Chapter 4

Nonlinear conjugate gradient methods for unconstrained set optimization problems whose objective functions have finite cardinality

4.1 Introduction

In this chapter, we propose a nonlinear conjugate gradient method and its two variations, namely Fletcher-Reeves and conjugate descent, for unconstrained set optimization problems whose objective functions have finite cardinality. The conjugate gradient method is one of the most popular methods in optimization. This method was originally proposed by Fletcher and Reeves [65] in 1964. Further, variants of conjugate gradient methods, including conjugate descent, were proposed by researchers [53, 64, 164, 165]. In [3], Al-Baali proved that the conjugate direction calculated by the method proposed in [65] satisfies a sufficient descent condition. Using this sufficient descent condition, Al-Baali proved the global convergence of the Fletcher-Reeves method [65] with the help of Zoutendijik condition [205]. Thereafter, global convergence of the variants of the conjugate gradient method has been proved under the sufficient descent condition with the help of the Zoutendijk condition by many researchers, for instance, see [1,53,79,91,202]. We recommend that interested readers see [6] for a rigorously detailed description of the developments on conjugate gradient methods for optimization problems with real-valued objective functions. In [162], the nonlinear conjugate gradient method and its variants for vector optimization problems have been introduced. In [162], Wolfe line search and Zoutendijk-type conditions for vector optimization problems have been proposed for the first time. In this chapter, we also use the idea given in [162] to extend the Wolfe line search procedure and Zoutendijk-type conditions for the considered set optimization problems. The work in this chapter introduces a nonlinear conjugate gradient method and its two variants (Fletcher–Reeves and conjugate descent) along with Wolfe line search for set optimization problems with the objective function as a set-valued mapping of finite cardinality.

4.2 Motivation

In set optimization, there are different algorithms in the literature: derivative-free algorithms [116, 118, 132], sorting-type algorithms [84, 85, 133, 134], and scalarization based algorithms [57, 59, 109, 110, 124, 174]. The detailed discussion on these algorithms has been given in subsection 1.7.5. Beyond these methods, Bouza et al. [32] introduced a steepest descent method for unconstrained set optimization problems where the set-valued mappings have finite cardinality. In their work, they also pointed out certain limitations in the aforementioned three categories when applied to the set optimization problems discussed in [32]. Inspired by their findings, in this chapter, we propose a nonlinear conjugate gradient method, along with two variants—Fletcher–Reeves and conjugate descent—for solving set optimization problems with objective functions of finite cardinality.

4.3 Contributions

The major contributions in this chapter are as follows:

- Necessary optimality conditions for stationary point for SOP (1.3) using Drummond-Svaiter function are given.
- A nonlinear conjugate gradient method and its two variants, Fletcher-Reeves and conjugate descent, for SOP (1.3) are developed.
- Zoutendijk type and Wolfe line search type conditions are extended.
- The well-definedness of the proposed methods is given.
- Global convergence of the proposed methods is proved with and without regularity assumption.
- A numerical description of methods is given along with a comparison with the existing steepest descent for SOP (1.3).

4.4 Optimality conditions

In this section, some optimality conditions are proposed for SOP (1.3). First, we start this section by mentioning some important index-related set-valued mappings sets given in [32].

Definition 4.1 (Index-related set valued mappings [32]).

(i) *The active index of minimal elements associated with objective set-valued mapping F of SOP* (1.3) *is $I : \mathbb{R}^n \rightrightarrows [p]$, which is given by*

$$I(x) = \{i \in [p] : f^i(x) \in \mathrm{Min}(F(x), K)\}.$$

(ii) *The active index of weakly minimal elements associated with objective set-valued mapping F of SOP (1.3) is $I_w : \mathbb{R}^n \rightrightarrows [p]$, which is given by*

$$I_w(x) = \{i \in [p] : f^i(x) \in \mathrm{WMin}(F(x), K)\}.$$

(iii) *For a vector $u \in \mathbb{R}^m$, the set-valued mapping $I_u : \mathbb{R}^n \rightrightarrows [p]$ is defined by*

$$I_u(x) = \{i \in I(x) : f^i(x) = u\}.$$

It is to notice that for any $x \in \mathbb{R}^n$, $I_u(x) = \emptyset$ for $u \notin \mathrm{Min}(F(x), K)$; $I_u(x) \cap I_v(x) = \emptyset$ for any $u \neq v \in \mathbb{R}^m$; $I(x) = \bigcup_{u \in \mathrm{Min}(F(x),K)} I_u(x)$.

Definition 4.2 (Cardinality of a set of minimal elements [32]). *The map $\omega : \mathbb{R}^n \to \mathbb{R}$ which is defined by*

$$\omega(x) = |\mathrm{Min}(F(x), K)|$$

is called the cardinality of the set of minimal elements of F at x.

For a given $\bar{x} \in \mathbb{R}^n$, we use the notation $\bar{\omega}$ throughout to mean the value of $\omega(\bar{x})$.

Next, we give the definition of the partition set of a point $x \in \mathbb{R}^n$ in order to systematically identify weakly minimal points of the SOP (1.3).

Definition 4.3 (Partition set at a point [32]). *Let at a given $x \in \mathbb{R}^n$, an enumeration of the set $\mathrm{Min}(F(x), K)$ be $\{u_1^x, u_2^x, \ldots, u_{\omega(x)}^x\}$. For the point x, the partition set is defined by*

$$P_x = \prod_{j=1}^{\omega(x)} I_{u_j^x}(x).$$

In this chapter, for an iterative point $x_k \in \mathbb{R}^n$, a generic element of the partition set P_{x_k} is denoted by a^i, and the components of a^i are denoted by a^i_j for all

$j \in [\omega(x_k)]$, where $i = 1, 2, 3, \ldots$. Precisely, if $|P_{x_k}| = p_k$ and $\text{Min}(F(x_k), K) = \left\{u_1^{x_k}, u_2^{x_k}, \ldots, u_{w(x_k)}^{x_k}\right\}$, then

$$P_{x_k} = \{a^1, a^2, \ldots, a^{p_k}\},$$

where for each $i \in [p_k]$,

$$a^i = \left(a_1^i, a_2^i, \ldots, a_{w(x_k)}^i\right),\ a_j^i \in I_{u_j^{x_k}},\ j \in [w(x_k)].$$

Next, we present a lemma that connects SOP (1.3) to a family of vector optimization problems at a weakly minimal point. This family of vector optimization problems locally represents SOP (1.3) around the point, which is exploited later to find the weakly minimal points of SOP (1.3).

Lemma 4.1 (See [32]). *Let $\bar{x} \in \mathbb{R}^n$. Consider the partition set $P_{\bar{x}}$ of $\bar{x}$. Define a vector-valued function $\tilde{f} : \mathbb{R}^n \to \prod_{j=1}^{\bar{\omega}} \mathbb{R}^m$, which is given by*

$$\tilde{f}^a(x) = (f^{a_1}(x), f^{a_2}(x), \ldots, f^{a_{\bar{\omega}}}(x))^\top \text{ for every } a = (a_1, a_2, \ldots, a_{\bar{\omega}}) \in P_{\bar{x}}.$$

Let $\tilde{K} \in \mathscr{P}(\mathbb{R}^{m\bar{\omega}})$ be the cone such that $\tilde{K} = \prod_{j=1}^{\bar{\omega}} K$. Consider the unconstrained vector optimization problem (VOP) with respect to the ordering cone $\tilde{K}$:

$$\begin{aligned} &\text{minimize} \quad \tilde{f}^a(x) \\ &\text{subject to} \quad x \in \mathbb{R}^n. \end{aligned} \tag{4.1}$$

Then, $\bar{x}$ is a local weakly minimal solution of SOP (1.3) if and only if $\bar{x}$ is a local weakly minimal solution of VOP (4.1) for every $a \in P_{\bar{x}}$.

Below, we present a definition of a stationary point. Later, we connect the stationary point with a weakly minimal point.

Definition 4.4 (Stationary Point). *A point $\bar{x} \in \mathbb{R}^n$ is called a stationary point of SOP (1.3) if for any $a = (a_1, a_2, \ldots, a_{\bar{\omega}}) \in P_{\bar{x}}$ and $d \in \mathbb{R}^n$, there exists $j \in [\bar{\omega}]$ such that* $\mathrm{J} f^{a_j}(\bar{x})d \notin -\operatorname{int}(K)$.

It is a known fact that if $\bar{x}$ is a weakly minimal point of the SOP (1.3), then $\bar{x}$ is a stationary point of (1.3) (see [32, Theorem 3.1]). Thus, a necessary condition for weak minimality of $\bar{x}$ is its stationarity. In this chapter, although ideally, we aim to identify weakly minimal points of the problem (1.3), but we end up finding the stationary points of (1.3) mainly due to two prime reasons:

(i) identification of a computationally viable algebraic equation or condition for weakly minimal points is difficult, and

(ii) there is an easily implementable (see Proposition 4.2 and Remark 4.2) condition for stationarity.

However, we note that if the objective function F is a convex set-valued function, then any stationary point of (1.3) is a weakly minimal point of (1.3), and thus for the convex-case, the proposed method successfully identifies weakly minimal points of (1.3).

Next, we introduce some functions that will be useful to propose the conjugate gradient method and its convergence analysis.

For any given $x \in \mathbb{R}^n$, we define a function $\vartheta_x : P_x \times \mathbb{R}^n \to \mathbb{R}$ by

$$\vartheta_x(a, v) = \max_{j \in [\omega(x)]} \left\{ \varphi \left(\mathrm{J} f^{a_j}(x) v \right) \right\} + \tfrac{1}{2}\|v\|^2. \tag{4.2}$$

Note that for every $x \in \mathbb{R}^n$ and $a \in P_x$, the function $\vartheta_x(a, \cdot)$ is strongly convex in $\mathbb{R}^n$ because φ is sublinear. Therefore, the function $\vartheta_x(a, \cdot)$ attains its global minimum value at a unique point.

Observe that for all $x \in \mathbb{R}^n$ and $a \in P_x$, we have

$$\min_{v \in \mathbb{R}^n} \vartheta_x(a, v) \leq \vartheta_x(a, v') \text{ for all } v' \in \mathbb{R}^n.$$

Thus, in particular, taking $v' = 0$, we get for any $a \in P_x$ that

$$\min_{v \in \mathbb{R}^n} \vartheta_x(a, v) \leq \vartheta_x(a, 0) = 0. \tag{4.3}$$

Also, note that the partition set P_x has finitely many elements. Therefore, ϑ_x attains its minimum over the set $P_x \times \mathbb{R}^n$. Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be the function

$$\phi(x) = \min_{(a,v) \in P_x \times \mathbb{R}^n} \vartheta_x(a, v). \tag{4.4}$$

Then, by (4.3), for all $x \in \mathbb{R}^n$, we have

$$\phi(x) \leq 0. \tag{4.5}$$

Moreover, if for $(a', v') \in P_x \times \mathbb{R}^n$ we have $\phi(x) = \vartheta_x(a', v')$, then

$$\phi(x) = 0 \text{ if and only if } v' = 0. \tag{4.6}$$

If $\bar{x} \in \mathbb{R}^n$ is a weakly minimal point of the SOP (1.3), then due to its stationarity, we get from Definition 4.4 that

$$\begin{aligned}
& \forall a \in P_{\bar{x}} \text{ and } v \in \mathbb{R}^n \ : \ \max_{j \in [\bar{\omega}]} \ \varphi(\mathrm{J}\, f^{a_j}(\bar{x})v) \geq 0 \\
\implies \ & \forall a \in P_{\bar{x}} \text{ and } v \in \mathbb{R}^n \ : \ \vartheta_{\bar{x}}(a, v) \geq 0 \\
\implies \ & \min_{(a,v) \in P_{\bar{x}} \times \mathbb{R}^n} \ \vartheta_{\bar{x}}(a, v) \geq 0 \\
\implies \ & \phi(\bar{x}) \geq 0
\end{aligned}$$

$$\stackrel{(4.5)}{\Longrightarrow} \quad \phi(\bar{x}) = 0.$$

So, at a weakly minimal point $\bar{x}$ of (1.3), if $(\bar{a}, \bar{v}) \in P_{\bar{x}} \times \mathbb{R}^n$ be such that $\phi(\bar{x}) = \vartheta_{\bar{x}}(\bar{a}, \bar{v})$, then

$$\vartheta_{\bar{x}}(\bar{a}, \bar{v}) = 0,$$

which implies that $\bar{v} = 0$. Indeed, since if $\bar{v} \neq 0$, then due to the stationarity of $\bar{x}$, we have $\vartheta_{\bar{x}}(\bar{a}, \bar{v}) > 0$.

