0$, then $\phi(z_{k-1}, u(z_k)) > 0$ and consequently, we get

$$\begin{aligned} \phi(z_k, d_k) &\leq \phi(z_k, u(z_k)) + \left(\frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})}\right)\phi(z_k, d_{k-1}) \\ &\leq \phi(z_k, u(z_k)) + \frac{\phi(z_k, u(z_k))\phi(z_k, d_{k-1})}{\phi(z_{k-1}, d_{k-1})}. \end{aligned}$$

Applying (2.20), we have

$$\phi(z_k, d_k) \leq (1-\sigma)\phi(z_k, u(z_k)).$$

This complete the proof. ∎

Theorem 3.11. *Let Assumptions 3.1 and 3.2 hold. Consider Algorithm 1 such that the sequence $\{z_k\}$ is generated with $\beta_k = \max\{\beta_k^{WLS^*}, 0\}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. If α_k satisfies condition (2.20). Then,*

$$\liminf_{k\to\infty} \|u(z_k)\| = 0. \tag{3.29}$$

Proof. Just like in the case of Theorem 3.5 and considering the result of Lemma 3.10, we only need to establish a constant ϵ such that

$$|\beta_k| \leq \epsilon \|s_{k-1}\|, \quad \forall \quad k \geq 2. \tag{3.30}$$

For the case when $\beta_k = 0$, the desired inequality (3.30) is satisfied for any ϵ. So, we only need to consider the case when $\beta_k = \beta_k^{WLS^*} > 0$. In this case, $\phi(z_{k-1}, u(z_k)) > 0$ and following (3.19), we have ...further justified by Lemma 2.4 (b), thus, we have

$$\begin{aligned} |\ \beta_k\ | = \beta_k^{WLS^*} &= \frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}|\phi(z_{k-1}, u(z_k))|}{-\phi(z_{k-1}, d_{k-1})} \\ &\leq \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_k)\|}{\|u(z_{k-1})\|}\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})} \\ &\leq \frac{2L\delta^2\|s_{k-1}\|}{c\bar{\delta}^3}, \end{aligned}$$

In a similar way to the proof of Theorem 3.7, we conclude that

$$|\ \beta_k\ | \leq \frac{2L\delta^2\|s_{k-1}\|}{c\bar{\delta}^3},$$

where $\epsilon = \frac{2L\delta^2}{c\bar{\delta}^3}$. This completes the proof. ∎

4. Numerical Experiments

In this section, we evaluate the performance of the proposed methods by examining them. We aim to measure their efficiency and robustness in addressing benchmark test problems involving some convex and nonconvex multi-objective optimization (MOO) sourced from various multiobjective optimization research articles in the literature. The algorithms were coded in MATLAB R2023b using a PC with the following specifications: Intel Core i5-1135G7 CPU running at 2.4GHz, and 16 GB of RAM. Subsequently, in the context of multiobjective optimization, we define $e = [1, \cdots, 1]^T \in \mathbb{R}^m$, $Q = \mathbb{R}^m_+$, and $C = \{e_1, e_2, \cdots, e_m\} \subset \mathbb{R}^m$.

Below, we present a summary of the methods under consideration. This encompasses both our proposed methods and those employed for comparison purposes:

- WYL: is the parameter β_k^{WYL};
- WHS: is the parameter β_k^{WHS};
- WLS: is the parameter β_k^{WLS};
- WHS*: is the parameter $\max\{\beta_k^{WHS^*}, 0\}$;
- *WLS**: is the parameter $\max\{\beta_k^{WLS^*}, 0\}$;
- PRP: a nonnegative PRP CG method with $\beta_k := \max\{\beta_k^{PRP}, 0\}$ in [40].

See Remark 3.3 for details on these parameters. Usually, a nonnegative CG parameter is denoted with a plus sign at the end, for instance, PRP+. However, throughout this section, we refer to a nonnegative PRP simply as PRP, likewise for the rest of the methods.

An essential part of these methods is the computation of the steepest descent direction and the optimal value, denoted as $u(z)$ and $v(z)$ defined in (2.6) and (2.7) respectively. To achieve this, we made use of the built-in function in MATLAB called *quadprog* to solve problem (2.16). In addition, the selection of the step size was performed using a line strategy that satisfies (2.20). Below are the initial parameters utilized in the implementation of our proposed methods for these line searches: $\rho = 10^{-4}$, $c = \frac{2}{5}$, $\sigma = 10^{-1}$, $\mu_1 = 10^{-1}$, $\mu_2 = \frac{1}{2}$.

