Chapter 5

A projected gradient method for constrained set optimization problems with set-valued mappings of finite cardinality

5.1 Introduction

In this chapter, we extend our study from unconstrained set optimization problems to constrained set optimization problems. We propose a projected gradient method for constrained set optimization problems whose objective functions have finite cardinality. Similar to steepest and conjugate gradient methods for unconstrained optimization problems, the projected gradient method is one of the most popular and oldest methods for constrained optimization problems. The gradient projection method was originally presented by J. B. Rosen in 1956 [171]. A number of significant improvements in the method were made in [172, 173] in which the projected gradient method for linear constraints was given. After that, the projected gradient method for constrained optimization problems in which a constrained set is a closed convex set and its convergence for particular convex sets have been given in [21, 81]. Next, Gafni and Bertsekas [69] modified the work given in [21, 81] by proving the convergence of the projected gradient method for a general convex set regardless of the nature of the constrained convex set. Thereafter, different variants and applications of the projected gradient method have been proposed by many researchers in conventional optimization, for instance, see [37, 148, 175, 190, 191] and references therein. Drummond and Iusem extended the projected gradient method from scalar to vector optimization problems in [55]. After that, several variants of the projected gradient method have been proposed for vector or multiobjective optimization problems [38, 61, 67, 68]. Using the idea given in [55], we attempt to propose a projected gradient method for constrained set optimization problems whose objective function has a finite cardinality in this chapter.

5.2 Motivation

Recently, a steepest descent method [32] has been introduced for unconstrained set-valued optimization problems in which the objective set-valued mapping has a finite number of continuously differentiable selections. These mappings have their own importance [109]. Particularly, the set-valued optimization problems having such objective set-valued mappings occur in finding the robust solutions of vector optimization problems under a finite uncertainty set. To our best knowledge, there is no method in the literature for considered constrained set optimization problems. This motivated us to propose a projected gradient method for set optimization problems over a closed convex set whose objective function has a finite number of vector-valued functions.

5.3 Contributions

The major contributions in this chapter are as follows:

- Necessary optimality conditions for stationary point for CSOP (1.4) using Gerstewitz function is given.

- Two projected gradient methods for CSOP (1.4) are developed.
- The well-definedness of the proposed methods is given.
- Global convergence of the proposed methods is proved with regularity assumption.
- A numerical description of the proposed methods is given.

5.4 Optimality conditions

In this section, some optimality conditions are proposed for CSOP (1.4). First, we start this section by introducing some important set-valued indices related to set-valued mappings. These indices are analogous to those given in [32].

Definition 5.1 (Active indices related to set-valued mappings).

(i) *The active index of minimal elements associated with the objective set-valued mapping F of CSOP* (1.4) *is $I : \mathcal{S} \rightrightarrows [p]$, which is given by*

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

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

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

It is to notice that for any $x \in \mathcal{S}$, $I_u(x) = \emptyset$ for $u \notin \text{Min}(F(x), K)$; $I_u(x) \cap I_v(x) = \emptyset$ for any $u \neq v \in \mathcal{S}$; $I(x) = \bigcup_{u \in \text{Min}(F(x),K)} I_u(x)$.

Definition 5.2 (Cardinality of a set of minimal elements). *The map $\omega : \mathcal{S} \to \mathbb{R}$, which is defined by*

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

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

For the sake of notational convenience, we use the notation $\bar{\omega} = \omega(\bar{x})$, throughout this chapter, where $\bar{x} \in \mathcal{S}$.

Next, we give the definition of the partition set at a point $x \in \mathcal{S}$ in order to handle CSOP (1.4) given in [32].

Definition 5.3 (Partition set at a point). *Let $\{u_1^x, u_2^x, \ldots, u_{\omega(x)}^x\}$ be an enumeration of the set $\operatorname{Min}(F(x), K)$, where $x \in \mathcal{S}$. For the point $x \in \mathcal{S}$, the partition set is defined by*

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

In this chapter, at an iterative point $x_k \in \mathcal{S}$, 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_j^i for all $j \in [\omega(x_k)]$, where $i = 1, 2, \ldots$. Precisely, if $|P_{x_k}| = p_k$ and $\operatorname{Min}(F(x_k), K) = \left\{u_1^{x_k}, u_2^{x_k}, \ldots, u_{w(x_k)}^{x_k}\right\}$, then

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

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

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

We present now a result that connects CSOP (1.4) to a family of vector optimization problems. This family of vector optimization problems locally represents CSOP (1.4) around the point, which is exploited later to find weakly minimal points of the CSOP (1.4).

Theorem 5.1 *Let $\bar{x} \in \mathcal{S}$. 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 $\tilde{K} = \prod_{j=1}^{\bar{\omega}} K$. Consider the constrained vector optimization problem (CVOP) with respect to the ordering cone $\tilde{K}$:

$$\text{minimize } \tilde{f}^a(x) \text{ subject to } x \in \mathcal{S}. \tag{5.1}$$

Then, $\bar{x}$ is a local weakly minimal solution of CSOP (1.4) if and only if $\bar{x}$ is a local weakly minimal solution of CVOP (5.1) for every $a \in P_{\bar{x}}$.

Proof: The proof is similar to that of Lemma 3.1 in [32]. □

Next, we discuss about the stationary point of CSOP (1.4). In vector optimization, the concept of stationarity can be found in [66]. For an unconstrained vector optimization problem $\min_{x\in\mathbb{R}^n} f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued function, a point $\bar{x} \in \mathbb{R}^n$ is called a stationary point of f if for any $v \in \mathbb{R}^n$, the condition $\mathrm{J}\, f(\bar{x})v \cap (-\operatorname{int}(K)) = \emptyset$ holds. This condition has been extended in [55] for constrained vector optimization problems $\min_{x\in C} f_c(x)$, where $f_c : \mathbb{R}^n \to \mathbb{R}^m$ and C is a nonempty, closed, and convex subset of $\mathbb{R}^n$. A point $\bar{x} \in C$ is called a stationary point of f_c if

$$\mathrm{J}\, f_c(\bar{x})(C - \bar{x}) \cap (-\operatorname{int}(K)) = \emptyset. \tag{5.2}$$

Based on the condition (5.2), we propose a concept of stationary point for CSOP (1.4).

Definition 5.4 (Stationary point of CSOP). *A point $\bar{x} \in \mathcal{S}$ is called a stationary point of CSOP (1.4) if for every $a \in P_{\bar{x}}$,*

$$\mathrm{J}\, \tilde{f}^a(\bar{x})(\mathcal{S} - \bar{x}) \cap (-\operatorname{int}(\tilde{K})) = \emptyset. \tag{5.3}$$

As $\operatorname{int}(\tilde{K}) = \operatorname{int}(K) \times \operatorname{int}(K) \times \cdots \times \operatorname{int}(K)$ ($\bar{w}$ times), (5.3) implies that a point $\bar{x} \in \mathcal{S}$ is stationary for CSOP (1.4) if and only if for every $a = (a_1, a_2, \ldots, a_{\bar{\omega}}) \in P_{\bar{x}}$

there exists at least one $j \in [\bar{w}]$ such that

$$\mathrm{J}\, f^{a_j}(\bar{x})(\mathcal{S} - \bar{x}) \subseteq (-\operatorname{int}(K))^c. \tag{5.4}$$

Hence, a point $\bar{x} \in \mathcal{S}$ is nonstationary for CSOP (1.4) if and only if there exists an $a = (a_1, a_2, \ldots, a_{\bar{\omega}}) \in P_{\bar{x}}$ for which there exists $\bar{v} \in \mathcal{S}$ such that

$$\mathrm{J}\, f^{a_j}(\bar{x})(\bar{v} - \bar{x}) \in -\operatorname{int}(K) \text{ for all } j \in [\bar{\omega}]. \tag{5.5}$$

Remark 5.1 *If $\mathcal{S} = \mathbb{R}^n$, then the condition (5.3) reduces to*

$$\forall a \in P_{\bar{x}} : \ \mathrm{J}\, \tilde{f}^a(\bar{x})(v) \cap (-\operatorname{int}(\tilde{K})) = \emptyset \textit{ for all } v \in \mathbb{R}^n,$$
$$\textit{i.e.}, \forall a \in P_{\bar{x}}\ \exists\ j \in [\bar{w}] \textit{ such that } \mathrm{J}\, \tilde{f}^a(\bar{x})(v) \notin -\operatorname{int}(K) \textit{ for all } v \in \mathbb{R}^n,$$

which is identical to the result in Proposition 3.1 *in* [32]*. Thus, for $\mathcal{S} = \mathbb{R}^n$, the concept of stationary points in Definition 5.4 reduces to the definition of stationary point for unconstrained set optimization problems given in Chapter 4.*

We now aim for deriving a necessary condition for weak minimal solutions for the CSOP (1.4). With the help of Theorem 5.1, we note that if $\bar{x}$ is a weakly minimal point of CSOP (1.4), then $\bar{x}$ is a stationary point of the CVOP (5.1) for each $a \in P_{\bar{x}}$ since every weakly minimal solution of a vector optimization problem is one of its stationary points [66].

For any given $x \in \mathcal{S}$ and $\bar{\beta} > 0$, we define a function $\vartheta_x : P_x \times \mathcal{S}_x \to \mathbb{R}$ by

$$\vartheta_x(a, v) = \max_{j \in [\omega(x)]} \bar{\beta} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j}(x) v \right) \right\} + \tfrac{1}{2} \|v\|^2, \tag{5.6}$$

where $\mathcal{S}_x = \mathcal{S} - x$. Note that for every $x \in \mathcal{S}$ and $a \in P_x$, the function $\vartheta_x(a, \cdot)$ is strongly convex in $\mathcal{S}_x$ because Ψ_e is sublinear. Therefore, the function $\vartheta_x(a, \cdot)$ attains its minimum, and this minimum is unique. Specifically, for all $x \in \mathcal{S}$ and $a \in P_x$, we have

$$\begin{aligned} &\min_{v \in \mathcal{S}_x} \vartheta_x(a, v) \leq \vartheta_x(a, v_1) \text{ for all } v_1 \in \mathcal{S}_x \\ \implies &\min_{v \in \mathcal{S}_x} \vartheta_x(a, v) \leq \vartheta_x(a, 0) = 0. \end{aligned} \tag{5.7}$$

In fact, from (5.7), we can see that if $\bar{v} \in \mathcal{S}_x$ such that $\vartheta_x(a, \bar{v}) = \min_{v \in \mathcal{S}_x} \vartheta_x(a, v)$, then by (5.6), we have

$$\vartheta_x(a, \bar{v}) = 0 \text{ if and only if } \bar{v} = 0. \tag{5.8}$$

Also, note that the partition set P_x has finite elements. Therefore, ϑ_x attains its minimum over the set $P_x \times \mathcal{S}_x$. Let $\phi : \mathcal{S} \to \mathbb{R}$ be the function

$$\phi(x) = \min_{(a,v) \in P_x \times \mathcal{S}_x} \vartheta_x(a, v). \tag{5.9}$$

Then, by (5.7), for all $x \in \mathcal{S}$, we have

$$\phi(x) \leq 0. \tag{5.10}$$

Moreover, if for $(\bar{a}, \bar{v}) \in P_x \times \mathcal{S}_x$ we have $\phi(x) = \vartheta_x(\bar{a}, \bar{v})$, then from (5.8), we get

$$\phi(x) = 0 \text{ if and only if } \bar{v} = 0. \tag{5.11}$$

Accumulating all, we obtain the following result.