Accumulating all, we obtain the following result.

Proposition 4.1 (Necessary condition for weakly minimal points).
Let $\hat{x}$ be a weakly minimal point of SOP (1.3) and $(\hat{a}, \hat{v}) \in P_{\hat{x}} \times \mathbb{R}^n$ be such that $\phi(\hat{x}) = \vartheta_{\hat{x}}(\hat{a}, \hat{v})$, where ϕ and $\vartheta_{\hat{x}}$ are as defined in (4.4) and (4.2), respectively. Then, $\hat{v} = 0$.

In the next section, we present a nonlinear conjugate gradient algorithm for SOP (1.3).

4.5 Nonlinear conjugate gradient method and its convergence

In this section, we propose a nonlinear conjugate gradient method for the solution approach for SOP (1.3). In the algorithm, we start by choosing an initial point. If it does not satisfy the necessary condition stated in Proposition 4.1 for a weakly minimal point, we update it as follows. First, we find (a^k, v_k) as in Step 2, and then using suitable conjugate parameter β_k, we find the conjugate direction. We use the Wolfe line search procedure to find the suitable step length. After that, we update the iteration in the calculated direction.

Algorithm 1 Nonlinear conjugate gradient algorithm for SOP (1.3) using Wolfe line search

Step 0: Initialization

Choose an initial point $x_0 \in \mathbb{R}^n$, constant parameters ρ, σ such that $0 < \rho < \sigma < 1$ and $e \in K$ such that $0 < \langle w, e \rangle \leq 1$ for all $w \in C$. Set the iteration counter $k = 0$. Provide a termination scalar $\epsilon > 0$.

Step 1: Computation of the partition set at the k-th iteration point

Find $M_k = \text{Min}(F(x_k), K)$ and $\omega_k = |M_k|$. Compute $P_k = P_{x_k}$ as given in Definition 4.3.

Step 2: Calculate the steepest descent direction

Find $(a^k, v_k) = \underset{(a,v) \in P_{x_k} \times \mathbb{R}^n}{\text{argmin}} \vartheta_{x_k}(a, v)$, where $\vartheta_{x_k} : P_{x_k} \times \mathbb{R}^n \to \mathbb{R}$ is given by

$$\vartheta_{x_k}(a, v) = \max_{j \in [\omega_k]} \left\{ \varphi \left(\mathrm{J}\, f^{a_j}(x_k) v \right) \right\} + \tfrac{1}{2} \|v\|^2.$$

Step 3: Stopping criterion

If $\|v_k\| < \epsilon$, then stop. Otherwise, go to **Step 4**.

Step 4: Calculate the conjugate direction

Choose

$$d_k = \begin{cases} v_k, & \text{if } k = 0 \\ v_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{4.7}$$

where β_k is an algorithmic parameter.

Step 5: Finding a step length

Find a positive step length α_k such that

$$\left. \begin{aligned} & f^{a_j^k}(x_k + \alpha_k d_k) \preceq f^{a_j^k}(x_k) + \rho \alpha_k \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) e \text{ for all } j \in [\omega_k] \\ \text{and } & \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k + \alpha_k d_k) d_k) \geq \sigma \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k), \end{aligned} \right\} \tag{4.8}$$

or,

$$\left. \begin{aligned} & f^{a_j^k}(x_k + \alpha_k d_k) \preceq f^{a_j^k}(x_k) + \rho \alpha_k \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) e \text{ for all } j \in [\omega_k] \\ \text{and } & \left| \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k + \alpha_k d_k) d_k) \right| \leq \sigma \left| \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) \right|. \end{aligned} \right\} \tag{4.9}$$

Step 6: Update

Update $x_{k+1} \leftarrow x_k + \alpha_k d_k$, $k \leftarrow k + 1$ and go to **Step 1**.

Remark 4.1 *If we run Algorithm 1 for* $p = 1$*, i.e.,* $F(x) = \{f^1(x)\}$ *in SOP (1.3), then there is no need of Step* 1*, and in this case Step* 2 *of Algorithm 1 reduces to finding* v_k *such that*

$$v_k = \operatorname*{argmin}_{v \in \mathbb{R}^n} \theta_{x_k}(v),$$

where $\theta_{x_k} : \mathbb{R}^n \to \mathbb{R}$ *is*

$$\theta_{x_k}(v) = \varphi(\mathrm{J}\, f^1(x_k)v) + \tfrac{1}{2}\|v\|^2.$$

In this case, the Wolfe line searches for finding the positive step length in Step 5 *reduce to*

$$\left.\begin{aligned} f^1(x_k + \alpha_k d_k) \preceq f^1(x_k) + \rho\alpha_k\varphi(\mathrm{J}\, f^1(x_k)d_k)e \\ \text{and } \varphi(\mathrm{J}\, f^1(x_k + \alpha_k d_k)d_k) \geq \sigma\varphi(\mathrm{J}\, f^1(x_k)d_k), \qquad \end{aligned}\right\}$$

or,

$$\left.\begin{aligned} f^1(x_k + \alpha_k d_k) \preceq f^1(x_k) + \rho\alpha_k\varphi(\mathrm{J}\, f^1(x_k)d_k)e \\ \text{and } \left|\varphi(\mathrm{J}\, f^1(x_k + \alpha_k d_k)d_k)\right| \leq \sigma\left|\varphi(\mathrm{J}\, f^1(x_k)d_k)\right|. \end{aligned}\right\}$$

Subsequently, Step 3*, Step* 4*, and Step* 6 *remain unchanged. Thus, it can be easily noticed that when* $p = 1$*, Algorithm 1 reduces to the nonlinear conjugate gradient method given in [162] for vector optimization problems. However, the method in [162] cannot be straightly extended for set-optimization problems due to the prime fact that while generating the sequence of iterates* $\{x_k\}$ *for the set optimization problem (1.3), the vector optimization problem (associated to* a^k *in Step* 2*) that we solve to generate* x_k *is completely different than that for finding* x_{k+1}*.*

4.5.1 Convergence analysis

We start this section with the *well-definedness* of Algorithm 1. Further, we prove the convergence of Algorithm 1. For the convergence analysis, we first aim to derive the Zoutendijk-type condition for SOP (1.3). With the help of the Zoutendijk-type condition, we establish the convergence of the method.

The well-definedness of Algorithm 1 depends on the following points:

(i) Existence of (a^k, v_k) in Step 2, which is assured as the set P_{x_k} is finite and $\vartheta_{x_k}(a, \cdot)$ is strongly convex in $\mathbb{R}^n$ for each $a \in P_{x_k}$.

(ii) Existence of a β_k in Step 4 for which d_k becomes a descent direction for F, which we prove below in Remark 4.4.

(iii) Existence of α_k in Step 5, which we prove in Proposition 4.3.

The next result characterizes stationary points of SOP (1.3) in terms of the functions ϑ_x and ϕ as defined in (4.2) and (4.4), respectively. Basically, it is proved that if Algorithm 1 stops in Step 3, a stationary point of SOP (1.3) is obtained that satisfies a necessary condition for weakly minimal points.

Proposition 4.2 *Let $(\bar{a}, \bar{v}) \in P_x \times \mathbb{R}^n$ be a point such that $\phi(x) = \vartheta_x(\bar{a}, \bar{v})$, where $x \in \mathbb{R}^n$. Then, the following statements are equivalent:*

(i) *The point x is not a stationary point of SOP (1.3),*

(ii) $\phi(x) < 0$, *and*

(iii) $\bar{v} \neq 0$.

Proof: (i)→(ii). Suppose that $x \in \mathbb{R}^n$ is a nonstationary point. Then, by Definition 4.4, there exists an element $a = (a_1, a_2, \ldots, a_{\omega(x)}) \in P_x$ and $v \in \mathbb{R}^n$ for which

$$\mathrm{J}\, f^{a_j}(x)v \in -\operatorname{int}(K) \text{ for all } j \in [\omega(x)].$$

Therefore, $\varphi(\mathrm{J}\, f^{a_j}(x)v) < 0$ for all $j \in [\omega(x)]$. By the definition of ϑ_x, for $\eta \in [0, 1]$,

$$\begin{aligned}\phi(x) = \vartheta_x(\bar{a}, \bar{v}) \leq \vartheta_x(a, \eta v) &= \max_{j\in[\omega(x)]}\{\varphi(\mathrm{J}\, f^{a_j}(x)\eta v)\} + \tfrac{1}{2}\|\eta v\|^2 \\ &= \eta\left(\max_{j\in[\omega(x)]}\{\varphi(\mathrm{J}\, f^{a_j}(x)v)\} + \tfrac{1}{2}\eta\|v\|^2\right) \text{ using the definition of } \varphi. \end{aligned} \tag{4.10}$$

Now take any η such that $0 < \eta < \frac{1}{\|v\|^2}\left(-2\max_{j\in[\omega(x)]}\{\varphi(\mathrm{J}\, f^{a_j}(x)v)\}\right)$. Then, from (4.10), we get

$$\phi(x) < \eta\left(\max_{j\in[\omega(x)]}\{\varphi(\mathrm{J}\, f^{a_j}(x)v)\} - \max_{j\in[\omega(x)]}\{\varphi(\mathrm{J}\, f^{a_j}(x)v)\}\right) = 0.$$

Hence, the desired result is obtained.

(ii)→(iii). On the contrary, let $\bar{v} = 0$. Then, by (4.6), we have $\phi(x) = 0$ which is contradictory to $\phi(x) < 0$. Hence, $\bar{v} \neq 0$.

(iii)→(i). Let $\bar{v} \neq 0$. Then, $\max_{j\in[\omega(x)]}\{\varphi(\mathrm{J}\, f^{\bar{a}_j}(x)\bar{v})\} < \vartheta_x(\bar{a}, \bar{v}) = \phi(x)$, i.e.,

$$\mathrm{J}\, f^{\bar{a}_j}(x)\bar{v} \in (-\operatorname{int}(K)) \text{ for all } j \in [\omega(x)] \text{ using (1.2).}$$

Hence, by Definition 4.4, x is not a stationary point of SOP (1.3). □

Remark 4.2 *Proposition 4.2 has established an equivalence between (i), (ii), and (iii). So, by using (4.5), (4.6), and Theorem 4.2, we obtain that a point $\bar{x} \in \mathbb{R}^n$ is a stationary point of SOP (1.3) if and only if $\phi(\bar{x}) = 0$ or $\bar{v} = 0$.*

From now on, let us assume that x_k, generated by Algorithm 1, is nonstationary point for all $k \geq 0$, i.e., $v_k \neq 0$ for all $k \geq 0$.

Remark 4.3 *At x_k, by Proposition 4.2, we have $v_k \neq 0$ and $\phi(x_k) < 0$. Since $\phi(x_k) =$*

$\vartheta_{x_k}(a^k, v_k)$, *therefore* $\vartheta_{x_k}(a^k, v_k) < 0$. *This implies*

$$\begin{aligned} &\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) + \tfrac{1}{2}\|v_k\|^2 < 0 \\ \Longrightarrow\ &\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) < -\tfrac{1}{2}\|v_k\|^2 < 0 \qquad (4.11) \\ \Longrightarrow\ &\varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) < 0 \textit{ for all } j \in [\omega_k] \qquad (4.12) \\ \Longrightarrow\ &\mathrm{J}\, f^{a_j^k}(x_k)v_k \in (-\operatorname{int}(K)) \textit{ for all } j \in [\omega_k] \textit{ by } (1.2). \end{aligned}$$

Analogous to scalar (see (4.2) in [79]) and vector optimization cases (see (14) in [162]), we say that d_k (the conjugate direction) satisfies the sufficient descent condition for all $k \geq 0$ if there exists $c > 0$ such that

$$\left(\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\right) \leq c\left(\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right) \text{ for all } k \geq 0, \qquad (4.13)$$

which implies that $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k) < 0$ as $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) < 0$ for all $k \geq 0$. Using (ii) of Proposition 4.3 in [32], we say that d_k is a *descent direction* for F at x_k if $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k) < 0$. Therefore, if d_k satisfies the *sufficient descent condition*, then d_k is a descent direction for F at x_k for all $k \geq 0$.

