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 KMUTT Fixed 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.

b Department of Mathematics, Faculty of Physical Sciences, Ahmadu Bello University Zaria, Kaduna, Nigeria.

c Department of Mathematics, Faculty of Natural and Applied Sciences, Sule Lamido University Kafin Hausa, Jigawa, 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

This is an open access article under the Diamond Open Access.
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 $\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R}$. The parameters include the $\beta_k$ of Polak-Ribière-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

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

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 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 $\mathbb{R}^m$, $\preceq_Q$, generated by $Q$ is $a \preceq_Q b \iff b - a \in Q$, and $\ll_Q$, generated by $\text{int}(Q)$ is $a \ll_Q b \iff b - a \in \text{int}(Q)$. If $Q = \mathbb{R}^m_+$, then problem (1.1) is considered to be multiobjective optimization problem, and if $Q = \mathbb{R}^m_+$ 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 exist \( z \in \mathbb{R}^n \) such that \( F(z) \preceq_Q F(\bar{z}) \) 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 exist \( z \in \mathbb{R}^n \) 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 \( C \subseteq Q^* \setminus \{0\} \) is compact, then \( Q^* \) is defined as the conic hull of a convex hull of \( C \):

\[
Q^* = \text{cone}(\text{conv}(C)). \tag{2.1}
\]

Again,

\[
-Q = \{ z \in \mathbb{R}^m \mid \langle z, p \rangle \leq 0, \ \forall \ p \in Q^* \}, \quad -\text{int}(Q) = \{ z \in \mathbb{R}^m \mid \langle z, p \rangle < 0, \ \forall \ p \in Q^* \setminus \{0\} \}.
\]
For a given point $z$, the term $\text{Image}(JF(z))$ 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

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

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 $JF(\bar{z})h \in -\text{int}(Q)$. 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 $F(\bar{z} + \tilde{r}h) \prec_Q F(\bar{z})$, for all $0 < \tilde{r} < s$, 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

$$C = \{ p \in Q^n \mid ||p|| = 1 \}, \quad (2.3)$$

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

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

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

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

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 $\nu : \mathbb{R}^n \to \mathbb{R}$, respectively by

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

and

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

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 $\nu(z) = -\frac{||\nabla F(z)||^2}{2}$.

Now, we can describe a CG method as

$$z_{k+1} = z_k + \alpha_k d_k, \quad k \geq 1, \quad (2.8)$$

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

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

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

$$\beta_k^{FR} := \frac{\phi(z_k, u(z_k))}{\phi(z_{k-1}, u(z_{k-1}))}, \quad \beta_k^{CD} := \frac{\phi(z_k, u(z_k))}{\phi(z_{k-1}, d_{k-1})}, \quad (2.10)$$
\[
\beta_k^{DY} := \frac{-\phi(z_k, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}, \quad \beta_k^{PRP} := \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, u(z_{k-1}))},
\]
\[
\beta_k^{HS} := \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}, \quad \beta_k^{LS} := \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})},
\]
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
\[
\phi(z, d) < 0.
\]
A point \(z\) is \(Q\)-critical point for \(F\) if
\[
\phi(z, d) \geq 0,
\]
for all \(d \in \mathbb{R}^n\). A direction \(d\) is said to satisfies \textit{sufficient descent condition} (SDC) at \(z\) if
\[
\phi(z, d) \leq c\phi(z, u(z)),
\]
for some \(c > 0\).

\textbf{Lemma 2.2.} \cite{11}. \textit{Suppose} \(F : \mathbb{R}^n \to \mathbb{R}^m\) \textit{is in} \(C^1\). \textit{Then}, the statements below hold:

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

(b) The mapping \((z, d) \mapsto \phi(z, d)\) is continuous;

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

(d) Let \(\|JF(z) - JF(z')\| \leq L\|z - z'\|, \) \textit{then} \(|\phi(z, d) - \phi(z', d)| \leq L\|d\|\|z - z'\|.