Proposition 5.1 (Necessary condition for weakly minimal points).
Let $\hat{x}$ be a weakly minimal point of CSOP (1.4) and $(\hat{a}, \hat{v}) \in P_{\hat{x}} \times \mathcal{S}_{\hat{x}}$ be such that $\phi(\hat{x}) = \vartheta_{\hat{x}}(\hat{a}, \hat{v})$, where ϕ and $\vartheta_{\hat{x}}$ are as defined in (5.9) and (5.6), respectively. Then, $\hat{v} = 0$.

In the next section, we present a projected gradient algorithm for CSOP (1.4) where we exploit the conditions (5.3), (5.5) and Proposition 5.1 to capture stationary points of the CSOP (1.4).

5.5 Projected gradient method and its convergence

In this section, we propose an algorithm (Algorithm 2) for the solution approach for CSOP (1.4) based upon a projection of a point on the closed and convex constraint set $\mathcal{S}$. In the algorithm, first, an initial point is arbitrarily chosen from the constraint set. If it does not satisfy the necessary condition stated in Proposition 5.1 for a weakly minimal point, we update it as follows. An element a in the partition set of the current point is chosen. Thereafter, a descent direction of the function $\vartheta_x(a, \cdot)$ (see (5.6)) is calculated using the projection of the current point. We use the classical backtracking procedure of Armijo-type condition to find a suitable step size and update the iteration in the calculated descent direction.

A detailed explanation of each step in Algorithm 2 is given below.

In Algorithm 2, a feasible initial point x_0 is chosen first, and then set the iteration counter $k = 0$.

Next, the partition set P_k at x_k is calculated in Step 1. To calculate P_k, we first find the set of minimal elements of $F(x_k)$ with respect to the cone K, i.e., $\text{Min}(F(x_k), K)$, by pair-wise comparing the elements of $F(x_k)$. After that, using (ii) of Definition 5.1, we find the set of indices with respect to each element of the enumeration of $\text{Min}(F(x_k), K)$, and then using Definition 5.3, we calculate P_k.

In Step 2, we find (a^k, v_k) by solving a family of vector optimization problems

$$\min_{(a,v)\in P_k \times \mathcal{S}_{x_k}} \vartheta_{x_k}(a, v),$$

where the partition set P_k is finite and $\vartheta_{x_k}(a, \cdot)$ is strongly convex in $\mathcal{S}_{x_k}$ for each

Algorithm 2 Projected gradient algorithm for CSOP (1.4) using Armijo line search

Step 0: Initialization

Choose an initial point $x_0 \in \mathcal{S}$, a trial step length $\alpha \in (0,1)$ for each iteration, and $\bar{\beta} > 0$.
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 $\omega_k = |\text{Min}(F(x_k), K))|$.
Compute $P_k = P_{x_k}$ as given in Definition 5.3.

Step 2: Calculate the projected gradient descent direction

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

$$\vartheta_{x_k}(a,v) = \max_{j \in [\omega_k]} \bar{\beta} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j}(x_k) v \right) \right\} + \tfrac{1}{2} \|v\|^2. \tag{5.12}$$

Step 3: Stopping criterion

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

Step 4: Finding a step length

Find a step length $\alpha_k = 2^{-t_k}$, where t_k is calculated by

$$t_k = \min\{q \in \mathbb{Z}_+ : f^{a^k_j}(x_k + 2^{-q} v_k) \preceq f^{a^k_j}(x_k) + \alpha 2^{-q}\, \mathrm{J}\, f^{a^k_j}(x_k) v_k \text{ for all } j \in [\omega_k]\}.$$

Step 5: Update

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

$a \in P_{x_k}$.

After that, it is checked whether the current iterative point x_k satisfies the necessary optimality condition for a weakly minimal point, i.e., $v_k = 0$. In numerical computation, the stopping condition $v_k = 0$ is replaced by $\|v_k\| < \epsilon$ for the priory chosen ϵ, which is checked in Step 3.

If the current iterative point x_k does not satisfy the stopping condition given in Step 3, we need to update it and move further to the next iteration. To update the point x_k, we need a direction (which is v_k that has been calculated already in Step 2) and step length. To calculate the step length, we use a backtracking Armijo rule in Step 4.

In Step 4, we start with $q = 0$ (where $q \in \mathbb{Z}_+$) and check whether the inequality

$$f^{a_j^k}(x_k + 2^{-q}v_k) \preceq f^{a_j^k}(x_k) + \alpha 2^{-q} \operatorname{J} f^{a_j^k}(x_k)v_k \text{ for all } j \in [\omega_k] \tag{5.13}$$

is satisfied or not. If this inequality is satisfied for $q = 0$, then we take step length $\alpha_k = \frac{1}{2^0} = 1$. Otherwise, the inequality (5.13) is checked for $q = 1$. If $q = 1$ satisfies (5.13), we take $\alpha_k = \frac{1}{2^1} = \frac{1}{2}$. In a similar manner, we keep increasing the value of q until (5.13) is satisfied. Note that as q increases, the value of step length α_k decreases.

Using v_k and α_k, the current iterative point is updated to $x_{k+1} = x_k + \alpha_k v_k$ in Step 5.

Then, set the iteration counter $k = k + 1$, which makes $x_k = x_{k+1}$. Thereafter, we move to Step 1 and repeat the process until the stopping condition given in Step 3 is satisfied.

Remark 5.2 *Note that if we run Algorithm 2 for $p = 1$, i.e., for $F(x) = \{f^1(x)\}$, then clearly there is no need of Step 1, and in this case, Step 2 of Algorithm 2 reduces to finding v_k (projected descent direction) such that*

$$v_k = \operatorname*{argmin}_{v \in \mathcal{S}_{x_k}} \theta_{x_k}(v),$$

where $\mathcal{S}_{x_k} = \mathcal{S} - x_k$ and $\theta_{x_k} : \mathcal{S}_{x_k} \to \mathbb{R}$ is

$$\theta_{x_k}(v) = \bar{\beta} \max_{j \in [\omega_k]} \left\{ \Psi_e \left(\operatorname{J} f^1(x)v \right) \right\} + \tfrac{1}{2}\|v\|^2.$$

Further, Step 4 provides step-length with $a_j^k = 1$, i.e.,

$$t_k = \min\{q \in \mathbb{Z}_+ : f^1(x_k + 2^{-q}v_k) \preceq f^1(x_k) + \alpha 2^{-q} \operatorname{J} f^1(x_k)v_k \textit{ for all } j \in [\omega]\}.$$

The Step 0 *and Step* 5 *remain unchanged. From this observation, one can easily notice that when* $p = 1$*, the projected gradient method proposed in Algorithm* 2 *reduces to the method given in* [55]*. The only difference is that we have used the Gerstewitz function instead of the support of a generator of the dual cone* K^**. However, it is proved in* [31] *that the class of Gerstewitz function is a subset of the class of support function of a generator of the dual cone. Thus, for* $p = 1$*, Algorithm* 2 *is simply Algorithm* 2.1 *in* [55].

The method by Drummond and Iusem [55] cannot be straightly extended for the considered constrained set optimization problems due to the "prime fact" that while generating the sequence of iterates $\{x_k\}$ in Algorithm 2 for the CSOP (1.4), the vector optimization problem $\min_{(a,v)\in P_k\times\mathcal{S}_{x_k}} \vartheta_{x_k}(a,v)$ for finding (a^k, v_k), where v_k is the projected gradient descent direction, in Step 2 depends on the partition set P_k at x_k. It is to note that the partition set P_k at x_k is commonly different than the partition set P_{k+1} at x_{k+1}. In this case, we are handling the different families of vector optimization problems for finding (a^k, v_k) at each iteration. Subsequently, if a^k in Step 2 is different at each iteration, then the inequality (which has been used to find the step length α_k at x_k in Step 4)

$$f^{a_j^k}(x_k + 2^{-q}v_k) \preceq f^{a_j^k}(x_k) + \alpha 2^{-q}\,\mathrm{J}\, f^{a_j^k}(x_k)v_k \text{ for all } j \in [\omega_k],$$

is also different at each iteration. Therefore, Algorithm 2 is not a straightforward extension of the projected gradient method for vector optimization problems.

5.5.1 Convergence analysis

In this section, we show that Algorithm 2 is well-defined. After that, we prove the convergence of Algorithm 2. Note that well-definedness of Algorithm 2 depends on the following three points:

(i) Existence of (a^k, v_k) in Step 2, which is already proved in the paragraph just before Proposition 5.1.

(ii) Existence of α_k in Step 4, which we prove below in Proposition 5.4.

(iii) Whether each x_k belongs to $\mathcal{S}$ or not? Indeed $x_k \in \mathcal{S}$ for all $k = 0, 1, 2, \ldots$, which we prove in Proposition 5.3.

Hence, Algorithm 2 is well-defined.

The next result characterizes stationary points of CSOP (1.4) in terms of the functions ϑ_x and ϕ as defined in (5.6) and (5.9), respectively. Basically, it is proved that if Algorithm 2 stops in Step 3, a stationary point of CSOP (1.4) is obtained that satisfies necessary condition for weakly minimal points.

Proposition 5.2 *Let x be a point in $\mathcal{S}$. Let $(\bar{a}, \bar{v}) \in P_x \times \mathcal{S}_x$ be a point such that $\phi(x) = \vartheta_x(\bar{a}, \bar{v})$. Then, the following statements are equivalent:*

(i) *The point x is not a stationary point of CSOP (1.4),*

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

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

Proof: (i)→(ii). Suppose that $x \in \mathcal{S}$ is a nonstationary point. Then, by (5.5), there exists an element $a = (a_1, a_2, \ldots, a_{\omega(x)}) \in P_x$ and $v \in \mathcal{S}_x$ for which

$$\mathrm{J}\, f^{a_j}(x)v \in -\operatorname{int}(K) \text{ for all } j \in [\omega(x)]. \tag{5.14}$$

Therefore, $\Psi_e(\mathrm{J}\, f^{a_j}(x)v) < 0$ for all $j \in [\omega(x)]$, and hence $\bar{\beta} \max_{j\in[\omega(x)]}\{\Psi_e(\mathrm{J}\, f^{a_j}(x)v)\} < 0$. Since $\mathcal{S}$ is convex and $x \in \mathcal{S}$, therefore $\eta v + 0(1-\eta) \in \mathcal{S} - x$ for all $\eta \in [0, 1]$, which implies that $\eta v \in \mathcal{S} - x$. Hence, by the definition of ϑ_x,

$$\phi(x) = \vartheta_x(\bar{a}, \bar{v}) \le \vartheta_x(a, \eta v) = \bar{\beta} \max_{j\in[\omega(x)]}\{\Psi_e(\mathrm{J}\, f^{a_j}(x)\eta v)\} + \tfrac{1}{2}\|\eta v\|^2$$

$$\leq \eta\left(\bar{\beta}\max_{j\in[\omega(x)]}\{\Psi_e(\mathrm{J}\, f^{a_j}(x)v)\}+\tfrac{1}{2}\eta\|v\|^2\right) \text{ using (ii) of Proposition 1.2.} \quad (5.15)$$

Now take any η such that $0<\eta<\frac{1}{\|v\|^2}\left(-2\bar{\beta}\max_{j\in[\omega(x)]}\{\Psi_e(\mathrm{J}\, f^{a_j}(x)v)\}\right)$. Then, from (5.15), we get

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

Hence, the desired result is obtained.