The next lemma will be useful for further analysis. It is based on the sufficient condition on β_k to ensure that d_k satisfies the sufficient descent condition.

Lemma 4.2 *Let Algorithm 1 be executed. Suppose that the sequence* $\{\beta_k\}$ *is defined as*

$$\beta_k \in \begin{cases} [0, \infty), & \textit{if } \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) \leq 0 \\ \left[0, -\mu \dfrac{\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)}{\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1})}\right), & \textit{if } \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) > 0, \end{cases} \qquad (4.14)$$

where $\mu \in [0,1)$. *Then,*

$$\left(\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\right) \leq (1-\mu)\left(\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right) \tag{4.15}$$

for all $k \geq 0$.

Proof: Note that when $k = 0$, we have $d_k = v_k$. Therefore, using (4.12) and $\mu \in [0, 1)$, (4.15) holds for $k = 0$.

Now we show that (4.15) holds whenever (4.14) holds for $k \geq 1$. For $k \geq 1$, by the definition of d_k in Step 3, for each $j \in [\omega_k]$, we have

$$\begin{aligned} &\mathrm{J}\, f^{a_j^k}(x_k)d_k = \mathrm{J}\, f^{a_j^k}(x_k)v_k + \beta_k\, \mathrm{J}\, f^{a_j^k}(x_k)d_{k-1} \\ \implies &\langle \mathrm{J}\, f^{a_j^k}(x_k)d_k, w\rangle = \langle \mathrm{J}\, f^{a_j^k}(x_k)v_k, w\rangle + \beta_k\langle \mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}, w\rangle \text{ for all } w \in C. \end{aligned} \tag{4.16}$$

Let us consider first that $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) \leq 0$, i.e., $\beta_k \geq 0$. Thus, $\varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) \leq 0$ for all $j \in [\omega_k]$, which implies that $\langle \mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}, w\rangle \leq 0$ for all $j \in [\omega_k]$ and $w \in C$. Therefore, from (4.16), for all $j \in [\omega_k]$ and $w \in C$, we have

$$\begin{aligned} \langle \mathrm{J}\, f^{a_j^k}(x_k)d_k, w\rangle &\leq \langle \mathrm{J}\, f^{a_j^k}(x_k)v_k, w\rangle \\ &\leq \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) \text{ using the definition of } \varphi \\ &\leq (1-\mu)\varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) \text{ using (4.12) and } \mu \in [0,1) \\ &\leq (1-\mu)\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k). \end{aligned} \tag{4.17}$$

Since (4.17) holds for all $j \in [\omega_k]$ and $w \in C$, therefore (4.15) holds when $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) \leq 0$.

Now assume that $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) > 0$. Then, in view of (4.16), for all $j \in [\omega_k]$

and $w \in C$, we have

$$\begin{aligned} \langle \mathrm{J}\, f^{a_j^k}(x_k)d_k, w\rangle &\le \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) + \beta_k \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) \\ &\le (1-\mu) \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) \text{ using (4.14).} \end{aligned} \tag{4.18}$$

Since (4.18) holds for all $j \in [\omega_k]$ and $w \in C$, therefore (4.15) holds when $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) > 0$. Hence, the proof is complete. □

Remark 4.4 *It is easy to verify from Lemma 4.2 that if*

$$\beta_k \in \begin{cases} [0,\infty), & \text{if } \max_{j\in[\omega(x)]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) \le 0 \\ \left[0, -\dfrac{\max_{j\in[\omega(x)]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)}{\max_{j\in[\omega(x)]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1})}\right), & \text{if } \max_{j\in[\omega(x)]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}) > 0, \end{cases} \tag{4.19}$$

then d_k is a descent direction for F, at x_k, i.e., $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k) < 0$ for all $k \ge 0$.

Next, we prove the existence of an α_k in Step 5 of Algorithm 1.

Proposition 4.3 *Let $\{x_k\}$ be a sequence generated by Algorithm 1. Assume that there exists $\mathfrak{B} \in \mathbb{R}^m$ such that $\mathfrak{B} \preceq f^i(x_k + \alpha d_k)$ for all $i \in [p]$ and $\alpha > 0$, where d_k is a descent direction for F at x_k. If the generator C of K is finite, then there exist intervals of positive step sizes satisfying (4.8) and (4.9).*

Proof: For $w \in C$ and $j \in [\omega_k]$, define two functions $\mathfrak{I}_w^j$ and $\mathfrak{L}_w^j$ from $\mathbb{R}$ to $\mathbb{R}$ by

$$\mathfrak{I}_w^j(\alpha) = \langle w, f^{a_j^k}(x_k + \alpha d_k)\rangle$$

and

$$\mathfrak{L}_w^j(\alpha) = \langle w, f^{a_j^k}(x_k)\rangle + \alpha\rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k).$$

Note that $\mathfrak{I}_w^j(0) = \langle w, f^{a_j^k}(x_k)\rangle = \mathfrak{L}_w^j(0)$. Since we have assumed that there exists $\mathfrak{B} \in \mathbb{R}^m$ such that $\mathfrak{B} \preceq f^i(x_k + \alpha d_k)$ for all $i \in [p]$ and $\alpha > 0$, therefore $\mathfrak{I}_w^j$ is bounded below for all $\alpha > 0$ and $j \in [\omega_k]$. Since d_k is a descent direction for F at x_k, we have $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k) < 0$.

In view of Lemma 3.1 in [161], for each $j \in [\omega_k]$, the line $\mathfrak{L}_w^j$ is unbounded below and must intersect the graph of $\mathfrak{I}_w^j$ at least once for a positive α. Since $f^{a_j^k}$'s are differentiable functions for all $j \in [\omega_k]$, $\mathfrak{I}_w^j$ is continuously differentiable for all $j \in [\omega_k]$. Therefore, there exists a $T_w^j > 0$, for each $j \in [\omega_k]$, such that

$$\mathfrak{I}_w^j(T_w^j) = \mathfrak{L}_w^j(T_w^j) \text{ and } \langle w, f^{a_j^k}(x_k + \alpha^j d_k)\rangle < \langle w, f^{a_j^k}(x_k)\rangle + \alpha^j \rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k) \tag{4.20}$$

for all $\alpha^j \in (0, T_w^j)$. For each $j \in [\omega_k]$, let $w^j \in C$ be an element such that $T_{w^j}^j = \min\{T_w^j : w \in C\}$. Thus, from (4.20), for all $w \in C$ and $j \in [\omega_k]$, we have

$$\begin{aligned}\langle w, f^{a_j^k}(x_k + \alpha d_k)\rangle \leq\ & \langle w, f^{a_j^k}(x_k)\rangle + \alpha\rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\\ \leq\ & \langle w, f^{a_j^k}(x_k)\rangle + \alpha\rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\langle w, e\rangle\end{aligned}$$

for all $\alpha \in [0, \tilde{T}]$, where $\tilde{T} = \min\{T_{w^j}^j : j \in [\omega_k]\}$. Hence, for all $\alpha \in [0, \tilde{T}]$, we have

$$f^{a_j^k}(x_k + \alpha d_k) \preceq f^{a_j^k}(x_k) + \rho\alpha \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)e \text{ for all } j \in [\omega_k].$$

Now we prove the second inequalities of (4.8) and (4.9). For this, for all $j \in [\omega_k]$, we define a function $\xi^j : \mathbb{R} \to \mathbb{R}$ by

$$\xi^j(\alpha) = \mathfrak{I}_{\tilde{w}}^j(\alpha) - \mathfrak{I}_{\tilde{w}}^j(0) - \alpha\rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k),$$

where $\tilde{w} \in C$ such that $T_{\tilde{w}} = \tilde{T}$. Since $\tilde{T}$ is taken such that (4.20) holds for all $j \in [\omega_k]$, therefore $\xi^j(0) = \xi^j(\tilde{T}) = 0$ for all $j \in [\omega_k]$. Thus, by mean value theorem, there exists an $\tilde{\alpha} \in (0, \tilde{T})$ such that $\frac{d\xi^j}{d\alpha}\big|_{\alpha=\tilde{\alpha}} = 0$ for all $j \in [\omega_k]$. Hence, $\frac{d\mathfrak{I}_{\tilde{w}}^j}{d\alpha}\big|_{\alpha=\tilde{\alpha}} = \rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)$ for all $j \in [\omega_k]$. Since $\frac{d\mathfrak{I}_{\tilde{w}}^j(\tilde{\alpha})}{d\alpha} \leq \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k + \tilde{\alpha}d_k)d_k)$ for all

$j \in [\omega_k]$, therefore

$$\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k + \tilde{\alpha} d_k)d_k) \geq \rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k)d_k). \tag{4.21}$$

Define a function $\zeta : \mathbb{R} \to \mathbb{R}$ by

$$\zeta(\alpha) = \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k + \alpha d_k)d_k).$$

It is easy to see from Lemma 1.14 that ζ is continuous and from (4.21), we have $\zeta(\tilde{\alpha}) \geq \rho\zeta(0)$. Therefore, by intermediate value theorem, there exists $\hat{\alpha} \in (0, \tilde{\alpha}]$ such that $\zeta(\hat{\alpha}) = \rho\zeta(0)$, i.e., $\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k + \hat{\alpha} d_k)d_k) = \rho \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k)d_k)$, which implies, using $0 < \rho < \sigma$ and the fact that d_k is a descent direction for F at x_k,

$$\sigma \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k)d_k) < \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a^k_j}(x_k + \hat{\alpha} d_k)d_k) < 0. \tag{4.22}$$

Hence, in view of (4.21) and (4.22), there is a neighborhood of $\hat{\alpha}$ contained in $[0, \tilde{T}]$ for which second inequalities of (4.8) and (4.9) hold. Therefore, in this neighborhood, (4.8) and (4.9) hold, which completes the proof.

□

We provide a result below based on the first inequality of (4.8). This result is useful for further analysis.

Lemma 4.3 *Suppose $\{x_k\}$ is generated by Algorithm 1. Then, for any $k \geq 0$, there exists a $j^k \in [\omega_k]$ such that*

$$f^{a^k_{j^k}}(x_k + \alpha_k d_k) \preceq f^{a^0_{j^0}}(x_0) + \rho \sum_{\Lambda=0}^{k} \alpha_\Lambda \max_{j\in[\omega_\Lambda]} \varphi(\mathrm{J}\, f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)e.$$

Proof: At the initial point x_0, i.e., when $k = 0$, since the first inequality of (4.8) holds for all $j \in [\omega_0]$, therefore for $j^0 \in [\omega_0]$, we have

$$f^{a^0_{j^0}}(x_0 + \alpha_0 d_0) \preceq f^{a^0_{j^0}}(x_0) + \rho\alpha_0 \max_{j\in[\omega_0]} \varphi(\mathrm{J}\, f^{a^0_j}(x_0)d_0)e. \tag{4.23}$$

Note that at $x_1 = x_0 + \alpha_0 d_0$, i.e., when $k = 1$, there exists a $j^1 \in [\omega_1]$ such that

$$f^{a^1_{j^1}}(x_1) \preceq f^{a^0_{j^0}}(x_1). \tag{4.24}$$

From the first inequality of (4.8), we have

$$\begin{aligned} f^{a^1_{j^1}}(x_1 + \alpha_1 d_1) &\preceq f^{a^1_{j^1}}(x_1) + \alpha_1\rho \max_{j\in[\omega_1]} \varphi(\mathrm{J}\, f^{a^1_j}(x_1)d_1)e \\ &\overset{(4.23)\ \&\ (4.24)}{\preceq} f^{a^0_{j^0}}(x_0) + \rho\sum_{\Lambda=0}^{1} \alpha_\Lambda \max_{j\in[\omega_\Lambda]} \varphi(\mathrm{J}\, f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)e. \end{aligned} \tag{4.25}$$

At $x_2 = x_1 + \alpha_1 d_1$, i.e., when $k = 2$, there exists a $j^2 \in [\omega_2]$ such that

$$f^{a^2_{j^2}}(x_2) \preceq f^{a^1_{j^1}}(x_2). \tag{4.26}$$

Now, using (4.25), (4.26), and the first inequality of (4.8), we have

$$f^{a^2_{j^2}}(x_2 + \alpha_2 d_2) \preceq f^{a^0_{j^0}}(x_0) + \rho\sum_{\Lambda=0}^{2} \alpha_\Lambda \max_{j\in[\omega_\Lambda]} \varphi(\mathrm{J}\, f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)e.$$

By doing the same procedure, at $x_{k+1} = x_k + \alpha_k d_k$, we have a $j^k \in [\omega_k]$ such that

$$f^{a^k_{j^k}}(x_k + \alpha_k d_k) \preceq f^{a^0_{j^0}}(x_0) + \rho\sum_{\Lambda=0}^{k} \alpha_\Lambda \max_{j\in[\omega_\Lambda]} \varphi(\mathrm{J}\, f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)e,$$

which completes the proof. □

Next, we prove the Zoutendijk-type condition for the proposed method. To prove this, let us suppose that the following assumptions are satisfied.