Consider the following convex quadratic problem
\[
\text{Minimize } \alpha + \frac{1}{2}\|u\|^2,
\]
subject to \(|JF(z)u_i| \leq \alpha, \) \(i = 1, 2, \ldots, m,\)

with linear inequality constraints, see for instance, \cite{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
\[
\phi(x + \alpha d, d) = 0.
\]

We now give the vector Wolfe conditions that was introduced by Lucambio Pérez and Prudente \cite{39}.

\textbf{Definition 2.3.} \cite{40} \textit{Let} \(d \in \mathbb{R}^n\) \textit{be a} \(Q\)-\textit{descent direction and} \(e \in Q\), \textit{then we have}
\[
0 < \langle p, e \rangle \leq 1,
\]
\textit{for all} \(p \in C\). \textit{Now,} \(\alpha > 0\) \textit{satisfies the \textit{standard Wolfe condition} (WWC) if}
\[
F(x + \alpha d) \preceq_Q F(x) + \rho \alpha \phi(x, d)e
\]
\[ \phi(z + \alpha d, d) \geq \sigma \phi(z, d), \]  
\[ (2.19) \]

where \(0 < \rho < \sigma < 1\). Furthermore, \(\alpha > 0\) satisfies the strong Wolfe condition (SWC) if
\[ F(z + \alpha d) \preceq Q F(z) + \rho \alpha \phi(z, d)e \]
\[ |\phi(z + \alpha d, d)| \leq \sigma |\phi(z, d)|. \]  
\[ (2.20) \]

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 \(\Delta\) for which the \(L := \{z \mid F(z) \preceq Q F(z_1)\} \subset \Delta\), 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 \(L := \{z \mid F(z) \preceq Q F(z_1)\}\) is bounded.

Note that, by Assumption 3.2 we have that for any \(\{z_k\}\) in \(L\), there exists \(\bar{M} > 0\) s.t
\[ \|z_k\| \leq \bar{M}, \]  
\[ (3.1) \]

for all \(k\). Therefore, we have from Lemma 2.2(d) that there exists \(\gamma > 0\) s.t
\[ \|JF(z_k)\| \leq \gamma, \]  
\[ (3.2) \]

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
\[ \|u(z_k)\| \leq \delta, \]  
\[ (3.3) \]

for all \(k\). From (2.5), Lemma 2.4 (b), (3.2) and (3.3), we have
\[ 0 < -\phi(z_k, u(z_k)) \leq -\langle JF(z_k)u(z_k), q\rangle \leq \|JF(z_k)\|\|u(z_k)\| \leq \delta \gamma, \]  
\[ (3.4) \]

with \(\|q\| = 1\).

We state the general prototype of the considered CG algorithm for VOPs.
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) $\phi_k$ 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 $\beta_k$ 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

$$0 < \bar{\delta} \leq \|u(z_k)\|, \quad (3.11)$$
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
\[
\|s_{k-1}\| \leq \lambda \implies |\beta_k| \leq \frac{1}{2q},
\]
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) \( \beta_k \) is nonnegative;
(b) \( d_k \) is a Q-descent direction of \( F \) at \( z_k \);
(c) \( \alpha_k \) 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 \( \alpha_k \) 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 \( \epsilon \) such that
\[
|\beta_k| \leq \epsilon \|s_{k-1}\|, \quad \forall \ k \geq 2. \tag{3.13}
\]
Now, assume that (3.11) holds. Then, by (3.3) and (3.11), we have
\[
0 < \bar{\delta} \leq \|u(z_k)\| \leq \delta, \quad \forall \ k \geq 2. \tag{3.14}
\]
Additionally, by (3.2) and Lemma 2.4 (b), we have
\[
\frac{\bar{\delta}^2}{2} \leq -\phi(z_k, u(z_k)) \leq \bar{\delta} \gamma. \tag{3.15}
\]
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
\[
-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k)) \leq L\|z_k - z_{k-1}\|\|u(z_k)\| \leq L\delta \|s_{k-1}\|, \tag{3.17}
\]
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

$$|\beta_k| = \beta_k^{WY} = \frac{-\phi(z_k, u(z_k)) + \|u(z_k)\| \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, u(z_{k-1}))}$$

where $\phi(z_k, u(z_k)) = \phi(z_{k-1}, u(z_{k-1}) + u(z_k)\|\phi(z_{k-1}, u(z_k))$.