(ii)→(iii). On contrary, let $\bar{v}=0$. Then, by (5.11), 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, $\bar{\beta}\max_{j\in[\omega(x)]}\{\Psi_e(\mathrm{J}\, f^{\bar{a}_j}(x)\bar{v})\}<\vartheta_x(\bar{a},\bar{v})=\phi(x)$, i.e.,

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

Hence, by (5.5), x is not a stationary point of CSOP (1.4). □

Remark 5.3 *Theorem 5.2 established an equivalence between (i), (ii), and (iii). So, by using (5.10), (5.11), and Theorem 5.2, we obtain that a point $\bar{x}\in\mathcal{S}$ is a stationary point of CSOP (1.4) if and only if $\phi(\bar{x})=0$ or $\bar{v}=0$.*

Next, we show that the sequence $\{x_k\}$ generated by Algorithm 2 for any initial point $x_0\in\mathcal{S}$ is a sequence of feasible points.

Proposition 5.3 *Let $\{x_k\}$ be a sequence generated by Algorithm 2. Then, $x_k\in\mathcal{S}$ for all $k\in\mathbb{N}$.*

Proof: We prove it by the principle of mathematical induction. By Step 0 of Algorithm 2, we have $x_0\in\mathcal{S}$. Let us assume that $x_k\in\mathcal{S}$. We require to show that x_{k+1} also belongs to $\mathcal{S}$. Note that $\alpha_k v_k \in \mathcal{S} - x_k$, where α_k and v_k are given in Step 4 and Step 2 in Algorithm 2, respectively. Therefore, we have $x_k + \alpha_k v_k \in \mathcal{S}$, which implies that $x_{k+1} \in \mathcal{S}$. Hence, the feasibility of the points in the sequence $\{x_k\}$ is established. $\square$

The following proposition gives the guarantee that at a nonstationary point, there exists a descent direction of the objective function F of CSOP (1.4).

Proposition 5.4 *Let $x \in \mathcal{S}$, $\alpha \in (0,1)$ and $(\bar{a}, \bar{v}) \in P_x \times \mathcal{S}_x$ be such that $\phi(x) = \vartheta_x(\bar{a}, \bar{v})$. Let us assume that x is not a stationary point of CSOP (1.4). Then, there exists $\bar{\gamma} < 1$ such that for all $j \in [\omega(x)]$ and $\gamma \in (0, \bar{\gamma}]$,*

$$f^{\bar{a}_j}(x + \gamma \bar{v}) \prec f^{\bar{a}_j}(x) + \alpha\gamma \operatorname{J} f^{\bar{a}_j}(x)\bar{v}. \tag{5.16}$$

Moreover, for all $\gamma \in (0, \bar{\gamma}]$, $F(x + \gamma\bar{v}) \prec^l F(x)$, i.e., $\bar{v}$ is a descent direction for F at x.

Proof: Given that x is not a stationary point of CSOP (1.4) and $(\bar{a}, \bar{v}) \in P_x \times \mathcal{S}_x$ be such that $\phi(x) = \vartheta_x(\bar{a}, \bar{v})$. Therefore, by Proposition 5.2, $\phi(x) < 0$, i.e.,

$$\begin{aligned}
&\Longrightarrow \max_{j \in [\omega(x)]} \{\Psi_e(\operatorname{J} f^{\bar{a}_j}(x)\bar{v})\} < -\tfrac{1}{2\beta}\|\bar{v}\|^2 < 0 \text{ because } \bar{v} \neq 0 \\
&\Longrightarrow \Psi_e(\operatorname{J} f^{\bar{a}_j}(x)\bar{v}) < 0 \text{ for all } j \in [\omega(x)] \\
&\Longrightarrow \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \in -\operatorname{int}(K) \text{ for all } j \in [\omega(x)] \text{ using (iv) of Proposition 1.2} \qquad (5.17) \\
&\Longrightarrow (1-\alpha) \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \in -\operatorname{int}(K) \\
&\Longrightarrow \operatorname{J} f^{\bar{a}_j}(x)\bar{v} - \alpha \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \in (-\operatorname{int}(K)). \qquad (5.18)
\end{aligned}$$

Since f^i is differentiable on $\mathbb{R}^n$ for any $i \in [p]$, $\operatorname{J} f^i(x)\bar{v} = \lim_{\delta \to 0+} \frac{1}{\delta}(f^i(x + \delta\bar{v}) - f^i(x))$ for all $i \in [p]$ (see [32]). Thus, from (5.18), for all $j \in [\omega(x)]$, we have

$$\lim_{\gamma \to 0+} \tfrac{1}{\gamma}(f^{\bar{a}_j}(x + \gamma\bar{v}) - f^{\bar{a}_j}(x)) - \alpha \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \in -\operatorname{int}(K)$$

$$\implies f^{\bar{a}_j}(x+\gamma\bar{v}) - f^{\bar{a}_j}(x) - \alpha\gamma \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \in -\operatorname{int}(K) \text{ for sufficiently small } \gamma$$
$$\implies f^{\bar{a}_j}(x+\gamma\bar{v}) \prec f^{\bar{a}_j}(x) + \alpha\gamma \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \text{ using Definition 1.26.}$$

Hence, there exists $\bar{\gamma} < 1$ such that for all $j \in [\omega(x)]$ and $\gamma \in (0, \bar{\gamma}]$, (5.16) holds.

Further, note from (5.17) that $\operatorname{J} f^{\bar{a}_j}(x)\bar{v} \in (-\operatorname{int}(K))$ for all $j \in [\omega(x)]$ because x is not a stationary point for CSOP (1.4). Therefore, for $\alpha > 0$, for all $j \in [\omega(x)]$ and $\gamma \in (0, \bar{\gamma}]$,

$$f^{\bar{a}_j}(x) + \alpha\gamma \operatorname{J} f^{\bar{a}_j}(x)\bar{v} \prec f^{\bar{a}_j}(x). \tag{5.19}$$

Now using Proposition 1.1, for all $\gamma \in (0, \bar{\gamma}]$, we have

$$\begin{aligned}
F(x) &\subseteq \{f^{\bar{a}_j}(x)\}_{j\in[\omega(x)]} + K \\
&\subseteq \{f^{\bar{a}_j}(x) + \alpha\gamma \operatorname{J} f^{\bar{a}_j}(x)\bar{v}\}_{j\in[\omega(x)]} + K + \operatorname{int}(K) \text{ using (5.19) and Definition 1.29} \\
&\subseteq \{f^{\bar{a}_j}(x+\gamma\bar{v})\}_{j\in[\omega(x)]} + K + K + \operatorname{int}(K) \text{ using (5.16) and Definition 1.26} \\
&\subseteq F(x+\gamma\bar{v}) + \operatorname{int}(K),
\end{aligned}$$

which implies that for all $\gamma \in (0, \bar{\gamma}]$, we have $F(x+\gamma\bar{v}) \prec^l F(x)$, i.e., $\bar{v}$ is a descent direction for F at x. □

Next, we prove that if the sequence $\{x_k\}$ is a convergent sequence generated by Algorithm 2, the the sequence of descent directions $\{v_k\}$ calculated in Step 2 of Algorithm 2 is bounded.

Proposition 5.5 *Let $\{x_k\}$ be a convergent sequence and $\{v_k\}$ be the sequence of projected gradient descent directions generated by Algorithm 2. Then, the sequence $\{v_k\}$ is bounded.*

Proof: From (5.7), for $a^k \in P_{x_k}$ we have

$$\vartheta_x(a^k, v_k) \leq 0 \text{ for all } k \in \mathbb{N}$$

$$\implies \|v_k\|^2 \leq -2 \max_{j\in[\omega_k]}\{\Psi_e(\mathrm{J}\, f^{a_j^k}(x_k)v_k)\} \leq \; 2 \max_{j\in[\omega_k]}\{|\Psi_e(\mathrm{J}\, f^{a_j^k}(x_k)v_k)|\}$$
$$\leq \; 2L \max_{j\in[\omega_k]}\{\|\mathrm{J}\, f^{a_j^k}(x_k)v_k\|\}, \text{ where } L \text{ is a Lipschitz constant of } \Psi_e$$
$$\leq \; 2L\|v_k\| \max_{j\in[\omega_k]}\{\|\mathrm{J}\, f^{a_j^k}(x_k)\|\},$$

which implies that $\|v_k\| \leq 2L \max_{j\in[\omega_k]} \|\mathrm{J}\, f^{a_j^k}(x_k)\|$, and hence $\{v_k\}$ is bounded because $\lim\limits_{k\to\infty} \|\mathrm{J}\, f^{a_j^k}(x_k)\|$ exists for each $j \in [\omega_k]$. □

The convergence of the proposed method is proved under the regularity condition of a point (Definition 4.5) and with the help of essential property of regularity (Lemma 4.4). We use these two from Chapter 4 in the convergence theorem.

First, we prove a result that will be useful further. Thereafter, the global convergence of Algorithm 2 has been proved.

Lemma 5.1 *Under the consideration of Algorithm 2, for all $k = 0, 1, 2, \ldots$, we have*

$$-\gamma\alpha_k \max_{j\in[\omega_k]} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j^k}(x_k)v_k \right) \right\} \leq \min_{y\in F(x_k)} \Psi_e(y) - \min_{z\in F(x_{k+1})} \Psi_e(z).$$

Proof: The proof is similar to Step 1 in proof of Theorem 4.10 in [32]. □

Next, we present the main theorem of this chapter that proves the convergence of the method given in Algorithm 2.

