J. Appl. Numer. Optim. 2 (2020), No. 3, pp. 249-277
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.2.2020.3.01

PROJECTION-TYPE METHODS WITH ALTERNATING INERTIAL STEPS FOR SOLVING MULTIVALUED VARIATIONAL INEQUALITIES BEYOND MONOTONICITY

CHINEDU IZUCHUKWU[1,2], YEKINI SHEHU[1,*]

[1]*Department of Mathematics, Zhejiang Normal University, Jinhua 321004, China*
[2]*School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, South Africa*

Abstract. In solving variational inequalities, the inertial extrapolation step is a highly powerful tool in algorithmic designs and analyses mainly due to the improved convergence speed that it contributes to the algorithms. However, it has been discovered that the presence of the inertial extrapolation steps in these methods for solving variational inequalities makes them lose some of their attractive properties, for example, the Fejér monotonicity (with respect to the solution set) of the sequence generated by projection-type methods for solving variational inequalities is lost when the iterative steps involve an inertial term, which makes these methods sometimes not converge faster than the corresponding algorithms without an inertial term. To avoid such a situation, we present two new projection-type methods with alternated inertial extrapolation steps for solving multivalued variational inequality problems, which inherit the Fejér monotonicity property of the projection-type method to some extent. Furthermore, we prove the convergence of the sequence generated by our methods under much relaxed assumptions on the inertial extrapolation factor and the multivalued mapping associated with the problem. Moreover, we establish the convergence rate of our methods and provide several numerical experiments of the new methods in comparison with other related methods in the literature.

Keywords. Alternating inertial steps; Armijo-type linesearch; Convergence rate; Multivalued variational inequalities; projection-type methods.

1. Introduction

In this paper, we consider the following Multivalued Variational Inequality Problem (MVIP): Find $x^* \in C$ and $u \in A(x^*)$ such that

$$\langle u, y - x^* \rangle \geq 0, \ \forall y \in C, \tag{1.1}$$

where C is a nonempty closed and convex set in $\mathbb{R}^m$, $A : C \rightrightarrows \mathbb{R}^m$ is a multivalued mapping with nonempty values, and $\langle ., . \rangle$ denotes the inner product in $\mathbb{R}^m$. The MVIP, which was first studied by Fang and Peterson [18], is important and useful generalization of the classical Variational Inequality Problem (VIP). The MVIP is also a useful generalization of the class of multivalued complementarity problems (see [12, 14, 25]), as well as constrained convex non-smooth

[*] Corresponding author.

E-mail addresses: izuchukwu_c@yahoo.com, izuchukwuc@ukzn.ac.za (C. Izuchukwu), yekini.shehu@unn.edu.ng (Y. Shehu).

Received November 26, 2020; Accepted December 19, 2020.

optimization problems (see [6, 12, 25, 40, 46]). Therefore, problem (1.1) is quite general and provides a unified treatment for the study of a wide class of problems, such as, price equilibrium problems, oligopolistic market equilibrium problems, Nash equilibrium problems, fixed point problems for multivalued mappings, game theory, and others (see [2, 9, 25, 38] and the references therein).

If A is a singlevalued mapping in (1.1), them the MVIP reduces to the classical VIP, and many methods have been developed in the literature to solve the classical VIP. For example, the gradient projection methods, extragradient methods [28], the subgradient extragradient methods [7], Tseng's methods [45], projection and contraction methods [24, 43], and many others. However, these methods are not easily transformed to the case of the MVIP since it is difficult to handle the multivalued mappings associated with the MVIP. Therefore, the methods for solving the MVIP are quite different and some of them have been extensively studied in the literature. For instance, Li and He [29] extended the projection method to solve the MVIP (1.1) in finite dimensional spaces in 2009. Later, in 2013, Fang and He [16] introduced an extragradient method for solving the MVIP (1.1) in finite dimensional spaces. In this method, the next iterate is a projection of the current iterate onto the feasible set C. That is, the number of projections onto the feasible set C are doubled per iteration. If the projection onto C does not have a closed form formula, then a minimal distance problem has to be solved twice per iteration during implementation, and this might seriously affect the efficiency of the method. Many efforts have been made to overcome this setback recently.

In 2014, Fang and Chen [15] extended the subgradient extragradient method for solving the MVIP (1.1) in which the second projection onto the feasible set C in the extragradient method is replaced by a projection onto a specific constructible half-space containing C. The subgradient extragradient method of Fang and Chen [15] is given as:

Algorithm 1.1. Choose $\tilde{x}_1 \in \mathbb{R}^m$ and two parameters $\gamma, \delta \in (0,1)$. Set $n = 1$.
Step 1. Apply **Procedure A** (see (2.5)) with $x = \tilde{x}_n$ and set $x_n = R(\tilde{x}_n)$.
Step 2. Choose $u_n \in A(x_n)$ and let k_n be the smallest nonnegative integer satisfying $v_n \in A(P_C(x_n - \gamma^{k_n} u_n))$,

$$\gamma^{k_n} \|u_n - v_n\| \le (1-\delta)\|x_n - P_C(x_n - \gamma^{k_n} u_n)\|. \tag{1.2}$$

Set $\rho_n = \gamma^{k_n}$ and $z_n = P_C(x_n - \rho_n u_n)$. If $x_n = z_n$, then stop.
Step 3. Compute $\tilde{x}_{n+1} = P_{C_n}(x_n - \rho_n v_n)$, where $C_n = \{y \in \mathbb{R}^m : \langle x_n - \rho_n u_n - z_n, y - z_n\rangle \le 0\}$.
Let $n = n+1$ and return to Step 1.
Inspired by Algorithm 1.1, Dong *et al.* [12] proposed the following projection and contraction method for solving the MVIP (1.1).