$$\leq \frac{\delta}{\delta} \left( \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{\phi(z_{k-1}, u(z_{k-1}))} \right)$$

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

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

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

$$|\beta_k| \leq \frac{2L\delta^2\|s_{k-1}\|}{\delta^3}, \quad \forall k \geq 2,$$

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 $\beta_k = \beta_k^{WY}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. If $d_k$ satisfies the SDC and $\alpha_k$ satisfies condition (2.20). Then,

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

**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

$$\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1}) \geq -(1 - \sigma)\phi(z_{k-1}, d_{k-1})$$

$$\geq -c(1 - \sigma)\phi(z_{k-1}, u(z_{k-1})) > 0. \quad (3.19)$$

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

$$|\beta_k| = \beta_k^{WY} = \frac{-\phi(z_k, u(z_k)) + \|u(z_k)\| \phi(z_{k-1}, u(z_k))}{\phi(z_{k-1}, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}$$

where $\phi(z_k, u(z_k)) = \phi(z_{k-1}, d_{k-1}) + u(z_k)\|\phi(z_{k-1}, u(z_k))$.

$$\leq \frac{\delta}{\delta} \left( \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{\phi(z_{k-1}, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \right)$$

This completes the proof.
Now Using (3.19), we get

\[ |\beta_k| \leq \frac{\delta}{\delta} \left| \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-c(1 - \sigma)\phi(z_{k-1}, u(z_{k-1}))} \right|. \]

By (3.15) and (3.17), we have

\[ |\beta_k| \leq \frac{\delta}{\delta} \left| \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-c(1 - \sigma)\phi(z_{k-1}, u(z_{k-1}))} \right| \leq \frac{2L\delta^2\|s_{k-1}\|}{c(1 - \sigma)\delta^3}. \]

Thus,

\[ |\beta_k| \leq \frac{2L\delta^2\|s_{k-1}\|}{c(1 - \sigma)\delta^3}, \quad (3.20) \]

where \( \epsilon = \frac{2L\delta^2}{c(1 - \sigma)\delta^3} \). 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 \( \alpha_k \) satisfies condition (2.20). Then,

\[ \lim \inf_{k \to \infty} \|u(z_k)\| = 0. \quad (3.21) \]

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

\[ -\phi(z_{k-1}, d_{k-1}) \geq -c\phi(z_{k-1}, u(z_{k-1})) > 0. \quad (3.22) \]

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

\[ |\beta_k| = |\beta_k^{WLS}| = \frac{-\phi(z_k, u(z_k)) + \|u(z_k)\|\phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})} \]

\[ \leq \frac{\delta}{\delta} \left( \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-\phi(z_{k-1}, d_{k-1})} \right) \]

\[ \leq \frac{\delta}{\delta} \left( \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-c\phi(z_{k-1}, u(z_{k-1}))} \right). \]

By (3.15) and (3.17), we have

\[ |\beta_k| \leq \frac{\delta}{\delta} \left( \frac{-\phi(z_k, u(z_k)) + \phi(z_{k-1}, u(z_k))}{-c\phi(z_{k-1}, u(z_{k-1}))} \right) \leq \frac{2L\delta^2\|s_{k-1}\|}{c\delta^3}, \]

which implies that

\[ |\beta_k| \leq \frac{2L\delta^2\|s_{k-1}\|}{c\delta^3}, \quad (3.23) \]

where \( \epsilon = \frac{2L\delta^2}{c\delta^3} \). 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 \( \beta_k = \max \{ \beta_{WHS}^k, 0 \} \) if \( \phi(z_{k-1}, u(z_k)) > 0 \) or \( \beta_k = 0 \) otherwise. Suppose \( \alpha_k \) 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 \( \beta_k = \beta_{WHS}^k \) and \( \phi(z_k, d_{k-1}) > 0 \), then \( \phi(z_{k-1}, u(z_k)) > 0 \) and consequently, we get
\[
\phi(z_k, d_k) \leq \phi(z_k, u(z_k)) + \left( \frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_k)\|}{\|u(z_{k-1})\|} \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \right) \phi(z_k, d_{k-1})
\leq \phi(z_k, u(z_k)) + \frac{-\phi(z_k, u(z_k)) \phi(z_k, d_{k-1})}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}
\leq \left( 1 - \frac{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \right) \phi(z_k, u(z_k))
\leq \left( \frac{1}{1 - \sigma} \right) \phi(z_k, u(z_k)),
\]