Theorem 5.2 *Suppose that the sequence $\{x_k\}$ is generated by Algorithm 2. Let $\bar{x}$ be an accumulation point of the sequence $\{x_k\}$ which is a regular point for F. Then, $\bar{x}$ is a stationary point of CSOP (1.4).*

Proof: From Lemma 5.1, for all $k = 0, 1, 2, \ldots$, we have

$$-\gamma\alpha_k \max_{j\in[\omega_k]} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j^k}(x_k)v_k \right) \right\} \leq \min_{y\in F(x_k)} \Psi_e(y) - \min_{z\in F(x_{k+1})} \Psi_e(z). \tag{5.20}$$

Note that $\bar{x}$ is an accumulation point of the sequence $\{x_k\}$, and $\{\alpha_k\}$, and $\{v_k\}$ are bounded sequences. Therefore, we can find a subsequence $\mathcal{K}$ in $\mathbb{N}$ such that $\{x_k\}_{k\in\mathcal{K}} \to \bar{x}$, $\{\alpha_k\}_{k\in\mathcal{K}} \to \bar{\alpha}$, and $\{v_k\}_{k\in\mathcal{K}} \to \bar{v}$.

Since the sequence $\{x_k\}_{k\in\mathcal{K}}$ is a convergent sequence, the modulus of the right-hand side of (5.20) approaches to 0 as $k \to \infty$. Therefore, we have □

$$-\gamma \lim_{k\to\infty} \alpha_k \max_{j\in[\omega_k]} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j^k}(x_k) v_k \right) \right\} \leq 0. \tag{5.21}$$

If x_k is not a stationary point, then from (5.2), we have $\mathrm{J}\, f^{a_j^k}(x_k)v_k \in (-\operatorname{int}(K))$ for all $j \in [\omega(x_k)]$ and $k \in \mathcal{K}$. Thus, by using (iv) of Proposition 1.2, for all $j \in [\omega(x_k)]$ and $k \in \mathcal{K}$, we have

$$\Psi_e \left(\mathrm{J}\, f^{a_j^k}(x_k) v_k \right) < 0. \tag{5.22}$$

In view of (5.21) and (5.22), we get

$$\lim_{k\to\infty, k\in\mathcal{K}} \alpha_k \max_{j\in[\omega_k]} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j^k}(x_k) v_k \right) \right\} = 0. \tag{5.23}$$

Note that the number of subsets of $[p]$ is finite and $\bar{x}$ is regular for F. Therefore, without loss of generality by Lemma 4.4, we have

$$\omega_k = \bar{\omega}, P_{x_k} = P' \text{ and } a_k = \bar{a} \in P' \text{ for all } k \in \mathcal{K}. \tag{5.24}$$

We observe that for all $a \in P'$ and $k \in \mathcal{K}$,

$$\phi(x_k) = \vartheta_{x_k}(\bar{a}, v_k) \leq \vartheta_{x_k}(a, v) \text{ for all } v \in \mathcal{S}_{x_k} \implies \vartheta_{\bar{x}}(\bar{a}, \bar{v}) \leq \vartheta_{\bar{x}}(a, v) \text{ as } k \to \infty, \tag{5.25}$$

which is equivalent to

$$(\bar{a}, \bar{v}) \in \operatorname*{argmin}_{(a,v)\in P'\times\mathcal{S}_{\bar{x}}} \vartheta_{\bar{x}}(a, v). \tag{5.26}$$

Now to show that $\bar{x}$ is a stationary point of CSOP (1.4), we prove that $\bar{v} = 0$. For this, we use (5.23). Since $\alpha_k > 0$ for all $k \in \mathcal{K}$, therefore $\bar{\alpha} \geq 0$. Let us consider first that $\bar{\alpha} > 0$. Then, from (5.23) and (5.24), we have

$$\lim_{k\to\infty} \max_{j\in[\bar{\omega}]} \left\{ \Psi_e \left(\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k \right) \right\} = 0. \tag{5.27}$$

Now from (5.10),

$$0 \geq \lim_{k\to\infty} \phi(x_k) \overset{(5.6),(5.9),(5.26)}{=} \lim_{k\to\infty} \left\{ \max_{j\in[\bar{\omega}]} \bar{\beta} \left\{ \Psi_e \left(\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k \right) \right\} + \tfrac{1}{2}\|v_k\|^2 \right\} \overset{(5.27)}{=} \tfrac{1}{2}\|\bar{v}\|^2 \geq 0,$$

which implies that $\bar{v} = 0$. Therefore, using Proposition 5.2, we conclude $\bar{x}$ is a stationary point of CSOP (1.4).

Now we assume that $\bar{\alpha} = 0$, i.e., $\{\alpha_k\}_{k\in\mathcal{K}} \to 0$. Therefore, for a large enough k', we have $\alpha_{k'} < 2^{-q}$ for some fixed but arbitrary positive integer q, which implies that Armijo line search in Step 4 of Algorithm 2 does not satisfy for $\gamma' = 2^{-q}$ at x_k. Thus, there exists a $j \in [\bar{\omega}]$ such that for all $k \in \mathcal{K}$,

$$\begin{aligned} & f^{\bar{a}_j}(x_k + \gamma' v_k) \npreceq f^{\bar{a}_j}(x_k) + \alpha\gamma' \,\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k \\ \Longrightarrow\ & \frac{1}{\gamma'}(f^{\bar{a}_j}(x_k + \gamma' v_k) - f^{\bar{a}_j}(x_k)) - \alpha \,\mathrm{J}\, f^{\bar{a}_j}(x_k) v_k \notin -K \\ \Longrightarrow\ & \tfrac{1}{\gamma'}(f^{\bar{a}_j}(\bar{x} + \gamma' \bar{v}) - f^{\bar{a}_j}(\bar{x})) - \alpha \,\mathrm{J}\, f^{\bar{a}_j}(\bar{x}) \bar{v} \notin -\operatorname{int}(K) \text{ when } k \to \infty. \end{aligned} \tag{5.28}$$

Taking $\gamma' \to 0+$ in (5.28), we obtain that there exists $j \in [\bar{w}]$ such that

$$\begin{aligned} & (1-\alpha)\,\mathrm{J}\, f^{\bar{a}_j}(\bar{x})\bar{v} \notin (-\operatorname{int}(K)) \\ \Longrightarrow\ & \Psi_e(\mathrm{J}\, f^{\bar{a}_j}(\bar{x})\bar{v}) \geq 0 \text{ using (iv) of Proposition 1.2 and } \alpha \in (0,1) \\ \Longrightarrow\ & \vartheta_{\bar{x}}(\bar{a}, \bar{v}) \geq \Psi_e(\mathrm{J}\, f^{\bar{a}_j}(\bar{x})\bar{v}) \geq 0 \\ \Longrightarrow\ & 0 \geq \min_{(a,v)\in P'\times\mathcal{S}_{\bar{x}}} \vartheta_{\bar{x}}(a, v) = \vartheta_{\bar{x}}(\bar{a}, \bar{v}) \geq \Psi_e(\mathrm{J}\, f^{\bar{a}_j}(\bar{x})\bar{v}) \geq 0 \text{ using (5.26) and (5.7)} \\ \Longrightarrow\ & \min_{(a,v)\in P'\times\mathcal{S}_{\bar{x}}} \vartheta_{\bar{x}}(a, v) = 0, \end{aligned}$$

which implies, by using (5.9) and Proposition 5.2, that $\bar{x}$ is a stationary point of CSOP (1.4).

Next, we analyze the convergence of the proposed method with a variable projection parameter in the Step 2 of Algorithm 2. Precisely, we vary $\bar{\beta}$ in (3) at each iteration k and analyze the convergence of the method. We alter $\bar{\beta}$ in the following way: we replace Step 2 of Algorithm 2 as follows.

Step 2: Calculate the projected gradient descent direction

Find $(a^k, v_k) = \underset{(a,v) \in P_k \times \mathcal{S}_{x_k}}{\text{argmin}} \ \vartheta_{x_k}(a, v)$, where $\mathcal{S}_{x_k} = \mathcal{S} - x_k$ and $\vartheta_{x_k} : P_{x_k} \times \mathcal{S}_{x_k} \to \mathbb{R}$ such that

$$\vartheta_{x_k}(a, v) = \max_{j \in [\omega_k]} \frac{\sigma_k}{\xi_k} \left\{ \Psi_e \left(\mathrm{J}\, f^{a_j}(x_k) v \right) \right\} + \tfrac{1}{2}\|v\|^2, \tag{5.29}$$

where $\{\sigma_k\} \subset \mathbb{R}_{++}$ such that $\sum_{k=0}^{\infty} \sigma_k = \infty$ but $\sum_{k=0}^{\infty} \sigma_k^2 < \infty$ and $\xi_k = \max_{j \in [\omega_k]} \|\mathrm{J}\, f^{a_j}(x_k)\|$, and keep rest steps of Algorithm 2 as they are. We call this new algorithm as Algorithm 3.

For Algorithm 3, the results in Propositions 5.2, 5.3 and 5.4 are also true, which can be realized just by pushing $\bar{\beta}$ inside the $\max_{j \in [\omega_k]}$ operator and replacing $\bar{\beta}$ by $\frac{\sigma_k}{\xi_k}$ in all the proofs of these three propositions. Hence, Algorithm 3 is well-defined. The result in Proposition 5.5 has a special form for Algorithm 3, which is proposed in the following result.

Proposition 5.6 *Let the sequences* $\{x_k\}$ *and* $\{v_k\}$ *be generated by Algorithm* 3*. Then, for any given* $a \in P_{x_k}$ *and* $j \in [\omega_k]$*, we have* $\|v_k\| \leq L\sigma_k$*, where* L *is the Lipschitz constant of* Ψ_e*.*

Proof: Note that the function $\vartheta_{x_k}(a, \cdot)$ is convex on $\mathcal{S}_{x_k}$. The first order optimality condition for

$$\min_{v \in \mathcal{S}_{x_k}} \vartheta_{x_k}(a, v) \text{ for all } a \in P_{x_k}$$

□

implies the existence of $g_k \in \partial\vartheta_{x_k}(a, v_k)$ such that

$$\langle g_k, v - v_k\rangle \geq 0 \text{ for all } v \in \mathcal{S}_{x_k}. \tag{5.30}$$

We know that the subdifferential set of a maximum of functions is the convex hull of the union of the subdifferential set of the individual functions (see [104]). Therefore, for any given $a \in P_{x_k}$, the expression of the elements in the subdifferential set $\partial\vartheta_{x_k}(a, v_k)$ gives

$$g_k = v_k + \tfrac{\sigma_k}{\xi_k}\sum_{j=1}^{w_k}\lambda_{jk}h_{jk}, \tag{5.31}$$

for some λ_{jk}'s in $\mathbb{R}_+$ with $\sum_{j\in[w_k]}\lambda_{jk} = 1$ and $h_{jk} \in \partial\Psi_e\left(\mathrm{J}\, f^{a_j}(x_k)v_k\right)$. Since Ψ_e is a Lipschitz function with Lipschitz constant L, for each $j \in [w_k]$, we have

$$\|h_{jk}\| \leq L\|\,\mathrm{J}\, f^{a_j}(x_k)\|. \tag{5.32}$$

Plugging the expression of g_k from (5.31), we get from (5.30) for $v = 0$ that

$$\|v_k\|^2 \leq -\left\langle \tfrac{\sigma_k}{\xi_k}\sum_{j=1}^{w_k}\lambda_{jk}h_{jk}, v_k\right\rangle \leq \tfrac{\sigma_k}{\xi_k}\sum_{j=1}^{w_k}\lambda_{jk}\|h_{jk}\|\|v_k\| \overset{(5.32)}{\leq} \tfrac{\sigma_k}{\xi_k}L\sum_{j=1}^{w_k}\lambda_{jk}\|Jf^{a_j}(x_k)\|\|v_k\|. \tag{5.33}$$

As $\xi_k = \max\limits_{j\in[\omega_k]}\|\mathrm{J}\, f^{a_j}(x_k)\|$, (5.33) yields $\|v_k\| \leq L\sigma_k$, which completes the proof.