Furthermore, Lemma 2.4 establishes that $z \in \mathbb{R}^n$ represents a Q-critical point of F only if $v(z) = 0$. Based on these findings, the experimentation process involved executing all the implemented methods until the point of convergence, defined as $v(z) \geq -5 \times eps^{\frac{1}{2}}$. Here, *eps* corresponds to the machine precision which is approximately 2.22×10^{-16}. Alternatively, the process terminates if the maximum number of iterations, $max.It = 5000$, is exceeded.

For computational purposes, we use a scaled processing technique for VOP with $Q = \mathbb{R}^m_+$:

$$\min_{z \in \mathbb{R}^n} (\lambda_1 F_1(z), \cdots, \lambda_m F_m(z)), \tag{4.1}$$

where $\lambda_i = \frac{1}{\max(1, \|\nabla F_i(z_1)\|_\infty)}$, $i = 1, 2, 3, \cdots, m$, and $z_1 \in \mathbb{R}^n$. This idea was derived from [20, 21]. It is observed that VOP with $Q = \mathbb{R}^m_+$ is always equivalent to (4.1), this is because they have the same Q-Pareto optimality.

Let us now discuss on the provided tables. Table 1 presents essential information regarding the selected test problems. In the first column, we have the names of the problems, such as "Lov5" aligning to the fifth problem introduced by A. Lovison in [36], and "SLCDT2" corresponding to the second problem given by Schütze, Lara, Coello, Dellnitz, and Talbi in [45]. The second column denotes the sources of the problems and the third and fourth columns labeled as "n" and "m" respectively, indicate the variables under consideration and the objective functions of the problems. To generate the starting points, a box constraint was utilized, defined as $\{z \in \mathbb{R}^n \mid \overline{l} \leq z \leq \bar{u}\}$, with the lower and upper bounds denoted in the fifth and sixth columns, respectively.

Tables 2 and 3 present the results of the proposed methods in comparison with the PRP CG method and are organized as follows: 'It', 'Fe', 'Ge', and 'time'. In this case, 'It', 'Fe', 'Ge', and 'time' denote the median numbers of iterations, function evaluations, gradient evaluations, and CPU time, respectively.

We focus on approximating the Pareto frontiers of the provided problems. To achieve this, we employed a methodology in which each implemented method was executed 100 times for each problem. All methods successfully solved the problems 100%, except for AP3 which 97% of it was solved by WHS while WYL, WLS, PRP, WHS* and WLS* solved it up to 96%. All were solved using starting points within a box constraint as described in the last two columns of Table 1.

Table 1. List of Test Problems

Problem	Refs	n	m	$\bar{l}^T$	$\bar{u}^T$
MOP1	[30]	1	2	-100000	100000
MOP2	[30]	2	2	$(-4, -4)$	(4, 4)
MOP3	[30]	2	2	$(-\pi, -\pi)$	(π, π)
DD1	[8]	5	2	(-20, -20, -20, -20, -20)	(20, 20, 20, 20, 20)
Toi4	[50]	4	2	$(-2, -2, -2, -2)$	(5,5,5,5)
PNR	[43]	2	2	$(-2, -2)$	(2, 2)
MMR1	[41]	2	2	(0, 0)	(1, 1)
AP3	[1]	2	2	(-100, -100)	(100, 100)
Lov5	[36]	3	2	$(-2, -2, -2)$	(2, 2, 2)
IKK1	[30]	2	3	$(-50, -50)$	(50, 50)
TE8-1	[49]	15	3	$(0, \cdots, 0)$	$(10, \cdots, 10)$
TE8-2	[49]	30	3	$(0, \cdots, 0)$	$(1, \cdots, 1)$
TE8-3	[49]	50	3	$(0, \cdots, 0)$	$(1, \cdots, 1)$
MOP5	[30]	2	3	$(-30, -30)$	(30, 30)
MOP7	[30]	2	3	$(-400, -400)$	(400, 400)
FDS-1	[14]	10	3	$(-2, \cdots, -2)$	$(2, \cdots, 2)$
FDS-2	[14]	100	3	$(-2, \cdots, -2)$	$(2, \cdots, 2)$
FDS-3	[14]	200	3	$(-2, \cdots, -2)$	$(2, \cdots, 2)$
SLCDT2	[46]	10	3	$(-100, \cdots, -100)$	$(100, \cdots, 100)$
Toi8	[1]	3	3	(-100, -100)	(100, 100)
AP1	[1]	2	3	(-10, -10)	(10, 10)
BK1	[30]	2	2	$(-5, -5)$	(10, 10)
DGO1	[30]	1	2	-10	13
DGO2	[30]	1	2	-10	13
FF1	[30]	2	2	(-1, -1)	(1, 1)
JOS1-1	[32]	10	2	$(0, \cdots, 0)$	$(1, \cdots, 1)$
JOS1-2	[32]	100	2	$(0, \cdots, 0)$	$(1, \cdots, 1)$
JOS1-3	[32]	1000	2	$(0, \cdots, 0)$	$(1, \cdots, 1)$
MLF1	[30]	1	2	-10	13
MLF2	[30]	2	2	$(-100, -100)$	(100, 100)
TE1	[49]	2	2	$(-1, -1)$	(1, 1)
TE2	[49]	2	2	$(-2, -2)$	(2, 2)
TE4	[49]	10	2	$(-10, \cdots, -10)$	$(10, \cdots, 10)$
TE6	[49]	2	2	$(0, 0)$	(100, 100)
TE7	[49]	3	3	$(0, 0, 0)$	(30,30,30)
SP1	[30]	2	2	$(-100, -100)$	(100, 100)
SSFYY2	[30]	1	2	-100	(100, 100)
SK1	[30]	1	2	-100	100
SK2	[30]	4	2	$(-10, -10, -10, -10)$	$(10, 10, 10, 10)$
VU1	[30]	2	2	$(-3, -3)$	(3, 3)