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

This completes the proof. \( \square \)

**Theorem 3.9.** Let Assumptions 3.1 and 3.2 hold. Consider Algorithm 1 such that the sequence \( \{z_k\} \) is generated with \( \beta_k = \max \{ \beta_{WHS}^k, 0 \} \) if \( \phi(z_{k-1}, u(z_k)) > 0 \) or \( \beta_k = 0 \) otherwise. If \( \alpha_k \) 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 \( \epsilon \) 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 $\epsilon$. So, we only need to consider the case when $\beta_k = \beta_k^{\text{WLS}^*} > 0$. In this case, $\phi(z_{k-1}, u(z_k)) > 0$ and following (3.19), we have

$$|\beta_k| = \beta_k^{\text{WLS}^*} = \frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_1)\|}{\|u(z_k)\|} \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \leq \frac{-\phi(z_k, u(z_k)) + \frac{\|u(z_1)\|}{\|u(z_k)\|} \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})} \leq \frac{2L\delta^2 \|s_{k-1}\|}{c(1 - \sigma)\delta^3}.$$

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

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

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

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 $\beta_k = \max \{\beta_k^{\text{WLS}^*}, 0\}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. Suppose that $\alpha_k$ 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 $\beta_k^{\text{WLS}^*} \geq 0$, we have

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

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

$$\phi(z_k, d_k) \leq \phi(z_k, u(z_k)) + \left(\frac{-\phi(z_k, u(z_k)) - \frac{\|u(z_1)\|}{\|u(z_k)\|} \phi(z_{k-1}, u(z_k))}{\phi(z_k, d_{k-1}) - \phi(z_{k-1}, d_{k-1})}\right) \phi(z_k, d_{k-1}) \leq \phi(z_k, u(z_k)) + \frac{\phi(z_k, u(z_k))\phi(z_k, d_{k-1})}{\phi(z_{k-1}, d_{k-1})}.$$

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 $\beta_k = \max \{\beta_k^{\text{WLS}^*}, 0\}$ if $\phi(z_{k-1}, u(z_k)) > 0$ or $\beta_k = 0$ otherwise. If $\alpha_k$ satisfies condition (2.20). Then,

$$\liminf_{k \to \infty} \|u(z_k)\| = 0. \quad (3.29)$$
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 $\epsilon$ such that

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

For the case when $\beta_k = 0$, the desired inequality (3.30) is satisfied for any $\epsilon$. 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

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

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

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

where $\epsilon = \frac{2L\delta^2}{c\delta^3}$. 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 $e = [1, \ldots, 1]^T \in \mathbb{R}^m$, $Q = \mathbb{R}^m_+$, and $C = \{e_1, e_2, \ldots, e_m\} \subset \mathbb{R}^m$.

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 $\beta_k^{WYL}$;
- **WHS**: is the parameter $\beta_k^{WHS}$;
- **WLS**: is the parameter $\beta_k^{WLS}$;
- **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: $\rho = 10^{-4}$, $\sigma = 10^{-1}$, $\mu_1 = 10^{-1}$, $\mu_2 = \frac{1}{2}$.

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 $v(z) \geq -5 \times \varepsilon_1^{\frac{1}{2}}$. Here, $\varepsilon_1$ corresponds to the machine precision which is approximately $2.22 \times 10^{-16}$. Alternatively, the process terminates if the maximum number of iterations, $\text{max.It} = 5000$, is exceeded.