Next, we prove an another result that will be used to prove the convergence of Algorithm 3.

Proposition 5.7 *Let the sequences* $\{x_k\}$ *and* $\{v_k\}$ *be generated by Algorithm* 3*. Then, for all* $x \in \mathcal{S}$ *and* $a \in P_{x_k}$*, we have*

$$\|x_{k+1}-x\|^2 \leq \|x_k-x\|^2+3L^2\sigma_k^2-2\tfrac{\alpha_k\sigma_k}{\xi_k}\max_{j\in[w_k]}(\Psi_e(\mathrm{J}\, f^{a_j}(x_k)v_k)-\Psi_e(\mathrm{J}\, f^{a_j}(x)v_k))). \tag{5.34}$$

where L is the Lipschitz constant of Ψ_e.

Proof: By Proposition 5.6, we have

$$\begin{aligned} L^2\sigma_k^2\alpha_k^2 + \|x_k - x\|^2 - \|x_{k+1} - x\|^2 \geq\ & \|x_{k+1} - x_k\|^2 + \|x_k - x\|^2 - \|x_{k+1} - x\|^2 \\ =\ & 2\langle x_k - x_{k+1}, x_k - x\rangle = 2\alpha_k\langle v_k, x - x_k\rangle. \quad (5.35) \end{aligned}$$

From (5.31), there exists $g_k \in \partial\vartheta_{x_k}(a, v_k)$ such that for all $v \in \mathcal{S}_{x_k}$, we have

$$\langle g_k, v - v_k\rangle \geq 0 \overset{(5.31)}{\Longrightarrow} \left\langle v_k + \frac{\sigma_k}{\xi_k}\sum_{j=1}^{w_k}\lambda_{jk}h_{jk}, v - v_k\right\rangle \geq 0 \quad (5.36)$$

In particular, taking $v = x - x_k$, and denoting $d_k = \frac{\sigma_k}{\xi_k}\sum_{j=1}^{w_k}\lambda_{jk}h_{jk}$ we get from (5.36) that

$$\begin{aligned} & \langle v_k + d_k, x - x_k - v_k\rangle \geq 0 \\ \Longrightarrow\ & \langle v_k, x - x_k - v_k\rangle + \langle d_k, x - x_k - v_k\rangle \geq 0 \\ \Longrightarrow\ & \langle v_k, x - x_k\rangle \geq \|v_k\|^2 - \langle d_k, x - x_k - v_k\rangle. \quad (5.37) \end{aligned}$$

Therefore, (5.35) and (5.37) yield

$$\begin{aligned} L^2\sigma_k^2\alpha_k^2 + \|x_k - x\|^2 - \|x_{k+1} - x\|^2 & \geq 2\alpha_k\|v_k\|^2 - 2\alpha_k\langle d_k, x - x_k\rangle + 2\alpha_k\langle d_k, v_k\rangle \\ & \geq 2\alpha_k\langle d_k, v_k\rangle - 2\alpha_k\langle d_k, x - x_k\rangle \\ & \geq -2\alpha_k\|d_k\|\|v_k\| - 2\alpha_k\langle d_k, x - x_k\rangle \\ & \geq -2\alpha_k\|d_k\|L\sigma_k - 2\alpha_k\langle d_k, x - x_k\rangle \\ & \qquad\qquad \text{by Proposition 5.6} \\ & \overset{(5.31)\&(5.32)}{\geq} -2\alpha_k L^2\sigma_k^2 - 2\alpha_k\langle d_k, x - x_k\rangle, \quad (5.38) \end{aligned}$$

which implies

$$\begin{aligned} & (2\alpha_k + \alpha_k^2)L^2\sigma_k^2 + \|x_k - x\|^2 - \|x_{k+1} - x\|^2 \\ \geq \quad & -2\alpha_k \tfrac{\sigma_k}{\xi_k} \left\langle \sum_{j=1}^{w_k} \lambda_{jk} h_{jk}, x - x_k \right\rangle \\ \geq \quad & 2\tfrac{\alpha_k\sigma_k}{\xi_k} \sum_{j=1}^{\omega_k} \lambda_{jk} \Psi_e(\mathrm{J}\, f^{a_j}(x_k)v_k) - \Psi_e(\mathrm{J}\, f^{a_j}(x)v_k)) \text{ because } h_{jk} \in \partial\Psi_e \left(\mathrm{J}\, f^{a_j}(x_k)v_k\right) \\ \geq \quad & 2\tfrac{\alpha_k\sigma_k}{\xi_k} \max_{j\in[w_k]} (\Psi_e(\mathrm{J}\, f^{a_j}(x_k)v_k) - \Psi_e(\mathrm{J}\, f^{a_j}(x)v_k))). \end{aligned} \tag{5.39}$$

Hence, as $0 < \alpha_k \leq 1$ for any k, we get from (5.39) that

$$\|x_{k+1} - x\|^2 \leq \|x_k - x\|^2 + 3L^2\sigma_k^2 - 2\tfrac{\alpha_k\sigma_k}{\xi_k} \max_{j\in[w_k]} (\Psi_e(\mathrm{J}\, f^{a_j}(x_k)v_k) - \Psi_e(\mathrm{J}\, f^{a_j}(x)v_k))).$$

Hence, the result follows. □

The next theorem shows that under nonemptiness conditions, any sequence generated by Algorithm 3 is convergent.

Theorem 5.3 *Let $\{x_k\}$ and $\{v_k\}$ be the sequences generated by Algorithm 3. If the set*

$$V = \{x \in \mathcal{S} : \mathrm{J}\, f^{a_j}(x)v_k \preceq \mathrm{J}\, f^{a_j}(x_k)v_k \ \forall\ k = 0, 1, 2, \ldots\}$$

is nonempty, then $\{x_k\}$ converges to some $\hat{x} \in V$.

Proof: For any point $x \in V$, we have $\Psi_e(\mathrm{J}\, f^{a_j}(x)v_k) \leq \Psi_e(\mathrm{J}\, f^{a_j}(x_k)v_k))$ for all $k = 0, 1, 2, \ldots$. So, from Proposition 5.7, we get for any $k = 0, 1, 2, \ldots$ that

$$\begin{aligned} \|x_{k+1} - x\|^2 \leq \ & \|x_k - x\|^2 + 3L^2\sigma_k^2 \\ \leq \ & \|x_{k-1} - x\|^2 + 3L^2(\sigma_k^2 + \sigma_{k-1}^2) \\ \leq \ & \cdots \ \leq \ \|x_0 - x\|^2 + 3L^2 \sum_{j=0}^{k} \sigma_k^2. \end{aligned} \tag{5.40}$$

Thus, as $\sum_{j=0}^{\infty} \sigma_k^2$ exists, the sequence $\{x_k\}$ is bounded. So, $\{x_k\}$ must have an accumulation point. Let $\hat{x}$ be an accumulation point of $\{x_k\}$. As $\hat{x}$ is an accumulation point of $\{x_k\}$, there exists an infinite subset $\mathcal{K}$ of $\mathbb{N}$ such that $\lim\limits_{k\to\infty, k\in\mathcal{K}} x_k = \hat{x}$. In all the next steps till (5.42), we take k only in $\mathcal{K}$.

In analogy to the lines of the proof of Theorem 5.2 from the beginning till equation (5.22), we get that

$$\Psi_e \left(\mathrm{J}\, f^{a_j^k}(x_k) v_k \right) < 0 \text{ for all } j \in [w_k].$$

Therefore, for any $j \in [\omega_k]$, we have

$$\begin{aligned} & \mathrm{J}\, f^{a_j^k}(x_k) v_k \in -K \\ \implies\ & f^{a_j^k}(x_{k+1}) \preceq f^{a_j^k}(x_k) + \alpha_k 2^{-t_k}\, \mathrm{J}\, f^{a_j^k}(x_k) v_k \prec f^{a_j^k}(x_k), \end{aligned} \tag{5.41}$$

which yields

$$\begin{aligned} F(x_k) &\subseteq \{f^{a_j^k}(x_k)\}_{j\in[w_k]} + K + \mathrm{int}(K) \\ &\subseteq \{f^{a_j^k}(x_k)\}_{j\in[w_k]} + K \\ &\subseteq \{f^{a_j^k}(x_{k+1})\}_{j\in[w_k]} + K + K + \mathrm{int}(K) \\ &\subseteq F(x_{k+1}) + \mathrm{int}(K). \end{aligned}$$

Therefore, $F(x_{k+1}) \prec^l F(x_k)$, and hence

$$F(\hat{x}) \prec^l \cdots \prec^l F(x_{k+3}) \prec^l F(x_{k+2}) \prec^l F(x_{k+1}) \prec^l F(x_k) \text{ for all } k \in \mathbb{N}. \tag{5.42}$$

Thus, $\hat{x} \in V$. We show that $\{x_k\}$ converges to $\hat{x}$ (not for $k \in \mathcal{K}$).

Let $\varepsilon > 0$. Since $\sum_{k=0}^{\infty} \sigma_k^2$ is convergent, there exists $k_1 \in \mathbb{N}$ such that $\sum_{k_1}^{\infty} \sigma_k^2 < \frac{\varepsilon}{2}$. As $\lim\limits_{k\to\infty, k\in\mathcal{K}} x_k = \hat{x}$, there exists $k_2 \in \mathcal{K}$ with $k_2 \geq k_1$ such that $\|x_{k_2} - \hat{x}\|^2 < \frac{\varepsilon}{2}$. Therefore, for any $k \geq k_2$, $k \in \mathbb{N}$, we have

$$\|x_k - \hat{x}\|^2 \overset{(5.40)}{\leq} \|x_{k_2} - \hat{x}\|^2 + \sum_{j=k_2}^{k-1} \sigma_j^2 \leq \tfrac{\varepsilon}{2} + \sum_{j=k_1}^{\infty} \sigma_k^2 < \varepsilon.$$

Hence, $\{x_k\}$ converges to $\hat{x} \in V$, and the result follows. □

5.6 Numerical results

In this section, we consider some numerical examples. These examples are solved by the proposed Algorithm 2. In our implementation, we take a standard cone, i.e., $K = \mathbb{R}^m_+$ in all the instances except for Example 5.8, and Gerstewitz function parameter $e = (1, 1, \ldots, 1)^\top \in \text{int}(K)$. The parameters $\bar{\beta}$ in Step 2 and α in Step 4 are taken as 1.5000 and 0.0001, respectively. We start with the step length $\alpha_k = 1$ in all the instances. Subsequently, 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 in a Windows 10 machine equipped with a 1.90 GHz CPU with 8 GB RAM.