Table 2. Performance of the proposed methods in comparison with PRP

	WYL				WHS				WLS			
Problem	It	Fe	Ge	time	It	Fe	Ge	time	It	Fe	Ge	time
MOP1	4	1828	75	0.0117	4	1828	75	0.0175	4	1828	75	0.0161
MOP2	1	3	3	0.0047	1	3	3	0.0066	1	3	3	0.0063
MOP3	8	298	129	0.0258	8	291	132	0.0343	8	285	136	0.0365
DD1	11	579	143	0.0338	11	579	141	0.0483	11	579	143	0.0492
Toi4	4	297	84	0.015	4	303	88	0.021	4	297	84	0.0209
PNR	7	106	93	0.0159	7	106	93	0.0233	7	106	93	0.0222
MMR1	10	173	127	0.0283	10	173	127	0.0435	10	173	127	0.0419
AP3	3	330	37	0.0124	3	330	37	0.0196	3	330	37	0.0238
Lov5	2	50	21	0.0086	2	50	21	0.0124	2	50	21	0.0346
IKK1	3	234	51	0.0087	3	235	50	0.01	3	234	51	0.0093
TE8-1	7	681	165	0.0466	7	681	165	0.0479	7	681	165	0.0486
TE8-2	16	1014	268	0.0904	16	1014	268	0.0879	16	1014	268	0.0892
TE8-3	16	1148	287	0.0987	16	1148	287	0.0962	16	1148	287	0.1012
MOP5	1	4	4	0.0061	1	4	4	0.0062	1	4	4	0.0065
MOP7	6	464	93	0.0312	7	527	111	0.0332	8	615	132	0.0391
FDS-1	24	1098	301	0.1264	24	1098	301	0.1265	24	1098	301	0.1274
FDS-2	70	966	804	0.7215	70	966	804	0.7139	70	966	804	0.7255
FDS-3	55	1071	643	0.4158	55	1071	643	0.4144	55	1071	643	0.4102
SLCDT2	3	794	54	0.0215	3	822	54	0.0217	3	794	54	0.0209
Toi8	3	137	52	0.0174	4	144	54	0.019	3	137	52	0.02
AP1	2	183	21	0.0106	2	183	21	0.0121	2	183	21	0.0125
BK1	5	428	95	0.018	5	428	95	0.0198	5	428	95	0.0168
DGO1	2	74	27	0.0069	2	74	27	0.0123	2	74	27	0.0097
DGO2	3	264	59	0.0109	3	264	60	0.0144	3	264	59	0.0137
FF1	14	231	145	0.0376	14	231	145	0.109	14	231	145	0.052
JOS1-1	4	2211	81	0.0216	4	2211	81	0.0287	4	2211	81	0.0278
JOS1-2	4	2211	81	0.0279	4	2211	81	0.0331	4	2211	81	0.0339
JOS1-3	5	2947	107	0.4269	5	2947	107	0.5314	5	2947	107	0.5306
MLF1	1	3	3	0.0057	1	3	3	0.0079	1	3	3	0.0075
MLF2	5	280	77	0.0163	5	283	77	0.0239	5	280	78	0.0226
TE1	4	383	59	0.0123	4	383	59	0.0182	4	383	59	0.0191
TE2	4	147	65	0.0139	4	147	65	0.0209	4	147	65	0.0208
TE4	8	494	120	0.0261	8	499	117	0.037	8	496	120	0.0359
TE6	4	1406	77	0.0165	4	1406	77	0.0253	4	1406	77	0.025
TE7	4	297	45	0.0159	4	298	45	0.0211	4	297	45	0.0217
SP1	6	329	86	0.02	7	329	84	0.0296	6	347	86	0.0289
SSFYY2	3	196	41	0.0089	3	196	41	0.0126	3	196	41	0.0118
SK1	3	203	33	0.0077	3	203	33	0.0125	3	203	33	0.0125
SK2	8	488	128	0.0237	8	489	129	0.0364	8	488	128	0.0353
VU1	98	748	798	0.2659	98	748	798	0.357	98	748	798	0.3605