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

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

where $\lambda_i = \frac{1}{\max\{1, \|\nabla F_i(z_1)\|_\infty\}}$, $i = 1, 2, 3, \ldots, m$, 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 $\{z \in \mathbb{R}^n | \bar{l} \leq z \leq \bar{u}\}$, 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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Refs</th>
<th>n</th>
<th>m</th>
<th>$\bar{j}^T$</th>
<th>$\bar{u}^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOP1</td>
<td>[30]</td>
<td>1</td>
<td>2</td>
<td>-100000</td>
<td>100000</td>
</tr>
<tr>
<td>MOP2</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-4, -4)</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>MOP3</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-\pi, -\pi)</td>
<td>(\pi, \pi)</td>
</tr>
<tr>
<td>DD1</td>
<td>[8]</td>
<td>5</td>
<td>2</td>
<td>(-20, -20, -20, -20, -20)</td>
<td>(20, 20, 20, 20, 20)</td>
</tr>
<tr>
<td>Toi4</td>
<td>[50]</td>
<td>4</td>
<td>2</td>
<td>(-2, -2, -2, -2)</td>
<td>(5.5, 5.5, 5.5)</td>
</tr>
<tr>
<td>PNR</td>
<td>[43]</td>
<td>2</td>
<td>2</td>
<td>(-2, -2)</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>MMR1</td>
<td>[41]</td>
<td>2</td>
<td>2</td>
<td>(0, 0)</td>
<td>(1, 1)</td>
</tr>
<tr>
<td>AP3</td>
<td>[1]</td>
<td>2</td>
<td>2</td>
<td>(-100, -100)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>Lov5</td>
<td>[36]</td>
<td>3</td>
<td>2</td>
<td>(-2, -2, -2)</td>
<td>(2, 2, 2)</td>
</tr>
<tr>
<td>IKK1</td>
<td>[30]</td>
<td>2</td>
<td>3</td>
<td>(-50, -50)</td>
<td>(50, 50)</td>
</tr>
<tr>
<td>TE8-1</td>
<td>[49]</td>
<td>15</td>
<td>3</td>
<td>(0, \cdots, 0)</td>
<td>(10, \cdots, 10)</td>
</tr>
<tr>
<td>TE8-2</td>
<td>[49]</td>
<td>30</td>
<td>3</td>
<td>(0, \cdots, 0)</td>
<td>(1, \cdots, 1)</td>
</tr>
<tr>
<td>TE8-3</td>
<td>[49]</td>
<td>50</td>
<td>3</td>
<td>(0, \cdots, 0)</td>
<td>(1, \cdots, 1)</td>
</tr>
<tr>
<td>MOP5</td>
<td>[30]</td>
<td>2</td>
<td>3</td>
<td>(-30, -30)</td>
<td>(30, 30)</td>
</tr>
<tr>
<td>MOP7</td>
<td>[30]</td>
<td>2</td>
<td>3</td>
<td>(-400, -400)</td>
<td>(400, 400)</td>
</tr>
<tr>
<td>FDS-1</td>
<td>[14]</td>
<td>10</td>
<td>3</td>
<td>(-2, \cdots, -2)</td>
<td>(2, \cdots, 2)</td>
</tr>
<tr>
<td>FDS-2</td>
<td>[14]</td>
<td>100</td>
<td>3</td>
<td>(-2, \cdots, -2)</td>
<td>(2, \cdots, 2)</td>
</tr>
<tr>
<td>FDS-3</td>
<td>[14]</td>
<td>200</td>
<td>3</td>