To find $\text{Min}(F(x_k), K)$ in Step 1, we follow the crude way of comparing the elements in $F(x_k)$ pair-wise. To calculate $(a^k, v_k) = \underset{(a,v) \in P_{x_k} \times \mathcal{S}_{x_k}}{\text{argmin}} \vartheta_{x_k}(a, v)$ in Step 2, we use an inbuilt function *fmincon* in MATLAB.

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 three-column table. In the table, the first column indicates the number of initial points that are solved by the algorithm. The second 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 third 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 $\bullet, +, \triangle, *, \star$, and $\square$ 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 algorithm 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 for each other shapes ($+, \triangle, *, \star$, and $\square$) also.

We consider CSOP (1.4) in the form of the following examples. In these examples, F and $\mathcal{S}$ are the objective function and nonempty closed, convex set considered in CSOP (1.4), respectively. We consider the first example from [132], which is as follows.

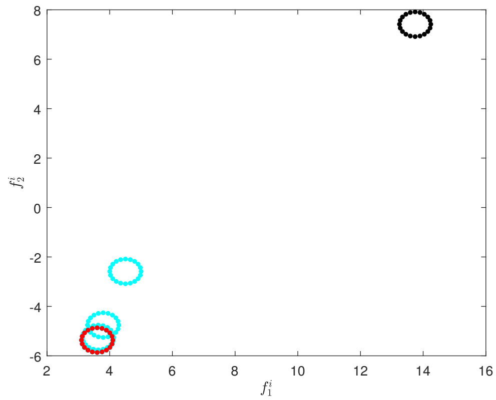
Example 5.1 *Let* $F : \mathbb{R}^2 \rightrightarrows \mathbb{R}^2$ *be the set-valued map defined by*

$$F(x) = \left\{ f^1(x), f^2(x), \ldots, f^{20}(x) \right\},$$

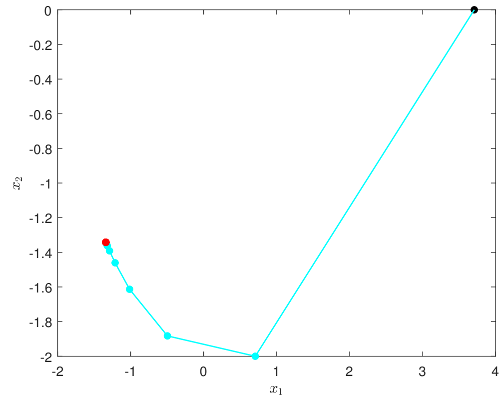
where f^i*, for* $i \in [20]$*, are from* $\mathbb{R}^2$ *to* $\mathbb{R}^2$ *as below*

$$f^i(x) = \begin{pmatrix} x_1^2 + x_2^2 + 0.5 \sin\left(\frac{2\pi(i-1)}{20}\right) \\ 2(x_1 + x_2) + 0.5 \cos\left(\frac{2\pi(i-1)}{20}\right) \end{pmatrix}$$

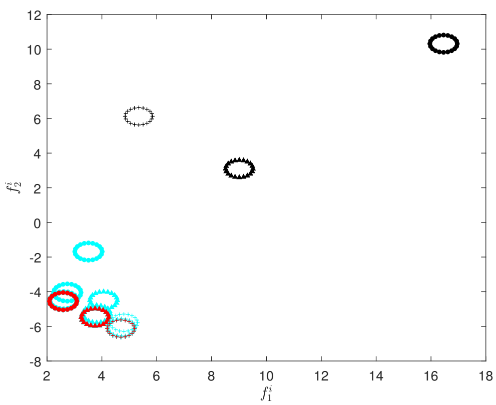
and $\mathcal{S} = [-2, 4] \times [-2, 4]$.



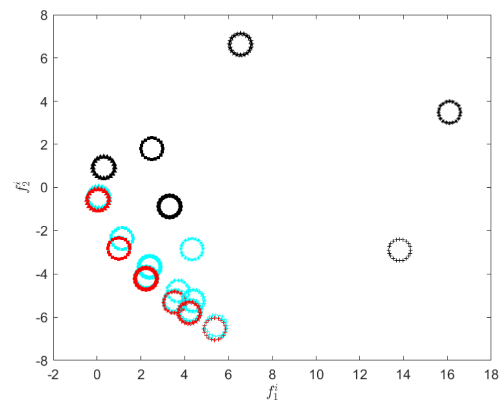
(a) The value of F at each iteration generated by Algorithm 2 for Example 5.1 for initial point $x_0 = (3.7073, 0)^\top$



(b) The value of x_k at each iteration generated by Algorithm 2 for Example 5.1 for initial point $x_0 = (3.7073, 0)^\top$



(c) The value of F at each iteration generated by Algorithm 2 for Example 5.1 for three different randomly chosen initial points



(d) The value of F at each iteration generated by Algorithm 2 for Example 5.1 for six different randomly chosen initial points

Figure 5.1: Output of Algorithm 2 for Example 5.1

The performance of the proposed Algorithm 2 for Example 5.1 is shown in the following table.

Table 5.1: Performance of Algorithm 2 on Example 5.1

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	$(2, 11, 6.8800, 6, 10, 2.6064)$	$(3.8976, 19.8335, 11.4384, 10.2195, 12, 4.0429)$

The next problem is an example considered in [84].

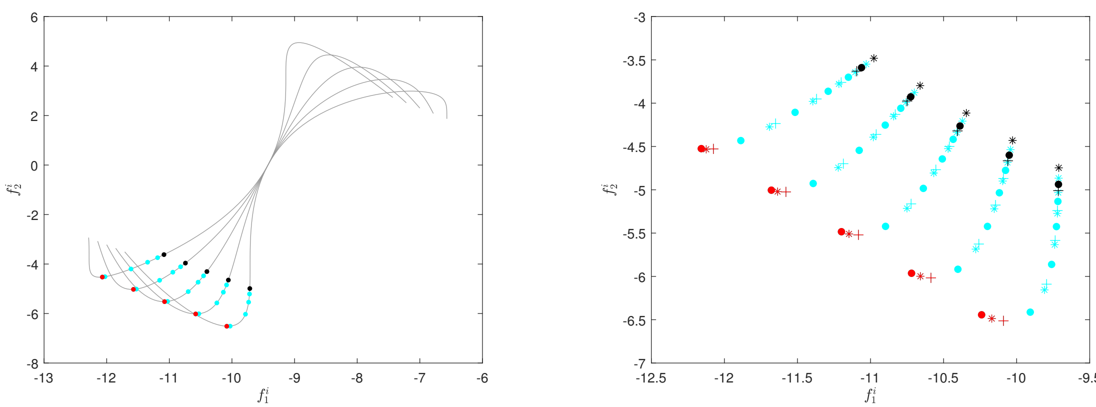
Example 5.2 *Let $F : \mathbb{R} \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

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

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

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

with $\mathcal{S} = [-5\pi, 5\pi]$.



(a) The value of F at intermediate iterates generated by Algorithm 2 for Example 5.2 for the initial point $x_0 = -10.4000$

(b) The value of F at each iteration generated by Algorithm 2 for Example 5.2 for three different randomly chosen initial points

Figure 5.2: Output of Algorithm 2 for Example 5.2

The performance of the proposed Algorithm 2 for Example 5.2 is shown in the following table.

Table 5.2: Performance of Algorithm 2 on Example 5.2

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	$(0, 18, 2.8800, 0, 0, 5.1233)$	$(0.1808, 5.7824, 1.0665, 0.3069, 1, 1.4286)$

The next two examples (Example 5.3 and Example 5.4) are freshly introduced in this chapter. In Example 5.3, we consider two variables and 14 functions with three component functions. While in Example 5.4, we consider one variable and 40 functions with two component functions.

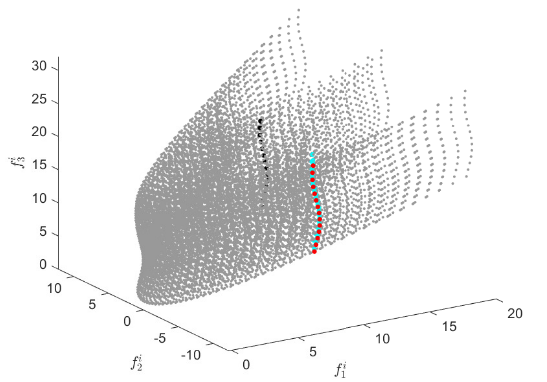
Example 5.3 *Let* $F : \mathbb{R}^2 \rightrightarrows \mathbb{R}^3$ *be a set-valued map defined as*

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

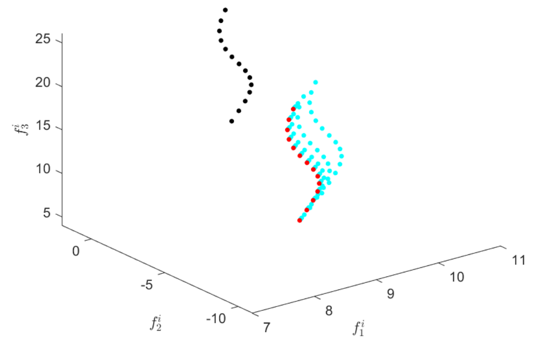
where f^i, *for* $i \in [14]$, *are from* $\mathbb{R}^2$ *to* $\mathbb{R}^3$, *defined by*

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

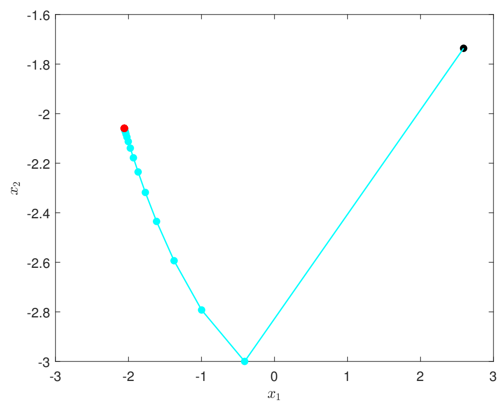
with $\mathcal{S} = [-3, 3] \times [-3, 3]$.