To ensure a fair and comprehensive algorithmic comparison, we utilized the well-known Dolan and Moré performance profile [10]. The performance profile assesses the statistical performance of methods ($s \in S$) in solving individual problems ($p \in P$). The performance ratio $\varphi_{p,s}$ is defined as:

$$\varphi_{p,s} = \frac{c_{p,s}}{\min c_{p,s} : s \in S}. \tag{4.2}$$

All methods share the same stopping criteria: $v(z) \geq -5 \times eps^{\frac{1}{2}}$ or reaching the maximum number of iterations limit. The overall performance of a method s within a factor τ is quantified by the cumulative distribution function $\rho_s : [0, \infty) \to [0, 1]$ which is given as

$$\rho_s(\tau) = \frac{1}{\mid P \mid} \mid \{p \in P : \varphi_{p,s} \leq \tau\} \mid . \tag{4.3}$$

Table 3. Continuation of Table 2

	WHS*				WLS*				PRP			
Problem	It	Fe	Ge	time	It	Fe	Ge	time	It	Fe	Ge	time
MOP1	4	1830	75	0.0174	4	1830	75	0.0169	4	1953	78	0.0557
MOP2	1	3	3	0.0067	1	3	3	0.0062	1	3	3	0.0173
MOP3	8	282	135	0.0378	8	292	132	0.0364	7	371	144	0.1004
DD1	11	547	132	0.0485	11	547	136	0.0481	13	512	149	0.1506
Toi4	4	301	87	0.0217	4	301	87	0.0208	5	413	108	0.07
PNR	7	106	93	0.0242	7	106	93	0.0233	7	113	96	0.0907
MMR1	10	173	127	0.0433	10	173	127	0.0418	10	177	127	0.1155
AP3	3	330	37	0.0489	3	330	37	0.0206	4	844.5	50	0.0164
Lov5	2	50	21	0.0322	2	50	21	0.0126	2	50	21	0.0067
IKK1	3	238	52	0.0085	3	238	52	0.0087	3	261	53	0.0083
TE8-1	7	681	161	0.0463	7	681	161	0.0472	7	905	195	0.0926
TE8-2	16	1014	268	0.0911	16	1014	268	0.0924	16	1062	268	0.2451
TE8-3	16	1148	287	0.1031	16	1148	287	0.1059	16	1177	287	0.2734
MOP5	1	4	4	0.0064	1	4	4	0.0062	1	4	4	0.016
MOP7	8	673	133	0.0365	9	808	156	0.0462	6	564	103	0.0793
FDS-1	24	1098	301	0.129	24	1098	301	0.1271	24	1098	301	0.3404
FDS-2	70	966	804	0.7353	70	966	804	0.7339	70	967	802	0.5787
FDS-3	55	1071	643	0.421	55	1071	643	0.414	55	1071	645	0.3489
SLCDT2	3	817	54	0.0219	3	817	54	0.0222	4	869	57	0.017
Toi8	3	140	52	0.0185	3	139	52	0.0187	4	450	75	0.0152
AP1	2	183	21	0.0099	2	183	21	0.0124	2	183	21	0.0044
BK1	5	428	95	0.0174	5	428	95	0.0162	5	510	110	0.0197
DGO1	2	74	27	0.0091	2	74	27	0.0089	2	74	27	0.012
DGO2	3	266	63	0.0141	3	266	63	0.0134	4	570	72	0.0181
FF1	14	231	145	0.0616	14	231	145	0.056	14	194	141	0.1578
JOS1-1	4	2211	81	0.0284	4	2211	81	0.0274	4	2211	81	0.077
JOS1-2	4	2211	81	0.0342	4	2211	81	0.0338	4	2211	81	0.0903
JOS1-3	5	2947	107	0.5339	5	2947	107	0.5332	5	2947	107	1.4922
MLF1	1	3	3	0.008	1	3	3	0.0075	1	3	3	0.0195
MLF2	5	273	74	0.0227	5	273	75	0.0217	5	365	97	0.0701
TE1	4	383	59	0.0194	4	383	59	0.0187	4	413	66	0.0568
TE2	4	147	64	0.0218	4	147	64	0.0208	4	160	72	0.0616
TE4	11	583	154	0.0519	10	567	148	0.0454	8	531	122	0.1001
TE6	4	1415	79	0.0262	4	1415	79	0.0235	5	2101	128	0.0781
TE7	4	292	44	0.0219	4	292	44	0.0212	4	308	49	0.0626
SP1	7	334	90	0.0343	7	335	90	0.0308	7	468	90	0.0836
SSFYY2	3	197	41	0.013	3	197	41	0.0124	3	205	43	0.0375
SK1	3	203	33	0.0132	3	203	33	0.0126	3	203	36	0.0366
SK2	8	469	127	0.0354	8	465	127	0.0353	8	509	135	0.1045
VU1	98	748	798	0.3829	98	748	798	0.3881	101	831	821	0.5794