<td>(-2, \cdots, -2)</td>
<td>(2, \cdots, 2)</td>
</tr>
<tr>
<td>SLCDT2</td>
<td>[46]</td>
<td>10</td>
<td>3</td>
<td>(-100, \cdots, -100)</td>
<td>(100, \cdots, 100)</td>
</tr>
<tr>
<td>Toi8</td>
<td>[1]</td>
<td>3</td>
<td>3</td>
<td>(-100, -100)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>AP1</td>
<td>[1]</td>
<td>2</td>
<td>3</td>
<td>(-10, -10)</td>
<td>(10, 10)</td>
</tr>
<tr>
<td>BK1</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-5, -5)</td>
<td>(10, 10)</td>
</tr>
<tr>
<td>DGO1</td>
<td>[30]</td>
<td>1</td>
<td>2</td>
<td>-10</td>
<td>13</td>
</tr>
<tr>
<td>DGO2</td>
<td>[30]</td>
<td>1</td>
<td>2</td>
<td>-10</td>
<td>13</td>
</tr>
<tr>
<td>FF1</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-1, -1)</td>
<td>(1, 1)</td>
</tr>
<tr>
<td>JOS1-1</td>
<td>[32]</td>
<td>10</td>
<td>2</td>
<td>(0, \cdots, 0)</td>
<td>(1, \cdots, 1)</td>
</tr>
<tr>
<td>JOS1-2</td>
<td>[32]</td>
<td>100</td>
<td>2</td>
<td>(0, \cdots, 0)</td>
<td>(1, \cdots, 1)</td>
</tr>
<tr>
<td>JOS1-3</td>
<td>[32]</td>
<td>1000</td>
<td>2</td>
<td>(0, \cdots, 0)</td>
<td>(1, \cdots, 1)</td>
</tr>
<tr>
<td>MLF1</td>
<td>[30]</td>
<td>1</td>
<td>2</td>
<td>-10</td>
<td>13</td>
</tr>
<tr>
<td>MLF2</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-100, -100)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>TE1</td>
<td>[49]</td>
<td>2</td>
<td>2</td>
<td>(-1, -1)</td>
<td>(1, 1)</td>
</tr>
<tr>
<td>TE2</td>
<td>[49]</td>
<td>2</td>
<td>2</td>
<td>(-2, -2)</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>TE4</td>
<td>[49]</td>
<td>10</td>
<td>2</td>
<td>(-10, \cdots, -10)</td>
<td>(10, \cdots, 10)</td>
</tr>
<tr>
<td>TE6</td>
<td>[49]</td>
<td>2</td>
<td>2</td>
<td>(0, 0)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>TE7</td>
<td>[49]</td>
<td>3</td>
<td>3</td>
<td>(0, 0, 0)</td>
<td>(30, 30, 30)</td>
</tr>
<tr>
<td>SP1</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-100, -100)</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>SSFYY2</td>
<td>[30]</td>
<td>1</td>
<td>2</td>
<td>-100</td>
<td>(100, 100)</td>
</tr>
<tr>
<td>SK1</td>
<td>[30]</td>
<td>1</td>
<td>2</td>
<td>-100</td>
<td>100</td>
</tr>
<tr>
<td>SK2</td>
<td>[30]</td>
<td>4</td>
<td>2</td>
<td>(-10, -10, -10, -10)</td>
<td>(10, 10, 10, 10)</td>
</tr>
<tr>
<td>VU1</td>
<td>[30]</td>
<td>2</td>
<td>2</td>
<td>(-3, -3)</td>
<td>(3, 3)</td>
</tr>
</tbody>
</table>
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:

\[
\varphi_{p,s} = \frac{c_{p,s}}{\min_{s'} c_{p,s'}} : s' \in S.
\]

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

\[
\rho_s(\tau) = \frac{1}{|P|} \left| \left\{ p \in P : \varphi_{p,s} \leq \tau \right\} \right|.
\]

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>YWL</th>
<th>WHS</th>
<th>WLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOP1</td>
<td>4</td>
<td>1828</td>
<td>75</td>
</tr>
<tr>
<td>MOP2</td>
<td>1</td>
<td>3</td>
<td>0.0047</td>
</tr>
<tr>
<td>MOP3</td>
<td>8</td>
<td>298</td>
<td>129</td>
</tr>
<tr>
<td>DD1</td>
<td>11</td>
<td>579</td>
<td>143</td>
</tr>
<tr>
<td>Toi4</td>
<td>4</td>
<td>297</td>
<td>84</td>
</tr>
<tr>
<td>PNR</td>
<td>7</td>
<td>106</td>
<td>93</td>
</tr>
<tr>
<td>MMR1</td>
<td>10</td>
<td>173</td>
<td>127</td>
</tr>
<tr>
<td>AP3</td>
<td>3</td>
<td>330</td>
<td>37</td>
</tr>
<tr>
<td>Lov5</td>
<td>2</td>
<td>50</td>
<td>21</td>
</tr>
<tr>
<td>IKK1</td>
<td>3</td>
<td>234</td>
<td>51</td>
</tr>
<tr>
<td>TE8-1</td>
<td>7</td>
<td>681</td>
<td>165</td>
</tr>
<tr>
<td>TE8-2</td>
<td>16</td>
<td>1014</td>
<td>268</td>
</tr>
<tr>
<td>TE8-3</td>
<td>16</td>
<td>1148</td>
<td>287</td>
</tr>
<tr>
<td>MOP5</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MOP7</td>
<td>6</td>
<td>464</td>
<td>93</td>
</tr>
<tr>
<td>FDS-1</td>
<td>24</td>
<td>1098</td>
<td>301</td>
</tr>
<tr>
<td>TE8-1</td>
<td>7</td>
<td>681</td>
<td>165</td>
</tr>
<tr>
<td>TE8-2</td>
<td>16</td>
<td>1014</td>
<td>268</td>
</tr>
<tr>
<td>TE8-3</td>
<td>16</td>
<td>1148</td>
<td>287</td>
</tr>
<tr>
<td>Toi8</td>
<td>3</td>
<td>137</td>
<td>52</td>
</tr>
<tr>
<td>AP1</td>
<td>2</td>
<td>183</td>
<td>21</td>
</tr>
<tr>
<td>BK1</td>
<td>5</td>
<td>428</td>
<td>95</td>
</tr>
<tr>
<td>DGO1</td>
<td>2</td>
<td>74</td>
<td>27</td>
</tr>
<tr>
<td>FF1</td>
<td>14</td>
<td>231</td>
<td>145</td>
</tr>
<tr>
<td>JOS1-1</td>
<td>4</td>
<td>2211</td>
<td>81</td>
</tr>
<tr>
<td>JOS1-2</td>
<td>4</td>
<td>2211</td>
<td>81</td>
</tr>
<tr>
<td>JOS1-3</td>
<td>5</td>
<td>2947</td>
<td>107</td>
</tr>
<tr>
<td>MFL1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MLF2</td>
<td>5</td>
<td>280</td>
<td>77</td>
</tr>
<tr>
<td>TE1</td>
<td>4</td>
<td>383</td>
<td>59</td>
</tr>
<tr>
<td>TE2</td>
<td>4</td>
<td>147</td>
<td>65</td>
</tr>
<tr>
<td>TE4</td>
<td>8</td>
<td>494</td>
<td>120</td>
</tr>
<tr>
<td>TE6</td>
<td>4</td>
<td>1406</td>
<td>77</td>
</tr>
<tr>
<td>TE7</td>
<td>4</td>
<td>297</td>
<td>45</td>
</tr>
<tr>
<td>SP1</td>
<td>6</td>
<td>329</td>
<td>86</td>
</tr>
<tr>
<td>SSFY1</td>
<td>3</td>
<td>196</td>
<td>41</td>
</tr>
<tr>
<td>SK1</td>
<td>3</td>
<td>203</td>
<td>33</td>
</tr>
<tr>
<td>SK2</td>
<td>8</td>
<td>488</td>
<td>128</td>
</tr>
<tr>
<td>VU1</td>
<td>98</td>
<td>748</td>
<td>798</td>
</tr>
</tbody>
</table>
Table 3. Continuation of Table 2