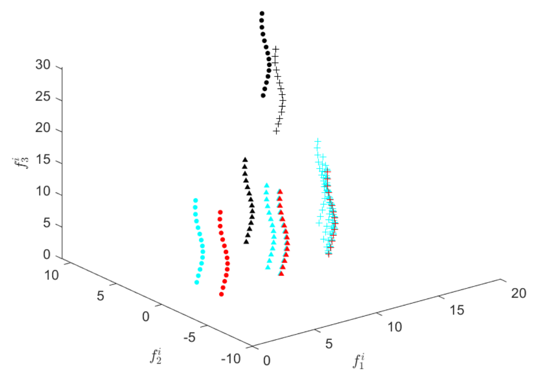
(a) The value of F at each iteration generated by Algorithm 2 for Example 5.3 for initial point $x_0 = (2.5909, -1.7362)^\top$



(b) The value of F at each iteration generated by Algorithm 2 for Example 5.3 for initial point $x_0 = (2.5909, -1.7362)^\top$



(c) The value of x_k at each iteration generated by Algorithm 2 for Example 5.3 for initial point $x_0 = (2.5909, -1.7362)^\top$



(d) The value of F at each iteration generated by Algorithm 2 for Example 5.3 for three different randomly chosen initial points

Figure 5.3: Output of Algorithm 2 for Example 5.3

The performance of the proposed Algorithm 2 for Example 5.3 is shown in the following table.

Table 5.3: Performance of Algorithm 2 on Example 5.3

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	$(2, 38, 9.7100, 8, 5, 6.0223)$	$(5.6643, 84.7053, 24.7024, 17.1115, 10, 15.7636)$

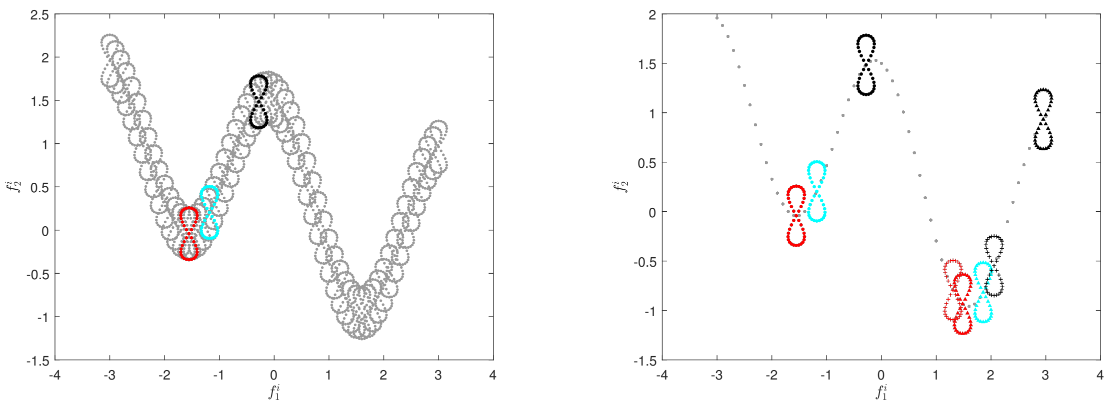
Example 5.4 *Let $F : \mathbb{R} \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

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

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

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

with $\mathcal{S} = [-3, 3]$.



(a) The value of F at intermediate iterates generated by Algorithm 2 for Example 5.4 for the initial point $x_0 = -0.2800$

(b) The value of F at each iteration generated by Algorithm 2 for Example 5.4 for three different randomly chosen initial points

Figure 5.4: Output of Algorithm 2 for Example 5.4

The performance of the proposed Algorithm 2 for Example 5.4 is shown in the following table.

Table 5.4: Performance of Algorithm 2 on Example 5.4

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	$(0, 2, 0.5400, 0, 0, 0.5932)$	$(0.2356, 1.9284, 0.6261, 0.5443, 1, 0.3817)$

The following example 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 5.5.

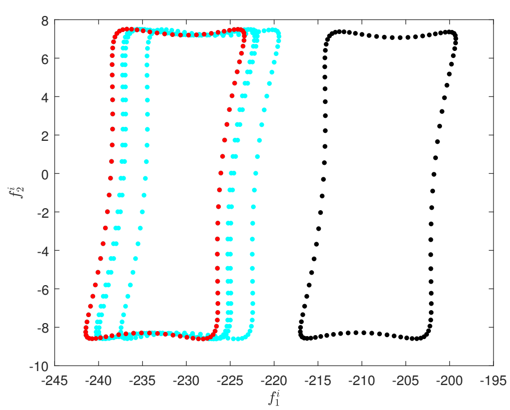
Example 5.5 *Let $F : \mathbb{R}^2 \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

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

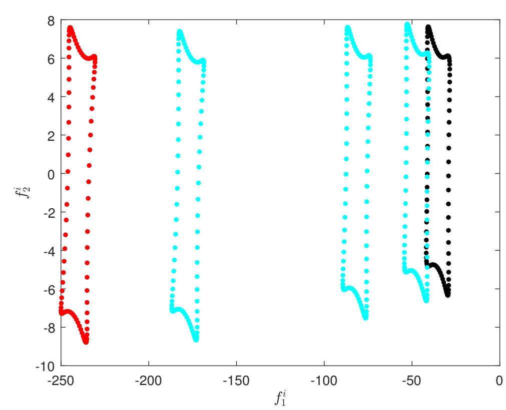
where f^i, for $i \in [100]$, are from $\mathbb{R}^2$ to $\mathbb{R}^2$ as below

$$f^i(x) = \begin{pmatrix} e^{\frac{x_1}{2}} \cos x_2 + x_1 \cos x_2 \sin^3\left(\frac{2\pi(i-1)}{100}\right) - x_2 \sin x_2 \cos\left(\frac{2\pi(i-1)}{100}\right) \\ e^{\frac{x_2}{20}} \sin x_1 + x_1 \cos x_2 \sin\left(\frac{2\pi(i-1)}{100}\right) + x_2 \cos x_2 \cos^3\left(\frac{2\pi(i-1)}{100}\right) \end{pmatrix}$$

with $\mathcal{S} = [-5\pi, 5\pi] \times [-5\pi, 5\pi]$.



(a) The value of F at each iteration generated by Algorithm 2 for Example 5.5 for initial point $x_0 = (11.3552, -8.6467)^\top$



(b) The value of F at each iteration generated by Algorithm 2 for Example 5.5 for initial point $x_0 = (7.8000, -8.6467)^\top$

Figure 5.5: Output of Algorithm 2 for Example 5.5

The performance of the proposed Algorithm 2 for Example 5.5 is shown in the following table.

Table 5.5: Performance of Algorithm 2 on Example 5.5

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
94	$(0, 74, 8.7872, 3, 0, 15.1359)$	$(1, 2.3562e+03, 109.6872, 9.2000, 3, 309.8197)$

In this example, an extra table (Table 5.6) is also added. In which the decrement in the values of vector-valued functions has been shown at each iteration.

Table 5.6: Solution found in the argument space for Example 5.5

Iteration number (k)	$x_k^\top$	$f^{25}(x_k)^\top$	$f^{50}(x_k)^\top$	$f^{75}(x_k)^\top$	$f^{100}(x_k)^\top$
0	(7.8000, -8.6467)	(-41.0915, -4.8145)	(-29.1317, -5.8181)	(-29.2837, 6.1106)	(-41.2435, 7.1142)
1	(8.3437, -8.6713)	(-53.7084, -5.1236)	(-41.3672, -6.0733)	(-40.8647, 6.2676)	(-53.2059, 7.2173)
2	(9.3552, -8.7150)	(-88.9583, -6.0380)	(-75.8813, -6.9091)	(-74.1372, 6.1279)	(-87.2141, 6.9990)
3	(10.8147, -8.7758)	(-186.5895, -7.1559)	(-172.4001, -7.9950)	(-168.7934, 5.8873)	(-182.9828, 6.7264)
4	(11.3874, -8.7965)	(-249.7640, -7.2729)	(-235.1233, -8.0905)	(-230.7974, 6.0822)	(-245.4381 6.8999)

The next problem is the robust counterpart of the vector-valued facility location problem under uncertainty [110]. For detailed information on this problem, one may refer to [32].

Example 5.6 *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*

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

with $\mathcal{S} = [-50, 50] \times [-50, 50]$, *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* $\{u_i = (u_{1i}, u_{2i})^\top : i \in [100]\}$ *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 5.6, 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, intermediate points, and final points generated by Algorithm 2 are denoted by black, cyan, and red colors, respectively, in Figure 5.6.*

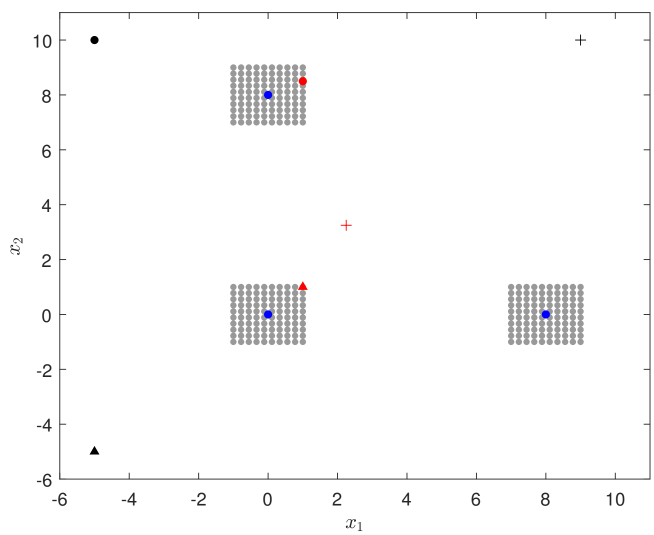


Figure 5.6: The value of x_k for three different initial points ($(-5, -5)^\top, (-5, 10)^\top$ and $(9, 10)^\top$) at each iteration generated by Algorithm 2 for Example 5.6

The performance of the proposed Algorithm 2 for Example 5.6 is shown in the following table.