Figures 1-3 display $\rho_s(\tau)$ on the y-axis against a logarithmic scale (base 2) of τ on the x-axis, we use τ for simplicity. For instance, $\rho_s(0)$ indicates the percentage of problems where a solver $s \in S$ outperforms others. A solver with the top-right curve is the most robust. This makes the performance profile a measure of method efficiency and robustness.

Based on the presented data in Figures 1-3, it is evident that Figure 1, which represents the median number of iterations (It), shows that WYL, WLS, WHS, WHS*, and WLS* have fewer iterations than the PRP method, making them more efficient and robust than PRP. Similarly, in Figure 2, which displays the median number of function evaluations (Fe), the methods WYL, WHS, WLS, WHS*, and WLS* evaluated fewer functions than the PRP method. Additionally, Figure 3, representing the median number of gradient evaluations (Ge), indicates that the methods WYL, WLS, WHS*, WHS, and WLS* evaluated fewer gradients than the PRP method. Finally, Figure 4, which shows the median time taken for computing iterations, function, and gradient evaluations reveals that WYL requires less time than all the other methods to achieve a stationary or Q-critical point, followed by WLS, WHS, WHS*, and WLS*. It is evident that PRP takes more time than all other methods. These results demonstrate the competitiveness and significance of the proposed CG methods within this setting.