<table>
<thead>
<tr>
<th>Problem</th>
<th>WLS* It Fe Ge</th>
<th>WLS* time</th>
<th>PRP It Fe Ge</th>
<th>PRP time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOP1</td>
<td>4 1830 75</td>
<td>0.0174</td>
<td>4 1830 75</td>
<td>0.0169</td>
</tr>
<tr>
<td>MOP2</td>
<td>1 3 3</td>
<td>0.0067</td>
<td>1 3 3</td>
<td>0.0062</td>
</tr>
<tr>
<td>MOP3</td>
<td>8 282 135</td>
<td>0.0378</td>
<td>8 292 132</td>
<td>0.0364</td>
</tr>
<tr>
<td>DD1</td>
<td>11 547 132</td>
<td>0.0485</td>
<td>11 547 136</td>
<td>0.0481</td>
</tr>
<tr>
<td>To4</td>
<td>4 301 87</td>
<td>0.0217</td>
<td>4 301 87</td>
<td>0.0208</td>
</tr>
<tr>
<td>PNR</td>
<td>7 106 93</td>
<td>0.0242</td>
<td>7 106 93</td>
<td>0.0233</td>
</tr>
<tr>
<td>MMR1</td>
<td>10 173 127</td>
<td>0.0433</td>
<td>10 173 127</td>
<td>0.0418</td>
</tr>
<tr>
<td>AP3</td>
<td>3 330 37</td>
<td>0.0489</td>
<td>3 330 37</td>
<td>0.0206</td>
</tr>
<tr>
<td>Lov5</td>
<td>4 301 87</td>
<td>0.0217</td>
<td>4 301 87</td>
<td>0.0208</td>
</tr>
<tr>
<td>PNR</td>
<td>7 106 93</td>
<td>0.0242</td>
<td>7 106 93</td>
<td>0.0233</td>
</tr>
<tr>
<td>MMR1</td>
<td>10 173 127</td>
<td>0.0433</td>
<td>10 173 127</td>
<td>0.0418</td>
</tr>
<tr>
<td>AP3</td>
<td>3 330 37</td>
<td>0.0489</td>
<td>3 330 37</td>
<td>0.0206</td>
</tr>
<tr>
<td>Lov5</td>
<td>4 301 87</td>
<td>0.0217</td>
<td>4 301 87</td>
<td>0.0208</td>
</tr>
<tr>
<td>PNR</td>
<td>7 106 93</td>
<td>0.0242</td>
<td>7 106 93</td>
<td>0.0233</td>
</tr>
<tr>
<td>MMR1</td>
<td>10 173 127</td>
<td>0.0433</td>
<td>10 173 127</td>
<td>0.0418</td>
</tr>
<tr>
<td>AP3</td>
<td>3 330 37</td>
<td>0.0489</td>
<td>3 330 37</td>
<td>0.0206</td>
</tr>
<tr>
<td>Lov5</td>
<td>4 301 87</td>
<td>0.0217</td>
<td>4 301 87</td>
<td>0.0208</td>
</tr>
<tr>
<td>PNR</td>
<td>7 106 93</td>
<td>0.0242</td>
<td>7 106 93</td>
<td>0.0233</td>
</tr>
<tr>
<td>MMR1</td>
<td>10 173 127</td>
<td>0.0433</td>
<td>10 173 127</td>
<td>0.0418</td>
</tr>
<tr>
<td>AP3</td>
<td>3 330 37</td>
<td>0.0489</td>
<td>3 330 37</td>
<td>0.0206</td>
</tr>
<tr>
<td>Lov5</td>
<td>4 301 87</td>
<td>0.0217</td>
<td>4 301 87</td>
<td>0.0208</td>
</tr>
<tr>
<td>PNR</td>
<td>7 106 93</td>
<td>0.0242</td>
<td>7 106 93</td>
<td>0.0233</td>
</tr>
<tr>
<td>MMR1</td>
<td>10 173 127</td>
<td>0.0433</td>
<td>10 173 127</td>
<td>0.0418</td>
</tr>
<tr>
<td>AP3</td>
<td>3 330 37</td>
<td>0.0489</td>
<td>3 330 37</td>
<td>0.0206</td>
</tr>
<tr>
<td>Lov5</td>
<td>4 301 87</td>
<td>0.0217</td>
<td>4 301 87</td>
<td>0.0208</td>
</tr>
<tr>
<td>PNR</td>
<td>7 106 93</td>
<td>0.0242</td>
<td>7 106 93</td>
<td>0.0233</td>
</tr>
</tbody>
</table>

Figures 1-3 display $\rho_s(\tau)$ on the y-axis against a logarithmic scale (base 2) of $\tau$ on the x-axis, we use $\tau$ 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, WLS, WHS, 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.

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.
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