Assumption 2 (i) *The cone K is finitely generated and there exists an open set $\mathcal{O}$ such that $\mathbb{S} = \{x \in \mathbb{R}^n : F(x) \preceq^l F(x_0)\} \subset \mathcal{O}$, where x_0 is the chosen initial point. Also, the Jacobian $\mathrm{J}\, f^i$ is Lipschitz continuous on $\mathcal{O}$ with constant $L_i > 0$ for all $i \in [p]$.*

(ii) *All monotonically nonincreasing sequences in $f^i(\mathbb{S})$ are bounded below for all $i \in [p]$.*

Proposition 4.4 *Let $\{x_k\}$ be a sequence generated by Algorithm 1. Suppose that Assumption 2 holds, d_k is a descent direction for F and a step length α_k is chosen such that it satisfies (4.8) at every iteration. Then,*

$$\sum_{k \geq 0} \frac{\left(\max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k)\right)^2}{\|d_k\|^2} < +\infty. \tag{4.27}$$

Proof: Using the second inequality of (4.8) and $0 < \sigma < 1$, we have

$$(\sigma - 1) \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) \leq \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k + \alpha_k d_k) d_k) - \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k). \tag{4.28}$$

Let j' and j'' in $[\omega_k]$ be such that $\max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k + \alpha_k d_k) d_k) = \varphi(\mathrm{J}\, f^{a_{j'}^k}(x_k + \alpha_k d_k) d_k)$ and $\max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) = \varphi(\mathrm{J}\, f^{a_{j''}^k}(x_k) d_k)$. Thus, from (4.28),

$$\begin{aligned}
& (\sigma - 1) \max_{j \in [\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) \\
\leq\ & \varphi(\mathrm{J}\, f^{a_{j'}^k}(x_k + \alpha_k d_k) d_k) - \varphi(\mathrm{J}\, f^{a_{j''}^k}(x_k) d_k) \\
\leq\ & \varphi(\mathrm{J}\, f^{a_{j'}^k}(x_k + \alpha_k d_k) d_k) - \varphi(\mathrm{J}\, f^{a_{j'}^k}(x_k) d_k) \\
\leq\ & \varphi(\mathrm{J}\, f^{a_{j'}^k}(x_k + \alpha_k d_k) d_k - \mathrm{J}\, f^{a_{j'}^k}(x_k) d_k) \text{ using (i) of Lemma 1.13} \\
=\ & \sup_{w \in C} \langle \mathrm{J}\, f^{a_{j'}^k}(x_k + \alpha_k d_k) d_k - \mathrm{J}\, f^{a_{j'}^k}(x_k) d_k, w \rangle
\end{aligned}$$

$$
\begin{aligned}
&\leq \sup_{w\in C}\|\mathrm{J}\,f^{a^k_{j'}}(x_k+\alpha_k d_k)d_k - \mathrm{J}\,f^{a^k_{j'}}(x_k)d_k\|\|w\|\\
&\leq \|\mathrm{J}\,f^{a^k_{j'}}(x_k+\alpha_k d_k) - \mathrm{J}\,f^{a^k_{j'}}(x_k)\|\|d_k\| \text{ as } \|w\|=1 \text{ for all } w\in C\\
&\leq L_{a^k_{j'}}\alpha_k\|d_k\|^2 \text{ from (i) of Assumption 2,} \qquad (4.29)
\end{aligned}
$$

where $L_{a^k_{j'}}$ is a Lipschitz constant of $\mathrm{J}\,f^{a^k_{j'}}$. Thus, using (4.29), $\sigma\in(0,1)$, and $\varphi(\mathrm{J}\,f^{a^k_j}(x_k)d_k)<0$ for all $j\in[\omega_k]$, we have

$$
\frac{\left(\max_{j\in[\omega_k]}\varphi(\mathrm{J}\,f^{a^k_j}(x_k)d_k)\right)^2}{\|d_k\|^2} \leq L_{a^k_{j'}}\alpha_k\frac{\max_{j\in[\omega_k]}\varphi(\mathrm{J}\,f^{a^k_j}(x_k)d_k}{(\sigma-1)}. \qquad (4.30)
$$

By Lemma 4.3, there exists a $j^k\in[\omega_k]$, for each $k\geq 0$, such that

$$
f^{a^k_{j^k}}(x_{k+1}) - f^{a^0_{j^0}}(x_0) \preceq \rho\sum_{\Lambda=0}^{k}\alpha_\Lambda \max_{j\in[\omega_\Lambda]}\varphi(\mathrm{J}\,f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)e.
$$

Therefore, using (ii) of Assumption 2, there exists an $\mathcal{A}^{a^k_{j^k}}\in\mathbb{R}^m$ such that

$$
\mathcal{A}^{a^k_{j^k}} - f^{a^0_{j^0}}(x_0) \preceq \rho\sum_{\Lambda=0}^{k}\alpha_\Lambda \max_{j\in[\omega_\Lambda]}\varphi(\mathrm{J}\,f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)e
$$

for all $k\geq 0$. This implies, for all $k\geq 0$ and $w\in C$, that

$$
\langle\mathcal{A}^{a^k_{j^k}} - f^{a^0_{j^0}}(x_0), w\rangle \leq \rho\langle e,w\rangle\sum_{\Lambda=0}^{k}\alpha_\Lambda \max_{j\in[\omega_\Lambda]}\varphi(\mathrm{J}\,f^{a^\Lambda_j}(x_\Lambda)d_\Lambda). \qquad (4.31)
$$

Since $a^k_{j^k}\in[p]$ for all k and p is finite, the set value of $\min_{a^k_{j^k}\in[p],w\in C}\{\langle\mathcal{A}^{a^k_{j^k}} - f^{a^0_{j^0}}(x_0), w\rangle\}$ is a constant, say τ, for all $k\geq 0$. Thus, in view of (4.31), for all $k\geq 0$, we get

$$
\begin{aligned}
&\tau \leq \rho\langle e,w\rangle\sum_{\Lambda=0}^{k}\alpha_\Lambda \max_{j\in[\omega_\Lambda]}\varphi(\mathrm{J}\,f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)\\
\implies &\frac{\tau}{\sigma-1} \geq \rho\langle e,w\rangle\sum_{\Lambda=0}^{k}\alpha_\Lambda\frac{\max_{j\in[\omega_\Lambda]}\varphi(\mathrm{J}\,f^{a^\Lambda_j}(x_\Lambda)d_\Lambda)}{\sigma-1},
\end{aligned}
$$

which gives that

$$\sum_{k\geq 0}\alpha_k \frac{\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)}{\sigma-1} < +\infty,$$

and hence using (4.30), we get the desired result. □

Remark 4.5 *If we take descent direction $d_k = v_k$ for all $k \geq 0$ and assume that the step length α_k satisfies (4.8) for all $k \geq 0$, then Algorithm 1 converges to a stationary point in the sense that $\lim\limits_{k\to\infty}\|v_k\| = 0$. The reason is as follows. Suppose that $v_k \neq 0$ for all $k \geq 0$. From (4.11), we then have*

$$\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) < -\tfrac{1}{2}\|v_k\|^2 < 0,$$

which implies, using Proposition 4.4, that

$$\sum_{k\geq 0}\frac{1}{4}\|v_k\|^2 \leq \sum_{k\geq 0}\frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2}{\|v_k\|^2} = \sum_{k\geq 0}\frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\right)^2}{\|d_k\|^2} < +\infty.$$

Thus, we have $\lim\limits_{k\to\infty}\|v_k\| = 0$. From this analysis, it can be concluded that the method of the steepest descent for SOP (1.3) converges to a stationary point using the scalarizing function φ (instead of the Gerstewtiz function that has been used in [32]) without taking the regularity assumption provided it uses standard Wolfe line search (4.8).

The following theorem is the main convergence result of the proposed Algorithm 1.

Theorem 4.1 *Let $\{x_k\}$ be a sequence generated by Algorithm 1. Suppose that Assumption 2 holds and*

$$\sum_{k\geq 0}\frac{1}{\|d_k\|^2} = +\infty. \tag{4.32}$$

If d_k satisfies the sufficient descent condition (4.13) and α_k is chosen such that it satisfies the standard Wolfe condition (4.8), then $\liminf\limits_{k\to\infty}\|v_k\| = 0$.

Proof: We prove the theorem by contradiction. Suppose that there exists a positive real number $\tilde{t}$ such that $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$. From Step 2, at x_k, we have $(a^k, v_k) \in P_{x_k} \times \mathbb{R}^n$ such that $\phi(x_k) = \vartheta_{x_k}(a^k, v_k)$. Then, by (4.13) and (4.12),

$$\begin{aligned} \frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\right)^2}{\|d_k\|^2} &\geq c^2 \frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2}{\|d_k\|^2} \\ &\overset{(4.11)}{\geq} c^2 \frac{\|v_k\|^4}{4\|d_k\|^2} \\ &\geq \frac{c^2\tilde{t}^4}{4\|d_k\|^2}. \end{aligned} \tag{4.33}$$

Note that if (4.13) holds and α_k is chosen such that it satisfies the standard Wolfe condition (4.8), then by Proposition 4.4, we get

$$\begin{aligned} &\sum_{k\geq 0} \frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\right)^2}{\|d_k\|^2} < +\infty \\ \overset{(4.33)}{\Longrightarrow}\, &c^2\tilde{t}^4 \sum_{k\geq 0} \frac{1}{4\|d_k\|^2} < +\infty, \end{aligned}$$

which contradicts (4.32). Hence, our assumption, $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$, is wrong. Therefore, we have $\liminf\limits_{k\to\infty}\|v_k\| = 0$. □

Next, we prove the convergence of Algorithm 1 when α_k satisfies the strong Wolfe conditions (4.9). For this, we use the concept of regular point given in [32], which is as follows.

Definition 4.5 (Regular Point [32]). *A point $\bar{x} \in \mathbb{R}^n$ is a regular point of F if the following conditions are satisfied:*

(i) $\mathrm{Min}(F(\bar{x}), K) = \mathrm{WMin}(F(\bar{x}), K)$,

(ii) *the value of ω in a neighborhood of $\bar{x}$ is constant.*

A useful property of a regular point of a set-valued mapping is described in the next lemma.

Lemma 4.4 (See [32]). *Let $\bar{x} \in \mathbb{R}^n$ be a regular point of F. Then, there exists a neighborhood U of $\bar{x}$ such that the following properties hold for every $x \in U$: $\omega(x) = \bar{\omega}$ and $P_x \subseteq P_{\bar{x}}$.*

Now we give the convergence theorem when step length is chosen with strong Wolfe conditions (4.9).