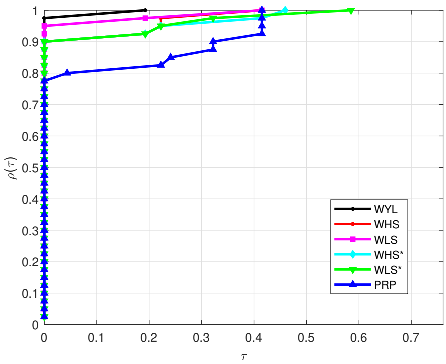

Fig. 1. Performance on It

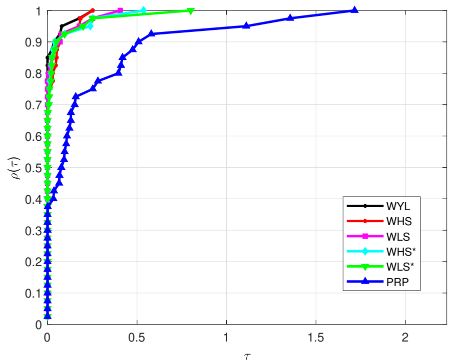

Fig. 2. Performance on Fe

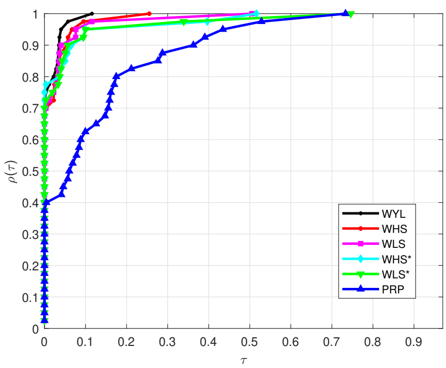

Fig. 3. Performance on Ge

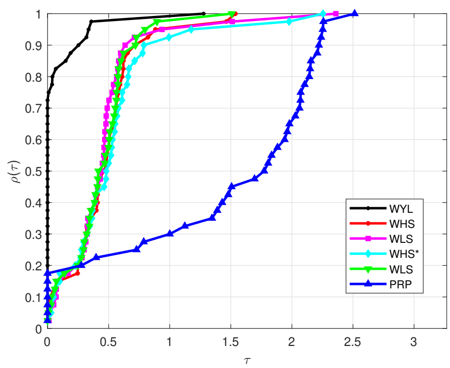

Fig. 4. Performance on CPU Time

To assess the effectiveness of Algorithm 1 with the WYL parameter in accurately generating Pareto frontiers, we examine four distinct problem instances: JOS1, SP1, LOV5, and MOP7. For each problem instance, we employ 300 randomly generated starting points within their respective search domains. Figure 5 illustrates the shapes of the approximate Pareto frontiers generated by Algorithm 1 with the WYL parameter. Within Figure 5, each blue point represents the final iteration, while the starting points are indicated by the beginning of the straight line. The outcomes depicted in Figure 5 demonstrate that Algorithm 1 with the WYL parameter effectively estimates the Pareto fronts for the considered problems, utilizing an appropriate number of starting points.

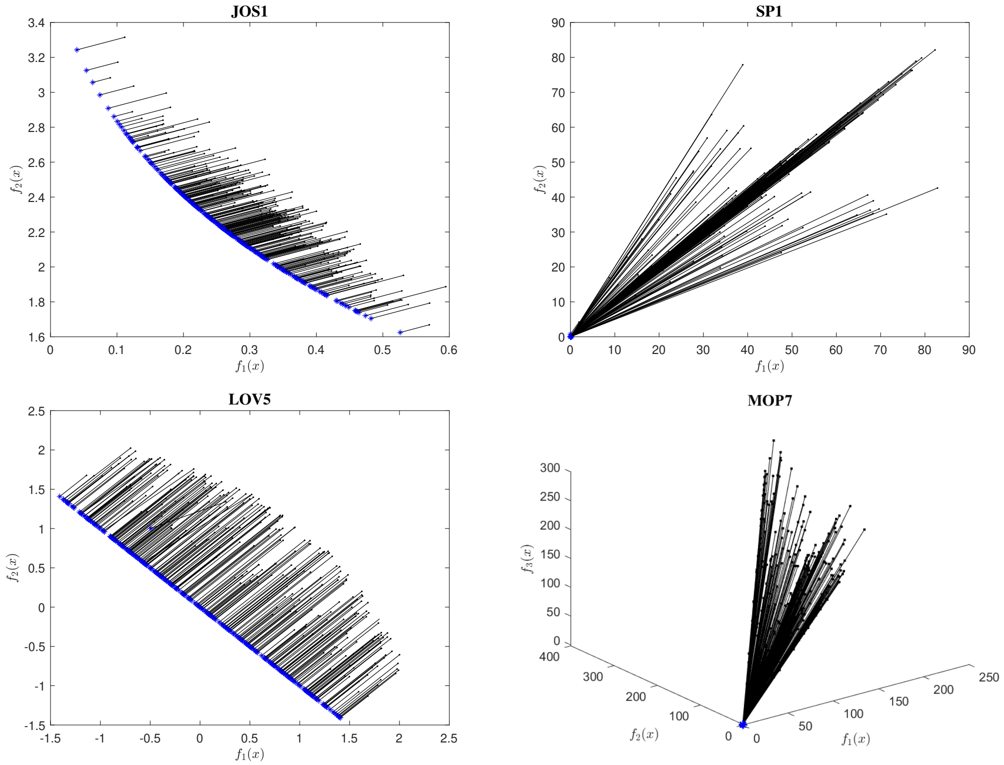

Fig. 5. Pareto frontiers of some selected problems JOS1, SP1, LOV5 and MOP7 are generated by Algorithm 1 with WYL parameter to show its ability in generating the Pareto fronts

5. Conclusion

We proposed five CG methods designed for solving VOPs. The first three methods are the WYL and its HS and LS types. Although these three methods lost their descent property in the vector setting, we established their global convergence by employing the Wolfe line search strategy. To capture the spirit of the sufficient descent condition, we modified the HS and LS types of the WYL and introduced two new methods that achieved this property with Wolfe line search; global convergence was also established using Wolfe line search. Some numerical experiments are presented by considering a considerable number of convex and non-convex multiobjective optimization test problems, demonstrating that our proposed methods are promising.