Table 5.7: Performance of Algorithm 2 on Example 5.6

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	$(0, 5, 2, 2, 2, 0.7107)$	$(2.0461, 49.9550, 10.2301, 9.0698, 14, 7.5582)$

Example 5.7 *Let $F : \mathbb{R}^3 \rightrightarrows \mathbb{R}^3$ be a set-valued map defined as*

$$F(x_1, x_2, x_3) = \{f^1(x_1, x_2, x_3), f^2(x_1, x_2, x_3), \ldots, f^{64}(x_1, x_2, x_3)\},$$

where for each $i \in [64]$, $f^i : \mathbb{R}^3 \to \mathbb{R}^3$ is given by

$$f^i(x_1, x_2, x_3) = h(x_1, x_2, x_3) + \tfrac{1}{16}\begin{pmatrix} \cos\phi_i \\ \cos\psi_i \sin\phi_i \\ \sin\psi_i \sin\phi_i \end{pmatrix}$$

with

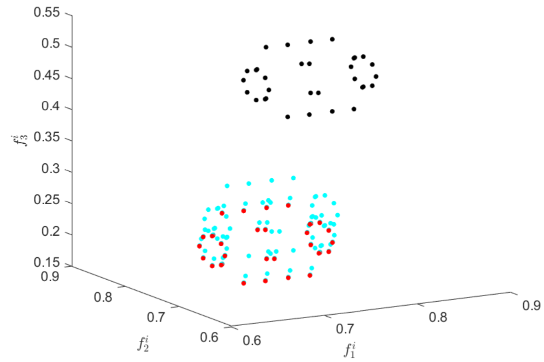
$$h(x_1, x_2, x_3) = \begin{bmatrix} (1 + g(x_3)) \cos u(x_1) \cos v(x_1, x_2, x_3) \\ (1 + g(x_3)) \cos u(x_1) \sin v(x_1, x_2, x_3) \\ (1 + g(x_3)) \sin u(x_1) \end{bmatrix},$$

$$g(x_3) = (x_3 - \tfrac{1}{2})^2,\ u(x_1) = \frac{\pi x_1}{2},\ v(x_1, x_2, x_3) = \frac{\pi(1 + 2g(x_3)x_2)}{4\left(1 + g(x_3)\right)}$$

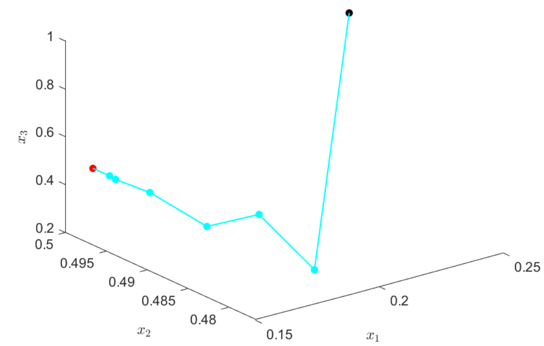
and the set $\{(\phi_i, \psi_i) : i \in [64]\}$ is an enumeration of the set

$$\{-\pi + \tfrac{2\pi}{7}(j-1) : j \in [8]\} \times \{\tfrac{2\pi}{7}(\ell - 1) : \ell \in [8]\}$$

and the set $\mathcal{S} = [0,1] \times [0,1] \times [0,1]$.



(a) The value of F at each iteration generated by Algorithm 2 for Example 5.7 for initial point $x_0 = (0.2434, 0.4936, 0.9549)^\top$



(b) The value of x_k at each iteration generated by Algorithm 2 for Example 5.7 for initial point $x_0 = (0.2434, 0.4936, 0.9549)^\top$

Figure 5.7: Output of the Algorithm 2 for Example 5.7

The performance of the proposed Algorithm 2 for Example 5.7 is shown in the following table.

Table 5.8: Performance of Algorithm 2 on Example 5.7

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
80	$(2, 79, 10.7625, 9, 7, 10.1186)$	$(4.4646, 499.6408, 34.0373, 19.5207, 20, 67.4957)$

In the next example, we take a cone which different from the standard cone $\mathbb{R}^m_+$. We take cone $K' = \{(y_1, y_2)^\top \in \mathbb{R}^2 : 4y_1 - y_2 \geq 0 \text{ and } -4y_1 + 5y_2 \geq 0\}$ and keep the other parameters same.

Example 5.8 *Let $F : \mathbb{R} \rightrightarrows \mathbb{R}^2$ be a set-valued map defined as*

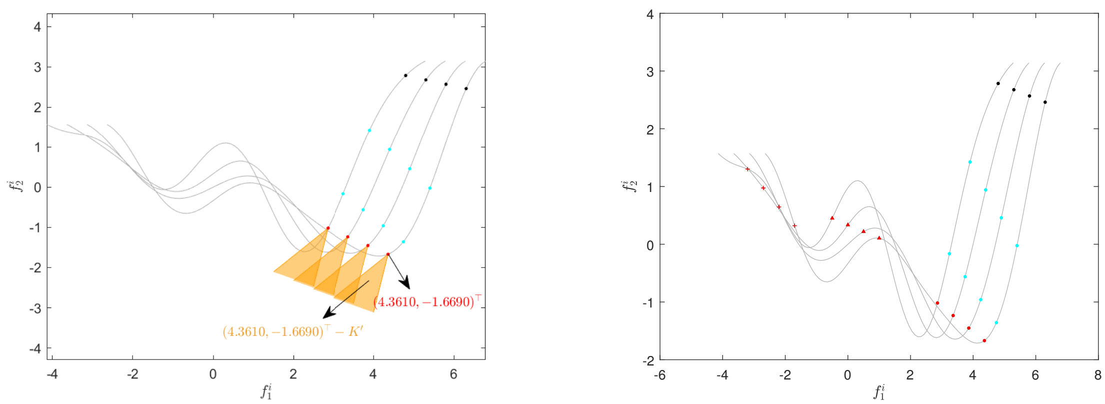
$$F(x) = \left\{ f^1(x), f^2(x), f^3(x), f^4(x) \right\},$$

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

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

with $\mathcal{S} = [-\pi, 2\pi]$*.*

In Figure 5.8(a), it can be easily seen that the red points are the optimal value of F *as at the set* $(4.4140, -1.6450)^\top - K'$ *does not have any element of* $F(x)$ *other than* $(4.4140, -1.6450)^\top$ *for all* $x \in [-\pi, 2\pi]$*. In Figure 5.8(b), we have chosen three initial points 5.8, 0.5, and* -2.2 *and the value of* F *at these three points are denoted by* $\bullet$*,* $\triangle$ *and* $+$*, respectively. The points 0.5 and* -2.2 *are the stationary points because at these points Algorithm 2 stops in 0-th iteration. However, the initial point 5.8 takes 6 iterations to reach the stationary point, which can be seen in figure (b) of Figure 5.8.*



(a) The value of F at intermediate iterative points generated by Algorithm 2 for Example 5.8 for the initial point $x_0 = 5.8$

(b) The value of F at each iteration generated by Algorithm 2 for Example 5.8 for three different initial points 5.8, 0.5, and -2.2

Figure 5.8: Output of the Algorithm 2 for Example 5.8

The performance of the proposed Algorithm 2 for Example 5.2 is shown in the following table.

Table 5.9: Performance of Algorithm 2 on Example 5.8

Number of initial points	Iterations (Min, Max, Mean, Median, Mode, Std.D)	CPU time (Min, Max, Mean, Median, ⌈Mode⌉, Std.D)
100	$(0, 4, 0.4200, 0, 0, 0.9763)$	$(17.2467, 48.2244, 30.9220, 30.3065, 28, 6.7912)$

Below, we give a brief discussion of the numerical implementation of the considered examples.

We ran the proposed algorithm for different types of problems. Some CSOPs are with objective functions of one variable (Examples 5.2, 5.4, and 5.8), and some are with objective functions of two or three variables (Examples 5.1, 5.3, 5.5, 5.6, and 5.7). We have run the algorithm for 100 randomly chosen initial points for each problem. The algorithm stops at stationary points within 100 iterations for all 100 randomly chosen initial points successfully for all examples except Examples 5.5 and 5.7. The algorithm stopped for 94 initial points in Example 5.5 and for 80 initial points in Example 5.7 within 100 iterations. However, it has been observed that most of the unstopped initial points were very close to satisfying the stopping condition when maximum iteration was reached. Thus, perhaps only a few more iterations were needed to reach the stationary point. It also has been observed that the algorithm stopped for most of the initial points in only 2 iterations in Examples 5.6. For a very few initial points, the algorithm takes more than 2 iterations. Similarly, in Example 5.4, the algorithm stopped for most of the initial points in only one iteration. In Example 5.8, we have considered a cone different from standard cone $\mathbb{R}^2_+$. This makes the numerical implementation for this particular example tougher. The algorithm stopped in 2 or 3 iterations for most of the initial points for Example 5.8. The maximum iterations, taken by the algorithm in Examples 5.7 and 5.5, are much more compared to other considered examples. However, the mean of the iterations in the associated tables for these two examples is not much. It shows that the algorithm works well for Examples 5.5 and 5.7 also.

5.7 Conclusion

In this chapter, we have proposed a projected gradient method for constrained set-valued optimization problems with the objective function being a collection of finitely many continuously differentiable vector-valued functions. Two variants of the method—with constant projection parameter $\bar{\beta}$ (Algorithm 2) and variable projection parameter $\frac{\sigma_k}{\xi_k}$ (Algorithm 3)—are analysed. Well-definedness and global convergence of both the

Algorithms 2 and 3 are provided. It is found that the global convergence of the algorithms does not require any convexity assumption on the objective function.

To derive the methods, some optimality conditions for these problems are derived (Proposition 5.1 and Theorem 5.1). The feasibility of the generated points (Proposition 5.3) by the proposed methods is proved. It is found that the sequence of projected gradient descent direction generated by the algorithm is bounded (Proposition 5.5). Some numerical examples are solved to exhibit the performance of the proposed method.

We have proved global convergence of both Algorithms 2 and 3. However, proving the global convergence was not an easy task because P_k is commonly different than P_{k+1}. In Proposition 5.5, the boundedness of $\{v_k\}$ (when v_k is calculated by Algorithm 2) has been proved. A similar result for Algorithm 3 has been proved in the form of Proposition 5.6. However, to prove the inequality $\|v_k\| \leq L\sigma_k$ in Proposition 5.6 was not a straightforward task (comparatively to Proposition 5.5). To prove this, the notion of subdifferentiability has been used as the maximum of differentiable functions may not be differentiable, which was challenging to handle. Specifically, in the proof of Proposition 5.6, we have used the first-order optimality condition for

$$\min_{v \in \mathcal{S}_{x_k}} \vartheta_{x_k}(a, v) \text{ for all } a \in P_{x_k}.$$

As a consequence, we have considered a $g_k \in \partial\vartheta_{x_k}(a, v_k)$ such that (5.30) is satisfied. Then, we have calculated the exact expression of g_k, which is given in (5.31). Subsequently, using some manipulations and calculations, we have proved the inequality $\|v_k\| \leq L\sigma_k$. However, the analysis was not so difficult in Proposition 5.5. Similar kinds of challenges have also been faced in proving Proposition 5.7. Inequalities (5.36), (5.37), and (5.39) have been established by the analysis and calculations using $g_k \in \partial\vartheta_{x_k}(a, v_k)$, which was a bit tricky. Using these two propositions, the global convergence of Algorithm 3 has been proved under the nonemptiness conditions without assuming the convexity assumption on the objective function.

In the numerical implementation, the most challenging task was to find the projected gradient descent direction v_k in Step 2. To calculate v_k, we needed to solve (5.12), which includes the Gerstewitz functional Ψ_e. To calculate the value of $\Psi_e(y)$, where $y \in \mathbb{R}^m$, with respect to the standard cone $\mathbb{R}^m_+$, is not difficult as it is equal to the maximum of all the components of y when $e = (1, 1, \ldots, 1)$; however, it is not an easy task when the cone is different (as we have taken in Example 5.8) from the standard cone $\mathbb{R}^m_+$.

To calculate the partition set P_k, we have directly used (ii) of Definition 5.1 and Definition 5.3. In (ii) of Definition 5.1, the set of indices with respect to each element of the enumeration of $\text{Min}(F(x_k), K)$ is calculated and then using the Cartesian product of the set of indices of each element of the enumeration of $\text{Min}(F(x_k), K)$ (as given in Definition 5.3) P_k is calculated. We have calculated $\text{Min}(F(x_k), K)$ by comparing the elements of $F(x_k)$.

****