Theorem 4.2 *Let $\{x_k\}$ be a sequence generated by Algorithm 1 which converges to $\bar{x}$. Let $\bar{x}$ be a regular point of F. Suppose that Assumption 2 and (4.32) hold. If $\beta_k \geq 0$, d_k is a descent direction of F, and α_k satisfies strong Wolfe condition (4.9), then $\liminf_{k\to\infty} \|v_k\| = 0$.*

Proof: We prove the theorem by the method of contradiction. Suppose that there exists a positive real number $\tilde{t}$ such that $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$. From Step 3 of Algorithm 1, for all $k \geq 1$, we have

$$\begin{aligned} & -\beta_k d_{k-1} = -d_k + v_k \\ \implies & \beta_k^2 \|d_{k-1}\|^2 \leq \big(\|d_k\| + \|v_k\|\big)^2 \\ \implies & \beta_k^2 \|d_{k-1}\|^2 \leq 2\|d_k\|^2 + 2\|v_k\|^2 \text{ using (iii) of Lemma 1.15} \\ \implies & \|d_k\|^2 \geq -\|v_k\|^2 + \tfrac{\beta_k^2}{2}\|d_{k-1}\|^2. \end{aligned} \tag{4.34}$$

Since $\beta_k \geq 0$ and $d_k = v_k + \beta_k d_{k-1}$, therefore for all $j \in [\omega_k]$ and $w \in C$, we have

$$\begin{aligned} \langle \mathrm{J}\, f^{a_j^k}(x_k) d_k, w\rangle &= \langle \mathrm{J}\, f^{a_j^k}(x_k) v_k, w\rangle + \beta_k \langle \mathrm{J}\, f^{a_j^k}(x_k) d_{k-1}, w\rangle \\ \implies \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_k) &\leq \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) v_k) + \beta_k \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k) d_{k-1}). \end{aligned}$$

Thus, we have

$$0 \le -\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) \le -\max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k) + \beta_k \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_{k-1}). \quad (4.35)$$

The sequence $\{x_k\}$ converges to a regular point of F. There are only a finite number of subsets of $[p]$. Therefore, using Lemma 4.4, we have $Q \subseteq P_{\bar{x}}$ and $\bar{a} \in Q$ such that

$$\omega_k = \bar{\omega}, P_{x_k} = Q, \text{ and } a^k = \bar{a} \text{ for all } k. \quad (4.36)$$

Thus, from (4.35) and (4.36), for all k, we have

$$\begin{aligned} -\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)v_k) &\le -\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k) + \beta_k \max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_{k-1}) \\ &\overset{(4.9)}{\le} -\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k) - \sigma\beta_k \max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})d_{k-1}), \quad (4.37) \end{aligned}$$

which gives

$$\begin{aligned} \Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)v_k)\Big)^2 \le \Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k)\Big)^2 + \sigma^2\beta_k^2\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})d_{k-1})\Big)^2 \\ + 2\sigma\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k)\Big)\beta_k\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})d_{k-1})\Big), \end{aligned}$$

which implies, by using (ii) of Lemma 1.15 with $\hat{a} = \sigma\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k)\Big)$, $\hat{b} = \beta_k\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})d_{k-1})\Big)$ and $\hat{c} = 1$, that $\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)v_k)\Big)^2$

$$\le (1+2\sigma^2)\Bigg[\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k)\Big)^2 + \tfrac{\beta_k^2}{2}\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})d_{k-1})\Big)^2\Bigg]. \quad (4.38)$$

In view of (4.11) and (4.38), we get

$$\tfrac{1}{4(1+2\sigma^2)}\|v_k\|^4 \le \Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)d_k)\Big)^2 + \tfrac{\beta_k^2}{2}\Big(\max_{j\in[\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})d_{k-1})\Big)^2. \quad (4.39)$$

Note that $\dfrac{\left(\max\limits_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_k)d_k)\right)^2}{\|d_k\|^2}+\dfrac{\left(\max\limits_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_{k-1})d_{k-1})\right)^2}{\|d_{k-1}\|^2}$

$$
\begin{aligned}
&= \frac{1}{\|d_k\|^2}\left[\left(\max_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_k)d_k)\right)^2+\frac{\|d_k\|^2}{\|d_{k-1}\|^2}\left(\max_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_{k-1})d_{k-1})\right)^2\right]\\
&\overset{(4.34)}{\geq} \frac{1}{\|d_k\|^2}\left[\left(\max_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_k)d_k)\right)^2+\left(\frac{\beta_k^2}{2}-\frac{\|v_k\|^2}{\|d_{k-1}\|^2}\right)\left(\max_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_{k-1})d_{k-1})\right)^2\right]\\
&\overset{(4.39)}{\geq} \frac{\|v_k\|^2}{\|d_k\|^2}\left[\frac{1}{4(1+2\sigma^2)}\|v_k\|^2-\frac{\left(\max\limits_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\,f^{\bar{a}_j}(x_{k-1})d_{k-1})\right)^2}{\|d_{k-1}\|^2}\right]\\
&\geq \frac{1}{4(1+2\sigma^2)}\frac{\|v_k\|^4}{\|d_k\|^2}\\
&\geq \frac{\tilde{t}^4}{4(1+2\sigma^2)}\frac{1}{\|d_k\|^2}\ \text{using } \|v_k\|\geq\tilde{t}.
\end{aligned}
\tag{4.40}
$$

The Zoutendijk-type condition (4.27) holds under the hypothesis that we have taken in the statement of the theorem. Therefore, from (4.40), we have

$$\sum_{k=0}^{\infty}\frac{1}{\|d_k\|^2}<+\infty,$$

which is a contradiction to (4.32). Hence, the proof is complete. □

Next, we give two different choices for parameter β_k in Step 4 of Algorithm 1 and analyze the convergence properties of Algorithm 1 for these two choices. One choice is the extension of the classical Fletcher-Reeves (FR) rule, and the other extends the conjugate descent (CD) rule given in [162].

4.5.2 Fletcher-Reeves method

In this section, we give the Fletcher-Reeves choice for β_k, at k-th iteration for $k \geq 1$, for the proposed nonlinear conjugate gradient method, which is as follows:

$$\beta_k^{FR} = \frac{\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)}{\max\limits_{j\in[\omega_{k-1}]} \varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1})v_{k-1})}. \tag{4.41}$$

It is easy to see that if $p = 1$, i.e., $F(x) = \{f^1(x)\}$ for all $x \in \mathbb{R}^n$ in SOP (1.3), then β_k^{FR} in (4.41) is equal to the FR formula considered in [162] for nonlinear conjugate gradient method for vector optimization problem.

In the following theorem, we prove the convergence of Algorithm 1 when we choose β_k such that $|\beta_k| \leq \delta\beta_k^{FR}$ for $\delta \in [0, 1)$ under some suitable assumptions.

Theorem 4.3 *Consider the sequence* $\{x_k\}$ *generated by Algorithm 1. Assume that Assumption 2 holds. Let* $|\beta_k| \leq \delta\beta_k^{FR}$ *for* $\delta \in [0, 1)$, d_k *satisfies sufficient descent condition* (4.13)*, and* α_k *satisfies* (4.8)*. Then,* $\liminf\limits_{k\to\infty} \|v_k\| = 0$.

Proof: Let us assume that there exists a $\tilde{t} > 0$ such that $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$. From Step 2 of Algorithm 1, we have $\|d_k\|^2 \leq (\|v_k\|+|\beta_k|\|d_{k-1}\|)^2$ for all $k \geq 1$. Let $\hat{a} = \|v_k\|$, $\hat{b} = |\beta_k|\|d_{k-1}\|$, and $\hat{c} = \frac{\delta}{\sqrt{2(1-\delta^2)}}$. Then, by (iv) of Lemma 1.15 and $\|d_k\|^2 \leq (\|v_k\|+|\beta_k|\|d_{k-1}\|)^2$, we have

$$\|d_k\|^2 \leq (\|v_k\|+|\beta_k|\|d_{k-1}\|)^2 \leq \tfrac{1}{1-\delta^2}\|v_k\|^2 + \tfrac{1}{\delta^2}\beta_k^2\|d_{k-1}\|^2. \tag{4.42}$$

Since x_k is not a stationary point, therefore $\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) < 0$. Thus, using

(4.42), we get $\frac{\|d_k\|^2}{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2}$

$$\begin{aligned}
&\leq \frac{1}{1-\delta^2}\frac{\|v_k\|^2}{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2} + \frac{1}{\delta^2}\beta_k^2\frac{\|d_{k-1}\|^2}{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2}\\
&\leq \frac{1}{1-\delta^2}\frac{\|v_k\|^2}{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2} + \frac{\|d_{k-1}\|^2}{\left(\max\limits_{j\in[\omega_{k-1}]} \varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1})v_{k-1})\right)^2} \qquad (4.43)
\end{aligned}$$

using $|\beta_k| \leq \delta\beta_k^{FR}$. Note that $4\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2 \geq \|v_k\|^4 \geq \tilde{t}^4$. Therefore, from (4.43),

$$\frac{\|d_k\|^2}{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2} \leq \frac{4}{(1-\delta^2)\tilde{t}^2} + \frac{\|d_{k-1}\|^2}{\left(\max\limits_{j\in[\omega_{k-1}]} \varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1})v_{k-1})\right)^2}.$$

By following the same way repeatedly, we obtain

$$\begin{aligned}
\frac{\|d_k\|^2}{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2} &\leq \frac{4}{(1-\delta^2)\tilde{t}^2}k + \frac{\|d_0\|^2}{\left(\max\limits_{j\in[\omega_0]} \varphi(\mathrm{J}\, f^{a_j^0}(x_0)v_0)\right)^2}\\
&\leq \frac{4}{(1-\delta^2)\tilde{t}^2}k + \frac{4}{\tilde{t}^2} \text{ as } d_0 = v_0\\
&= \frac{4}{\tilde{t}^2}\left(\frac{1}{1-\delta^2}k+1\right). \qquad (4.44)
\end{aligned}$$

Hence, we get

$$\frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2}{\|d_k\|^2} \geq \frac{\tilde{t}^2(1-\delta^2)}{4}\frac{1}{k+1-\delta^2} \geq \frac{\tilde{t}^2(1-\delta^2)}{4}\frac{1}{k+1}. \qquad (4.45)$$

Since d_k satisfies sufficient descent condition (4.13), therefore by (4.45), we have

$$\sum_{k\geq 0}\frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)d_k)\right)^2}{\|d_k\|^2} \geq \sum_{k\geq 0} c^2\frac{\left(\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\right)^2}{\|d_k\|^2}$$

$$\geq c^2 \frac{\tilde{t}^2(1-\delta^2)}{4}$$
$$\geq \sum_{k\geq 0} \frac{1}{k+1} = +\infty,$$

which gives a contradiction to Zoutendijk-type condition (4.27). Therefore, the assumption $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$ is wrong. Hence, $\liminf_{k\to\infty} \|v_k\| = 0$. □

Next, we prove the convergence of the method, when parameter β_k is chosen such that $0 \leq \beta_k \leq \delta\beta_k^{FR}$ and α_k satisfies the strong Wolfe condition (4.9), under the assumption that the sequence $\{x_k\}$ generated by Algorithm 1 converges to a regular point.

Theorem 4.4 *Suppose that the sequence $\{x_k\}$ generated by Algorithm 1 converges to a regular point of F. Assume that Assumption 2 holds. Let $0 \leq \beta_k \leq \delta\beta_k^{FR}$ for $\delta \in [0,1)$, d_k is a descent direction for F at x_k ,and α_k satisfies (4.9). Then, $\liminf_{k\to\infty} \|v_k\| = 0$.*

Proof: Since x_k is not a stationary point of F for all $k \geq 0$, therefore by (4.11), for all $k \geq 0$,

$$\begin{aligned}
\|v_k\|^2 &< -2 \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) \\
&\leq 2 \max_{j\in[\omega_k]} \{|\varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)|\} \\
&\leq 2 \max_{j\in[\omega_k]} \{\|\mathrm{J}\, f^{a_j^k}(x_k)v_k\|\} \text{ by (iii) of Lemma 1.13} \\
&\leq 2\|v_k\| \max_{j\in[\omega_k]} \{\|\mathrm{J}\, f^{a_j^k}(x_k)\|\},
\end{aligned}$$

which implies that $\{v_k\}$ is bounded because $\lim_{k\to\infty} \|\mathrm{J}\, f^{a_j^k}(x_k)\|$ exists for each $j \in [w_k]$. Therefore, we have a constant γ such that $\|v_k\| \leq \gamma$ for all k. From this, it is easy to see that for all $k \geq 0, j \in [\omega_k]$ and $w \in C$,

$$0 > \max_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k) \geq \langle \mathrm{J}\, f^{a_j^k}(x_k)v_k, w\rangle \geq -\|\mathrm{J}\, f^{a_j^k}(x_k)\|\|v_k\| \geq -\gamma\tilde{b}, \tag{4.46}$$

where $\|\mathrm{J}\, f^{a_j^k}(x_k)\| \leq \tilde{b}$ for all $j \in [\omega_k]$.