Acknowledgments The authors express their gratitude for the financial assistance received from the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), King Mongkut's University of Technology Thonburi (KMUTT). The first author acknowledges the support provided by the Petchra Pra Jom Klao PhD scholarship of KMUTT (Contract No. 23/2565). Additionally, the authors are grateful to the authors referenced in [28] for making their codes available.

References

[1] M. A. Ansary and G. Panda, A modified quasi-Newton method for vector optimization problem, Optimization, 64 (2015), 2289–2306.

[2] S. Babaie-Kafaki, A survey on the Dai–Liao family of nonlinear conjugate gradient methods, RAIRO - Oper. Res., 57 (2023), 43–58.

[3] J. Bello Cruz, A subgradient method for vector optimization problems, SIAM J. Optim., 23 (2013), 2169–2182.

[4] H. Bonnel, A.N. Iusem, and B.F. Svaiter, Proximal methods in vector optimization, SIAM J. Optim., 15 (2005), 953–970.

[5] W. Chen, Y. Zhao, and X. Yang, Conjugate gradient methods without line search for multiobjective optimization, arXiv preprint arXiv:2312.02461, 2023.

[6] Y. Dai and Y.-X. Yuan, Convergence properties of the Fletcher-Reeves method, IMA J. Numer. Anal., 16(1996), 155–164.

[7] Y.-H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim., 10 (1999), 177–182.

[8] I. Das and J.E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim., 8 (1998), 631–657.

[9] P. De, J.B. Ghosh, and C.E. Wells, On the minimization of completion time variance with a bicriteria extension, Operations Research, 40 (1992), 1148–1155.

[10] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program., 91 (2002), 201–213.

[11] L.G. Drummond and B.F. Svaiter, A steepest descent method for vector optimization, J. Comput. Appl. Math., 175 (2005), 395–414.

[12] R. Fletcher, Unconstrained optimization, Practical methods of optimization, 1, 1980.

[13] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, The computer journal, 7(1964), 149–154.

[14] J. Fliege, L.G. Drummond, and B.F. Svaiter, Newton's method for multiobjective optimization, SIAM J. Optim., 20 (2009), 602–626.

[15] J. Fliege and B.F. Svaiter, Steepest descent methods for multicriteria optimization, Math. Methods Oper. Res., 51 (2000), 479–494.

[16] J. Fliege and L.N. Vicente, Multicriteria approach to bilevel optimization, J. Optim. Theory Appl., 131 (2006), 209–225.

[17] J. Fliege and R. Werner, Robust multiobjective optimization & applications in portfolio optimization, Eur. J. Oper. Res., 234 (2014), 422–433.

[18] E.H. Fukuda and L.M.G. Drummond, A survey on multiobjective descent methods, Pesqui. Oper., 34 (2014), 585–620.

[19] J.C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim., 2 (1992), 21–42.

[20] M. Gonçalves, F. Lima, and L. Prudente, Globally convergent Newton-type methods for multiobjective optimization, Comput. Optim. Appl., 83 (2022), 403–434.

[21] M.L. Gonçalves, F. Lima, and L. Prudente, A study of Liu-Storey conjugate gradient methods for vector optimization, Appl. Math. Comput., 425 (2022), 127099.

[22] M.L. Gonçalves and L. Prudente, On the extension of the Hager–Zhang conjugate gradient method for vector optimization, Comput. Optim. Appl., 76 (2020), 889–916.

[23] M. Gravel, J.M. Martel, R. Nadeau, W. Price, and R. Tremblay, A multicriterion view of optimal resource allocation in job-shop production, Eur. J. Oper. Res., 61 (1992), 230–244.

[24] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, Pac. J. Optim., 2 (2006), 35–58.

[25] Q.R. He, C.R. Chen, and S.-J. Li, Spectral conjugate gradient methods for vector optimization problems, Comput. Optim. Appl., 86 (2023), 457–489.

[26] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving, J. Res. NBS., 49 (1952), 409.

[27] T.S. Hong, D.L. Craft, F. Carlsson, and T.R. Bortfeld, Multicriteria optimization in intensity-modulated radiation therapy treatment planning for locally advanced cancer of the pancreatic head, Int. J. Radiat. Oncol. Biol. Phys., 72 (2008), 1208–1214.

[28] Q. Hu, L. Zhu, and Y. Chen, Alternative extension of the Hager–Zhang conjugate gradient method for vector optimization, Comput. Optim. Appl., (2024), 1–34.

[29] H. Huang and S. Lin, A modified Wei–Yao–Liu conjugate gradient method for unconstrained optimization, Appl. Math. Comput., 231 (2014), 179–186.

[30] S. Huband, P. Hingston, L. Barone, and L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput., 10 (2006), 477–506.

[31] J. Jahn, A. Kirsch, and C. Wagner, Optimization of rod antennas of mobile phones, Math. Methods Oper. Res., 59 (2004), 37–51.

[32] Y. Jin, M. Olhofer, and B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how, In Proceedings of the genetic and evolutionary computation conference, pages 1042–1049, 2001.

[33] J. Jahn, Scalarization in vector optimization, Math. Program., 29 (1984), 203–218.

[34] T.M. Leschine, H. Wallenius, and W A. Verdini, Interactive multiobjective analysis and assimilative capacity-based ocean disposal decisions, Eur. J. Oper. Res., 56 (1992), 278–289.

[35] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms, part 1: theory, J. Optim. Theory Appl., 69 (1991), 129–137.

[36] A. Lovison, Singular continuation: Generating piecewise linear approximations to pareto sets via global analysis, SIAM J. Optim., 21 (2011), 463–490.

[37] D.T. Luc, Scalarization of vector optimization problems, J. Optim. Theory Appl., 55 (1987), 85–102.

[38] D.T. Luc, Theory of vector optimization, Springer, 1989.

[39] L. Lucambio Pérez and L. Prudente, A wolfe line search algorithm for vector optimization, ACM Trans. Math. Softw. , 45 (2019), 1–23.

[40] L.R. Lucambio Pérez and L.F. Prudente, Nonlinear conjugate gradient methods for vector optimization, SIAM J. Optim., 28 (2018), 2690–2720.

[41] E. Miglierina, E. Molho, and M.C. Recchioni, Box-constrained multi-objective optimization: a gradient-like method without "a priori" scalarization, Eur. J. Oper. Res., 188 (2008), 662–682.

[42] E. Polak and G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, Revue française d'informatique et de recherche opérationnelle. Série rouge, 3 (1969):35–43.

[43] M. Preuss, B. Naujoks and G. Rudolph, Pareto set and emoa behavior for simple multimodal multiobjective functions, In PPSN, pages 513–522. Springer, 2006.

[44] S. Qu, M. Goh, and F.T. Chan, Quasi-Newton methods for solving multiobjective optimization, Oper. Res. Lett., 39 (2011), 397–399.

[45] O. Schütze, A. Lara, and C.C. Coello, The directed search method for unconstrained multi-objective optimization problems, Proceedings of the EVOLVE–A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation, pages 1–4, 2011.

[46] O. Schütze, M. Laumanns, C.A. Coello, M. Dellnitz, and E.-G. Talbi, Convergence of stochastic search algorithms to finite size pareto set approximations, J. Glob. Optim., 41 (2008), 559–577.

[47] Y. Shengwei, Z. Wei, and H. Huang, A note about WYL's conjugate gradient method and its applications, Appl. Math. Comput., 191 (2007), 381–388.

[48] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. Di Stefano, Real-world applications of multiobjective optimization, Multiobjective optimization: interactive and evolutionary approaches, pages 285–327, 2008.

[49] J. Thomann and G. Eichfelder, Numerical results for the multiobjective trust region algorithm mht, Data in brief, 25:104103, 2019.

[50] P. Toint, Test problems for partially separable optimization and results for the routine pspmin. the university of namur, department of mathematics, Technical report, Belgium, Tech. Rep, 1983.

[51] Z. Wei, H. Huang, and Y. Tao, A modified Hestenes-Stiefel conjugate gradient method and its convergence, J. Math. Res. Expo., 30 (2010), 297–308.

[52] Z. Wei, S. Yao, and L. Liu, The convergence properties of some new conjugate gradient methods, Appl. Math. Comput., 183 (2006), 1341–1350,.

[53] J. Yahaya, I. Arzuka, and M. Isyaku, Descent modified conjugate gradient methods for vector optimization problems, Bangmod Int. J. Math. Com. Sci., 9 (2023), 72–91.

[54] J. Yahaya and P. Kumam, Efficient hybrid conjugate gradient techniques for vector optimization, Results Control Optim. , (2023), 100348.

[55] J. Yahaya, P. Kumam, and J. Abubakar, Efficient nonlinear conjugate gradient techniques for vector optimization problems, Carpathian J. Math., 40 (2024), 515–533.

[56] L. Zhang, An improved Wei–Yao–Liu nonlinear conjugate gradient method for optimization computation, Appl. Math. Comput., 215 (2009), 2269–2274.