Now, we assume that there exists a $\tilde{t} > 0$ such that $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$. It is easy to see that by doing the same calculation and analysis as in Theorem 4.3, condition (4.44) holds. Thus, using (4.46) and (4.44),

$$\|d_k\|^2 \leq 4\frac{\gamma^2\tilde{b}^2}{\tilde{t}^2(1-\delta^2)}k + 4\frac{\gamma^2\tilde{b}^2}{\tilde{t}^2},$$

which implies that

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} = +\infty.$$

Now using the Theorem 4.2, we get $\liminf_{k\to\infty}\|v_k\| = 0$ which is a contradiction to assumption $\|v_k\| \geq \tilde{t}$ for all $k \geq 0$ and hence the proof is complete. □

In scalar optimization, if $0 \leq \beta_k \leq \beta_k^{FR}$ and α satisfies strong Wolfe condition (4.9) with $\sigma < 0.5$, it is possible to show that the descent direction d_k generated by Algorithm 1 is sufficient descent [3]. However, it is an open problem in vector optimization. We could not also prove or disprove such conditions for SOP (1.3) in this work.

4.5.3 Conjugate descent method

The formula for conjugate descent choice for β_k, at k-th iteration for $k \geq 1$, for the proposed nonlinear conjugate gradient method, is as follows:

$$\beta_k^{CD} = \frac{\max\limits_{j\in[\omega_k]} \varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)}{\max\limits_{j\in[\omega_{k-1}]} \varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1})d_{k-1})}. \tag{4.47}$$

It is easy to see that if $p = 1$, i.e., $F(x) = \{f^1(x)\}$ for all $x \in \mathbb{R}^n$ in SOP (1.3), then β_k^{CD} in (4.47) is equal to the CD formula considered in [162] for nonlinear conjugate gradient method for vector optimization problems.

In the next lemma, we prove that d_k satisfies sufficient descent condition when we choose $0 \leq \beta_k \leq \beta_k^{CD}$ with constant $c = 1 - \sigma$.

Lemma 4.5 *Suppose that the sequence $\{x_k\}$ generated by Algorithm 1 converges to a regular point of F. Let $0 \leq \beta_k \leq \beta_k^{CD}$ and α_k satisfies the strong Wolfe condition (4.9). Then, d_k satisfies the sufficient descent condition with constant $1 - \sigma$.*

Proof: We prove it by the principle of mathematical induction. For $k = 0$, it is easy to see that d_0 satisfies sufficient descent with constant $1 - \sigma$ as $d_0 = v_0$, $0 < \sigma < 1$ and $\max_{j \in [\omega_0]} \varphi(\mathrm{J}\, f^{a_j^0}(x_0) v_0) < 0$. Let us assume that d_{k-1} satisfies the sufficient condition for some $k \geq 1$ with constant $1 - \sigma$, i.e., we have

$$\max_{j \in [\omega_{k-1}]} \varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1}) d_{k-1}) \leq (1 - \sigma) \max_{j \in [\omega_{k-1}]} \varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1}) v_{k-1}) < 0.$$

Thus, $\beta_k^{CD} > 0$ and β_k is well-defined. Note that the sequence $\{x_k\}$ converges to a regular point of F. Therefore, using (4.36) and (4.37), we have

$$\begin{aligned}
\max_{j \in [\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k) d_k) &\leq \max_{j \in [\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k) - \sigma \beta_k \max_{j \in [\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1}) d_{k-1}) \\
&\leq \max_{j \in [\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k) - \sigma \beta_k^{CD} \max_{j \in [\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1}) d_{k-1}) \\
&\overset{(4.47)}{=} (1 - \sigma) \max_{j \in [\bar{\omega}]} \varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k),
\end{aligned}$$

which implies that d_k satisfies sufficient descent condition, and hence the proof is complete. □

Now we prove the convergence of the proposed method if β_k is chosen as $\beta_k = \tilde{\delta} \beta_k^{CD}$, where $0 \leq \tilde{\delta} < 1 - \sigma$.

Theorem 4.5 *Suppose that the sequence $\{x_k\}$ generated by Algorithm 1 converges to a regular point of F. Assume that Assumption 2 holds. Let $\beta_k = \tilde{\delta} \beta_k^{CD}$ for $\tilde{\delta} \in [0, 1 - \sigma)$, and α_k satisfies (4.9). Then, $\liminf_{k \to \infty} \|v_k\| = 0$.*

Proof: We have

$$\begin{aligned}
0 \ \leq \ \beta_k \ &= \frac{\tilde{\delta}}{(1-\sigma)}(1-\sigma)\frac{\max\limits_{j\in[\omega_k]}\varphi(\mathrm{J}\, f^{a_j^k}(x_k)v_k)}{\max\limits_{j\in[\omega_{k-1}]}\varphi(\mathrm{J}\, f^{a_j^{k-1}}(x_{k-1})d_{k-1})}\\
&\leq \frac{\tilde{\delta}}{(1-\sigma)}\frac{\max\limits_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\, f^{\bar{a}_j}(x_k)v_k)}{\max\limits_{j\in[\bar{\omega}]}\varphi(\mathrm{J}\, f^{\bar{a}_j}(x_{k-1})v_{k-1})} \text{ using (4.36) and Lemma 4.5}\\
&= \tfrac{\tilde{\delta}}{(1-\sigma)}\beta_k^{FR},
\end{aligned}$$

which implies that $\liminf\limits_{k\to\infty}\|v_k\| = 0$ using $0 \leq \frac{\tilde{\delta}}{(1-\sigma)} < 1$ and Theorem 4.4. □

4.6 Numerical results

In this section, we consider some numerical examples. These examples are solved by Algorithm 1 with two different choices (FR and CD) for β_k. In our implementation, we take a standard cone, i.e., $K = \mathbb{R}^m_+$ in all the instances except for Example 4.6. The set C is taken as the canonical basis of $\mathbb{R}^m$ for all the examples except for Example 4.6 and $e = (1, 1, \ldots, 1)^\top \in \operatorname{int}(K)$ throughout this section. The parameters ρ and σ in Step 5 are taken as 0.0001 and 0.1, respectively. For the stopping condition, we choose $\|v_k\| \leq 0.001$, or a maximum number of 100 iterations are reached. We implement all the calculations in MATLAB 2018b software. This MATLAB software is installed on a Windows 10 machine equipped with a 1.90 GHz CPU and 8 GB RAM.

To find $\operatorname{Min}(F(x_k), K)$ in Step 1 of Algorithm 1, we follow the crude way of comparing the elements in $F(x_k)$ pair-wise. To calculate $(a^k, v_k) = \operatorname{argmin}_{(a,v)\in P_{x_k}\times\mathbb{R}^n}\vartheta_{x_k}(a, v)$ in Step 2, we use the inbuilt function *fminsearch* in MATLAB. For the step length in Step 5, we use the algorithm given in [163] to find the α_k satisfying the strong Wolfe condition (4.9). We take some test problems from the literature, while some are freshly introduced in this chapter. For each problem, we generate 100 random initial points and make a four-column table. In the table, the first column indicates the number of initial points that are solved by the method. In the second column, we mention the method to solve the problem. We have taken three different methods to solve the problems. Two of them are generated from Algorithm 1 with $\beta_k = 0.98\beta_k^{FR}$, and $\beta_k = 0.99(1-\sigma)\beta_k^{CD}$ (these two choices for β_k are taken in [162] as well), named as FR and CD, respectively. The third method is the steepest descent method given in [32] and is abbreviated as SD. The third column is a 6-tuple (Min, Max, Mean, Median, Mode, Std.D) whose components are the minimum, maximum, mean, median, mode, and standard deviation of the number of iterations by which the stopping condition is reached. Similarly, the fourth column is also a 6-tuple (Min, Max, Mean, Median, $\lceil$Mode$\rceil$, Std.D) that indicates the minimum, maximum, mean, median, least integer greater or equal to mode, and standard deviation of CPU time (in seconds) taken by the algorithm to reaching the stopping condition. We depict the values of F at each iteration for each problem in which the initial and the final points are shown in black and red color, respectively, while the cyan color indicates the intermediate points. In some figures, we use different shapes such as $\bullet, +$, and $\triangle$ to denote the values of F for different initial points. In these figures, the value of F at the initial point, at the intermediate points and at the final point generated by the method is depicted by the same shape for a particular initial point. For example, if the value of F at a particular initial point is depicted by black bullet $\bullet$, then the value of F at intermediate points and at the final point generated by the algorithm is depicted by cyan bullet $\bullet$ and red bullet $\bullet$, respectively. This is applicable to other shapes ($+$ and $\triangle$) also.

A brief description of all the considered examples is given below.

(i) Example 4.1 is taken from [84]. In this example, the number of variables (n) is one, and the number of components of each vector-valued function (m) is two. The cardinality of the objective function F, i.e., the number of vector-valued functions in the objective function in this example, is 5. The ordering cone K is taken as

$\mathbb{R}^2_+$.

(ii) Example 4.2 is taken from [84]. This example is the robust counterpart of the vector-valued facility location problem under uncertainty. In this example, the set $\mathcal{U}$ is an uncertainty set and l_1, l_2, and l_3 are the chosen locations. The values of n and m are 2 and 3, respectively in this example. The cardinality of the objective function F, i.e., the number of vector-valued functions in the objective function in this example, is 100. The ordering cone is taken as $\mathbb{R}^3_+$.

(iii) Example 4.3 is freshly introduced in this chapter. In this example, the values of n and m are 1 and 2, respectively. The cardinality of the objective function F, i.e., the number of vector-valued functions in the objective function in this example, is 40. The ordering cone is taken as $\mathbb{R}^2_+$.

(iv) Example 4.4 is a slight modification of Example 4.2 taken in [116]. The highlighted point of this modification is that we get different types of shapes of the functional values for the different points; for instance, see figures (a) and (b) of Figure 4.4. In this example, the values of n and m are 2 and 2, respectively. The cardinality of the objective function F, i.e., the number of vector-valued functions in the objective function in this example, is 50. The ordering cone is taken as $\mathbb{R}^2_+$.

(v) Example 4.5 is freshly introduced in this chapter. In this example, the values of n and m are 2 and 3, respectively. The cardinality of the objective function F, i.e., the number of vector-valued functions in the objective function in this example, is 14. The ordering cone is taken as $\mathbb{R}^2_+$.

(vi) Example 4.6 is also a new problem introduced in this chapter. In this example, the values of n and m are 1 and 2, respectively. The cardinality of the objective function F, i.e., the number of vector-valued functions in the objective function in this example, is 4. The ordering cone is taken as $K' = \{(y_1, y_2)^\top \in \mathbb{R}^2 :$

$4y_1 - y_2 \geq 0$ and $-4y_1 + 5y_2 \geq 0\}$, which is different from the standard cone in $\mathbb{R}^m$. According to the cone K', we find the set C with the help of the dual of K' and keep the other parameters the same for this problem. This makes Example 4.6 more difficult and different from the other previously considered problems in this chapter.

Example 4.1 *Let $F : \mathbb{R} \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

$$F(x) = \left\{ f^1(x), f^2(x), f^3(x), f^4(x), f^5(x) \right\},$$

where f^i, for $i \in [5]$, are from $\mathbb{R}$ to $\mathbb{R}^2$, defined by

$$f^i(x) = \begin{pmatrix} x \\ \frac{x}{2}\sin(x) \end{pmatrix} + \sin^2(x) \left[\frac{(i-1)}{4} \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \left(1 - \frac{i-1}{4}\right) \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right].$$

In Figure 4.1, the collection of objective values for all $x \in [-12, -7]$ is depicted with gray color while the black color bullet dots denote the value of F at initial point $x_0 = -10.3000$ and red color bullet dots denote the value of F at final point at which CD method stops. There is no cyan bullet dots in Figure 4.1, which indicates that CD method takes only one iteration to reach the stationary point from this initial point. In this example, we generate 100 initial points randomly chosen from the interval $[-5\pi, 5\pi]$ and run the methods FR, CD, and SD. The performance of these three methods for the randomly chosen points for this example is given in Table 4.1.

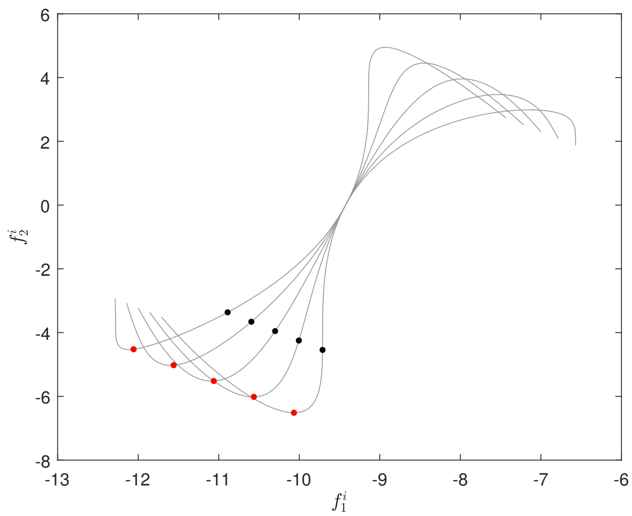


Figure 4.1: The value of F at intermediate iterates generated by CD method for Example 4.1 for the initial point $x_0 = -10.3000$

The performance of the three different methods for Example 4.1 is shown in the following table.

Table 4.1: Performance comparison of FR, CD, and SD methods on Example 4.1

Number of solved points	Method	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	FR	$(0, 3, 0.6500, 0, 0, 0.8333)$	$(0.1141, 19.7676, 5.1042, 0.1398, 1, 6.1081)$
100	CD	$(0, 3, 0.6200, 0, 0, 0.8851)$	$(0.1119, 18.5214, 4.3529, 0.1260, 1, 5.9638)$
100	SD	$(0, 38, 5.1800, 0, 0, 8.1741)$	$(0.1150, 67.3828, 21.3525, 18.7999, 34, 12.0271)$

Example 4.2 *Let $F : \mathbb{R}^2 \rightrightarrows \mathbb{R}^3$ be a set-valued map defined as*

$$F(x) = \left\{ f^1(x), f^2(x), \ldots, f^{100}(x) \right\},$$

where f^i, for $i \in [100]$, are from $\mathbb{R}^2$ to $\mathbb{R}^3$ such that

$$f^i(x) = \frac{1}{2} \begin{pmatrix} \|x - l_1 - u_i\|^2 \\ \|x - l_2 - u_i\|^2 \\ \|x - l_3 - u_i\|^2 \end{pmatrix},$$

where $l_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $l_2 = \begin{pmatrix} 0 \\ 8 \end{pmatrix}$, $l_3 = \begin{pmatrix} 8 \\ 0 \end{pmatrix}$ *and the set* $\mathcal{U} = \{u_i = (u_{1i}, u_{2i})^\top : i \in [100]\}$ *is an enumeration of the set*

$$\left\{-1, -1 + \tfrac{1}{r}, -1 + \tfrac{2}{r}, \ldots, -1 + \tfrac{2(r-1)}{r}, 1\right\} \times \left\{-1, -1 + \tfrac{1}{r}, -1 + \tfrac{2}{r}, \ldots, -1 + \tfrac{2(r-1)}{r}, 1\right\}$$

having $r = 4.5$.

In Figure 4.2, the gray points represent the set $(l_1 + u_i) \cup (l_2 + u_i) \cup (l_3 + u_i)$ *for all* $i \in [100]$ *and blue points represent* l_1, l_2, l_3. *The value of initial points and final points generated by CD method is denoted by black and red colors, respectively, in Figure 4.2. There are no intermediate points in Figure 4.2 because CD method takes only one iteration to reach the stationary points from these chosen initial points. In this example, we generate 100 initial points randomly chosen from the square* $[-50, 50] \times [-50, 50]$ *and run the methods FR, CD, and SD. The performance of these three methods for the randomly chosen points for this example is given in Table 4.2.*

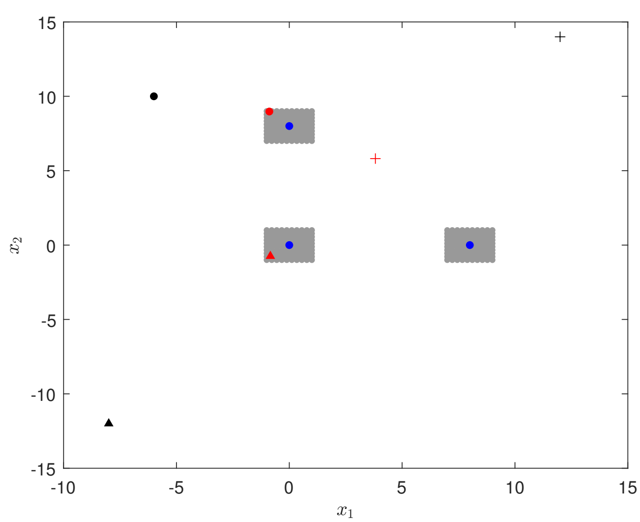


Figure 4.2: The value of x_k for three different initial points $((-8, -12)^\top, (-6, 10)^\top$ and $(12, 14)^\top)$ at each iteration generated by CD method for Example 4.2

The performance of the three different methods for Example 4.2 is shown in the *following table.*

Table 4.2: Performance comparison of FR, CD, and SD methods on Example 4.2

Number of solved points	Method	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	FR	$(0, 1, 0.9900, 1, 1, 0.1000)$	$(1.0053, 7.6198, 2.6961, 1.1344, 2, 2.2379)$
100	CD	$(0, 1, 0.9900, 1, 1, 0.1000)$	$(1.0257, 7.1041, 2.2424, 1.0826, 2, 2.0213)$
100	SD	$(0, 3, 1.1000, 1, 1, 0.6711)$	$(0.8222, 18.0552, 2.8121, 1.2572, 2, 2.7033)$

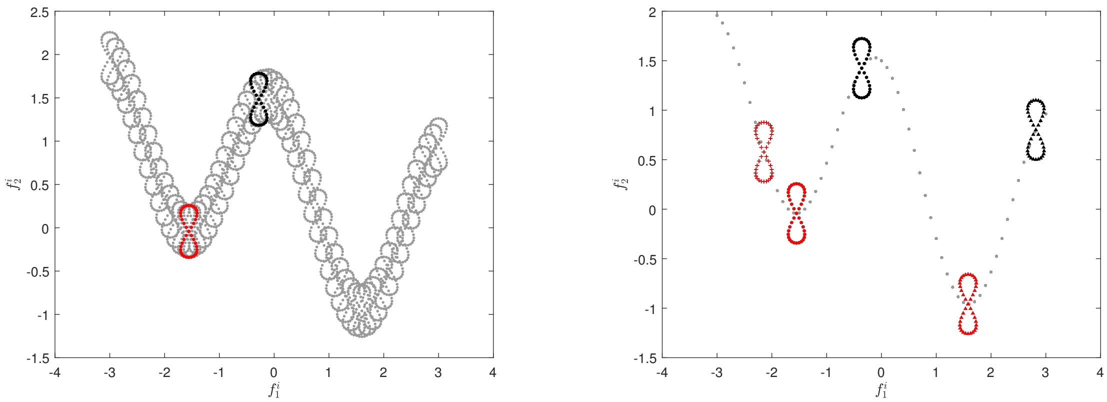
Example 4.3 *Let $F : \mathbb{R} \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

$$F(x) = \left\{ f^1(x), f^2(x), \ldots, f^{40}(x) \right\},$$

where f^i, for $i \in [40]$, are from $\mathbb{R}$ to $\mathbb{R}^2$ such that

$$f^i(x) = \begin{pmatrix} 0.3\cos\left(\frac{2\pi(i-1)}{40}\right)\sin\left(\frac{2\pi(i-1)}{40}\right) + x \\ 0.3\cos\left(\frac{2\pi(i-1)}{40}\right) + \cos(2x) + \frac{1}{(1+e^{2x})} \end{pmatrix}.$$

In (a) of Figure 4.3, the collection of discretized objective values for all $x \in [-3, 3]$ is depicted with gray color while the black color dots denote the value of F at initial point $x_0 = -0.2800$ and red color dots denote the value of F at final point at which CD method stops. In (b) of Figure 4.3, the output of CD method for three different initial points is exhibited. In this example, we generate 100 initial points randomly chosen from the interval $[-3, 3]$ and run the methods FR, CD, and SD. The performance of these three methods for the randomly chosen points for this example is given in Table 4.3.



(a) The value of F at intermediate iterates generated by CD method for Example 4.3 for the initial point $x_0 = -0.2800$

(b) The value of F at each iteration generated by CD method for Example 4.3 for three different randomly chosen initial points

Figure 4.3: Output of CD method for Example 4.3

The performance of the three different methods for Example 4.3 is shown in the following table.

Table 4.3: Performance comparison of FR, CD, and SD methods on Example 4.3

Number of solved points	Method	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	FR	$(0, 3, 0.7400, 0, 0, 0.9059)$	$(0.1933, 10.7919, 2.7668, 0.2300, 1, 3.2717)$
100	CD	$(0, 3, 0.7300, 0, 0, 0.9519)$	$(0.1921, 11.1003, 2.9315, 0.2300, 1, 3.6067)$
100	SD	$(0, 4, 0.8000, 1, 0, 0.8409)$	$(0.2279, 2.3555, 0.6903, 0.5716, 1, 0.4890)$

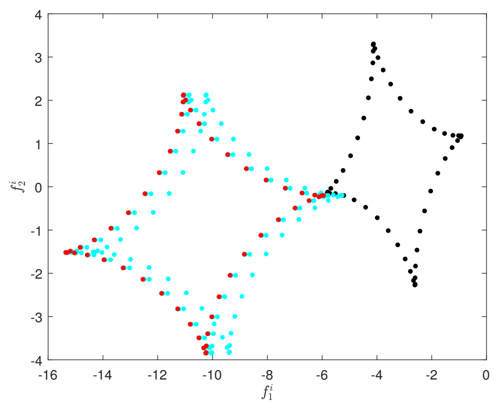
Example 4.4 *Let* $F : \mathbb{R}^2 \rightrightarrows \mathbb{R}^2$ *be a set-valued map defined as*

$$F(x) = \left\{ f^1(x), f^2(x), \ldots, f^{50}(x) \right\},$$

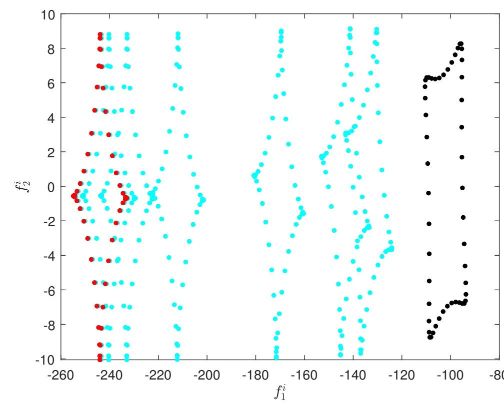
where f^i, *for* $i \in [50]$, *are from* $\mathbb{R}^2$ *to* $\mathbb{R}^2$ *as below*

$$f^i(x) = \begin{pmatrix} e^{\frac{x_1}{2}} \cos x_2 + x_1 \cos x_2 \sin^3\left(\frac{2\pi(i-1)}{50}\right) - x_2 \sin x_2 \cos\left(\frac{2\pi(i-1)}{50}\right) \\ e^{\frac{x_2}{20}} \sin x_1 + x_1 \sin x_2 \sin\left(\frac{2\pi(i-1)}{50}\right) + x_2 \cos x_2 \cos^3\left(\frac{2\pi(i-1)}{50}\right) \end{pmatrix}.$$

In (a) and (b) of Figure 4.4, the value of initial points, intermediate points, and final points generated by CD method is denoted by black, cyan, and red colors bullet dots, *respectively. In this example, we generate 100 initial points randomly chosen from the square $[-5\pi, 5\pi] \times [-5\pi, 5\pi]$ and run the methods FR, CD, and SD. The performance of these three methods for the randomly chosen points for this example is given in Table 4.4.*



(a) The value of F at each iteration generated by CD method for Example 4.4 for initial point $x_0 = (2.5040, -2.8717)^\top$



(b) The value of F at each iteration generated by CD method for Example 4.4 for initial point $x_0 = (9.8400, -8.1547)^\top$

Figure 4.4: Output of CD method for Example 4.4

The performance of the three different methods for Example 4.4 is shown in the following table.

Table 4.4: Performance comparison of FR, CD, and SD methods on Example 4.4

Number of solved points	Method	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
85	FR	$(0, 82, 14.6479, 4, 0, 21.0578)$	$(0.4207, 2130, 231.1153, 117.4000, 2, 346.6122)$
89	CD	$(0, 75, 12.9296, 3, 0, 19.1232)$	$(0.4000, 2073, 209.0708, 98.4000, 2, 340.9545)$
82	SD	$(0, 94, 15.6056, 4, 0, 23.5612)$	$(0.4116, 2468, 269.1139, 126.3500, 2, 448.5198)$

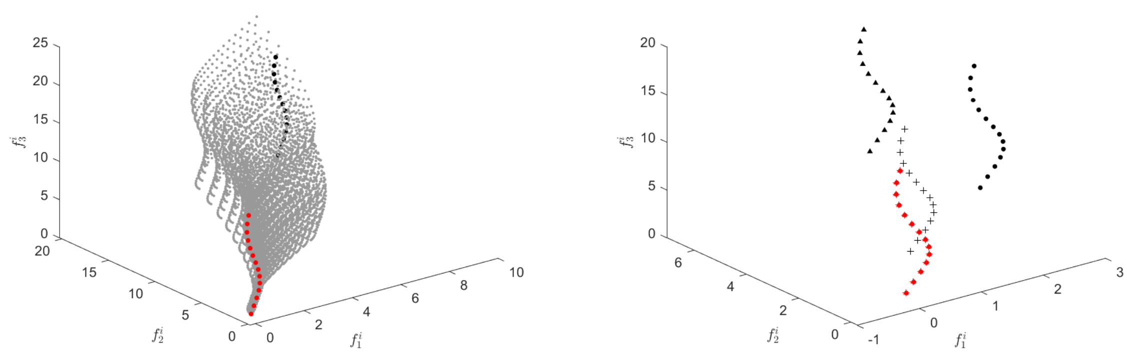
Example 4.5 *Let $F : \mathbb{R}^2 \rightrightarrows \mathbb{R}^3$ be a set-valued map defined as*

$$F(x) = \left\{ f^1(x), f^2(x), \ldots, f^{14}(x) \right\},$$

where f^i, for $i \in [14]$, are from $\mathbb{R}^2$ to $\mathbb{R}^3$, defined by

$$f^i(x) = \begin{pmatrix} x_1^2 + x_2^2 + 0.25 \sin\left(\frac{2\pi(i-1)}{14}\right) \\ 2(x_1^3 + x_2) + 0.25 \cos\left(\frac{2\pi(i-1)}{14}\right) \\ x_1^2 + x_2^2 + i \end{pmatrix}.$$

In Figure 4.5(a), the collection of discretized objective values for all $x \in [0,2]$ is depicted with gray color while the black color dots denote the value of F at initial point $x_0 = (1.7583, 1.9118)^\top$ and red color dots denote the value of F at final point at which CD method stops. In Figure 4.5(b), the output of CD method for three different initial points is exhibited. In this example, we generate 100 initial points randomly chosen from the square $[0,2] \times [0,2]$ and run the methods FR, CD, and SD. The performance of these three methods for the randomly chosen points for this example is given in Table 4.5.



(a) The value of F at each iteration generated by CD method for Example 4.5 for initial point $x_0 = (1.7583, 1.9118)^\top$

(b) The value of F at each iteration generated by CD method for Example 4.5 for three different randomly chosen initial points

Figure 4.5: Output of CD method for Example 4.5

The performance of the three different methods for Example 4.5 is shown in the following table.

Table 4.5: Performance comparison of FR, CD, and SD methods on Example 4.5

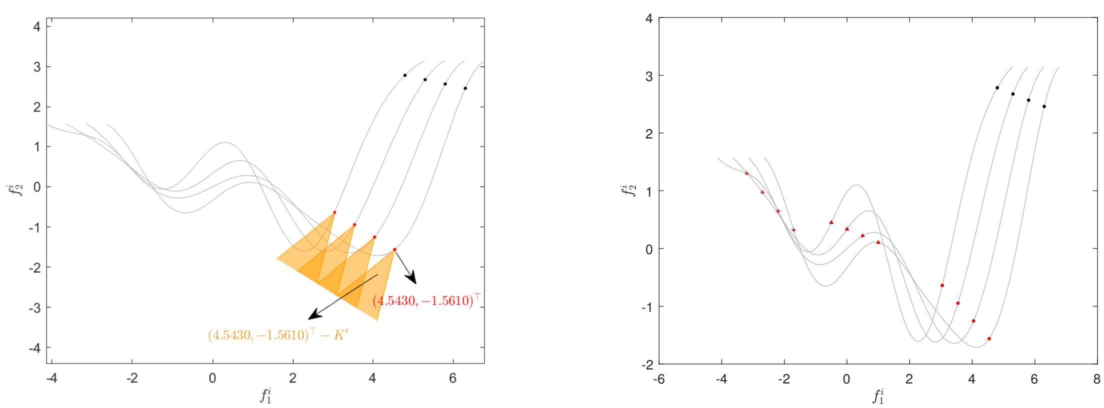
Number of solved points	Method	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	FR	$(1, 21, 1.6500, 1, 1, 2.3927)$	$(6.9702, 68.4791, 10.9771, 7.1476, 8, 9.8500)$
100	CD	$(1, 17, 1.5700, 1, 1, 1.8870)$	$(7.0316, 50.2363, 10.2016, 7.3843, 8, 7.1464)$
100	SD	$(1, 12, 1.9900, 1, 1, 2.2896)$	$(5.9710, 34.6351, 8.4209, 6.0777, 6, 5.5178)$

Example 4.6 *Let $F : \mathbb{R} \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

$$F(x) = \left\{ f^1(x), f^2(x), f^3(x), f^4(x) \right\},$$

where f^i, for $i \in [4]$, are from $\mathbb{R}$ to $\mathbb{R}^2$ as below

$$f^i(x) = \begin{pmatrix} x + \frac{(i-3)}{2} \\ \frac{x}{2}\cos x - \frac{(i-3)}{2}\sin^2 x \end{pmatrix}.$$



(a) The value of F at intermediate iterative points generated by CD method for Example 4.6 for the initial point $x_0 = 5.8$

(b) The value of F at each iteration generated by CD method for Example 4.6 for three different initial points 5.8, 0.5, and -2.2

Figure 4.6: Output of CD method for Example 4.6

The performance of the three different methods for Example 4.6 is shown in the following table.

In Figure 4.6(a), the collection of objective values for all $x \in [-\pi, 2\pi]$ is depicted with gray color while the black color dots denote the value of F at initial point $x_0 = 5.8$

Table 4.6: Performance comparison of FR, CD, and SD methods on Example 4.6

Number of solved points	Method	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	FR	$(0, 2, 0.3900, 0, 0, 0.7507)$	$(7.6934, 45.7067, 13.5836, 8.1615, 8, 10.5022)$
100	CD	$(0, 2, 0.3200, 0, 0, 0.6175)$	$(7.6887, 45.7790, 13.1272, 7.6666, 8, 11.7220)$
100	SD	$(0, 5, 0.7200, 0, 0, 1.4076)$	$(7.6942, 49.4739, 13.9506, 7.7894, 8, 11.8455)$

and red color dots denote the value of F at final point at which CD method stops. In this figure, it can be easily seen that the red points are the optimal value of F as the set $(4.5430, -1.5610)^\top - K'$ does not have any element of $F(x)$ other than $(4.5430, -1.5610)^\top$ for all $x \in [-\pi, 2\pi]$. In Figure 4.6(b), we have chosen three initial points 5.8, 0.5, and -2.2 and the value of F at these three points is denoted by $\bullet$, $\triangle$ and $+$, respectively. The points 0.5 and -2.2 are the stationary points because at these points CD method stops in 0-th iteration. However, the initial point 5.8 takes one iteration to reach the stationary point, which can be seen in figure (b) of Figure 4.6. In this example, we generate 100 initial points randomly chosen from the interval $[-\pi, 2\pi]$ and run the methods FR, CD, and SD. The performance of these three methods for the randomly chosen points for this example is given in Table 4.6.

Below, we give a brief description on the basis of the observation about the performance of all three methods when we run the methods for each example considered in this chapter.

(i) For Example 4.1, both the methods CD and FR perform very well. These two methods are better than SD method in terms of both iterations and CPU time. It is also observed that both methods, CD and FR, converge to stationary points in two iterations for most of the initial points. However, it is not the same for SD method.

(ii) For Example 4.2, both CD and FR methods perform better than SD method in terms of both iterations and CPU time. These two methods take only one iteration to stop even when the initial points are very far away from the solution, which is not the same for SD method. However, it is also observed that for most of the initial points SD method also stops in one iteration but not for all.

(iii) For Example 4.3, both CD and FR methods perform better than SD in terms of iterations. However, SD method takes less CPU time than CD and FR methods. It is observed that it happens because the calculation of step length for this particular example in CD and FR methods takes more time. Once the step length is calculated, these two methods converge to stationary points quickly. That is why CD and FR methods take fewer iterations but more CPU time than the SD method for this example.

(iv) For Example 4.4, both CD and FR methods perform better than SD in terms of both iterations and CPU time. For this example, the number of solved points by all three methods is different. CD method solves most number of points followed by FR.

(v) For Example 4.5, SD method performs better in terms of both iterations and CPU time than CD and FR methods. However, the Mean of iterations and CPU time shows that CD and FR methods are not much behind (in terms of performance) than the SD method for this example.

(vi) For Example 4.6, both CD and FR methods perform better than SD in terms of both iterations and CPU time. In this example, it is noticed that both CD and FR methods stop in one iteration for most of the initial points even when the initial point is far away from the stationary point. However, this is not the same for SD method.

4.7 Conclusion

In this chapter, we have proposed nonlinear conjugate gradient methods for unconstrained set-valued optimization problems, with the objective function being a collection of finitely many continuously differentiable vector-valued functions. An algorithm (Algorithm 1) with general conjugate parameter β_k has been presented. After that, two variants of the proposed method for two different choices of conjugate parameter β_k, namely Fletcher-Reeves (4.41) and conjugate descent (4.47) have been analyzed.

For Algorithm 1, a sufficient condition (Lemma 4.2) on β_k has been given to ensure that conjugate direction d_k satisfies the sufficient descent condition (4.13). The standard and strong Wolfe line search conditions ((4.8) and (4.9)) for the considered set optimization problems have been introduced. Further, we have proved the existence of step length α_k that satisfies the strong and standard Wolfe line search conditions (Proposition 4.3). To prove the convergence of the method, a Zoutendijk-type condition has been established (Proposition 4.4). It has also been proved that Algorithm 1 converges to a stationary point *without taking the regularity condition* (Definition 4.5) if d_k satisfies sufficient descent condition and α_k satisfies the standard Wolfe condition (Theorem 4.1). Subsequently, we have shown that Algorithm 1 converges to a stationary point if d_k is a descent direction for F at x_k and α_k satisfies the strong Wolfe condition (Theorem 4.2). Similarly, the convergence of FR method has been proved with and without taking regularity assumption if β_k is nonnegative and bounded above by any fraction of FR choice β_k^{FR} (Theorem 4.3 and Theorem 4.4). Further, we have reported that d_k satisfies sufficient descent condition with constant $(1-\sigma)$ if β_k is nonnegative and bounded above by conjugate descent choice β_k^{CD} (Lemma 4.5) under the regularity assumption. Using this result, the convergence of the CD method has been established with regularity assumption if β_k is nonnegative and bounded above by an appropriate fraction of conjugate descent choice β_k^{CD} (Theorem 4.5).

We have tested the proposed methods on some existing and freshly introduced set optimization problems. We have compared the performance of FR, CD, and the steepest descent method [32] on these set optimization problems. It is found that the CD method gives the best performance and the FR method gives the second best among these three methods.

****