Algorithm 1.2. Choose $\tilde{x}_1 \in \mathbb{R}^m$ and four parameters $\tau > 0$, $\gamma, \delta \in (0,1)$ and $\alpha \in (0,2)$. Set $n = 1$.
Step 1. Apply **Procedure A** with $x = \tilde{x}_n$ and set $x_n = R(\tilde{x}_n)$.
Step 2. Choose $u_n \in A(x_n)$ and find the smallest nonnegative integer l_k such that $\rho_n = \tau\gamma^{l_k}$ and $v_n \in A(P_C(x_n - \rho_n u_n))$, which satisfies

$$\rho_n \|u_n - v_n\| \le (1-\delta)\|x_n - P_C(x_n - \rho_n u_n)\|. \tag{1.3}$$

Set $y_n = P_C(x_n - \rho_n u_n)$. If $x_n = y_n$, then stop.
Step 3. Compute $\tilde{x}_{n+1} = x_n - \alpha\beta_n d(x_n, y_n)$, where $d(x_n, y_n) - \rho_n(u_n - v_n)$, $\phi(x_n, y_n) := \langle x_n - y_n, d(x_n, y_n)\rangle$ and $\beta_n := \frac{\phi(x_n,y_n)}{\|d(x_n,y_n)\|^2}$.
Let $n = n+1$ and return to Step 1.

By modifying the subgradient extragradient method with the projection and contraction method, He *et al.* [25] proposed the following projection-type method for solving MVIP (1.1):

Algorithm 1.3. Choose $x_1 \in \mathbb{R}^m$ as an initial point and fix four parameters $\gamma, \sigma \in (0,1)$ and $0 < \rho_0 \le \rho_1 < \infty$. Set $C_1 = \mathbb{R}^m, \tilde{x}_1 = x_1$, and $n = 1$.
Step 1: Apply **Procedure A** to obtain $x_n = R(\tilde{x}_n)$.
Step 2: Choose $u_n \in A(x_n)$ and $\rho_n \in [\rho_0, \rho_1]$. Set $y_n = P_C(x_n - \rho_n u_n)$. If $x_n = y_n$, then stop. Otherwise, compute $z_n = \alpha_n y_n + (1-\alpha_n)x_n$ and choose the largest $\alpha \in \{\gamma^0, \gamma, \gamma^2, \gamma^3, \cdots\}$ such that there exists $w_n \in A(z_n)$ satisfying

$$\langle w_n, x_n - y_n\rangle \ge \sigma\langle u_n, x_n - y_n\rangle.$$

Step 3: Take a point $v_n \in A(y_n)$, set $d(x_n, y_n) = (x_n - y_n) - \rho_n(u_n - v_n)$ and compute $\bar{x}_n = x_n - \beta_n d(x_n, y_n)$, where $\beta_n = \frac{\phi(x_n,y_n)}{\|d(x_n,y_n)\|^2}$ and $\phi(x_n, y_n) = \langle x_n - y_n, d(x_n, y_n)\rangle$.
Step 4: Set $C_n = \{y \in \mathbb{R}^m | \langle w_n, y - z_n\rangle \le 0\}$ for $n \ge 2$ and $C_n^* = \cap_{i=1}^n C_i$. Compute $\tilde{x}_{n+1} = P_{C_n^*}(\bar{x}_n)$. If $\tilde{x}_{n+1} = x_n$, then stop. Otherwise, let $n := n+1$ and return Step 1.

As observed in [25, Section 4], Algorithm 1.1, Algorithm 1.2 and Algorithm 1.3 do not work well in some examples because of the presence of **Procedure A** in the iterative steps. Thus, the authors in [25] proposed the following more general projection-type method without **Procedure A** for solving MVIP (1.1), which can be implemented in such examples.

Algorithm 1.4. Choose $x_1 \in \mathbb{R}^m$ as an initial point and fix four parameters $\gamma, \sigma \in (0,1)$ and $0 < \rho_0 \le \rho_1 < \infty$. Set $C_1 = \mathbb{R}^m$ and $n = 1$.
Step 1: Choose $u_n \in A(x_n)$ and $\rho_n \in [\rho_0, \rho_1]$. Set $y_n = P_C(x_n - \rho_n u_n)$. If $x_n = y_n$, then stop. Otherwise, compute $z_n = \alpha_n y_n + (1-\alpha_n)x_n$ and choose the largest $\alpha \in \{\gamma^0, \gamma, \gamma^2, \gamma^3, \cdots\}$ such that there exists $w_n \in A(z_n)$ satisfying

$$\langle w_n, x_n - y_n\rangle \ge \sigma\langle u_n, x_n - y_n\rangle.$$

Step 2: Taking a point $v_n \in A(y_n)$, set $d(x_n, y_n) = (x_n - y_n) - \rho_n(u_n - v_n)$
and compute $\bar{x}_n = x_n - \beta_n d(x_n, y_n)$, where $\beta_n = \frac{\phi(x_n,y_n)}{\|d(x_n,y_n)\|^2}$, $\phi(x_n, y_n) = \langle x_n - y_n, d(x_n, y_n)\rangle$.
Step 3: Set $C_n = \{y \in \mathbb{R}^m | \langle w_n, y - z_n\rangle \le 0\}$ for $n \ge 2$ and $C_n^* = \cap_{i=1}^n C_i$. Compute $x_{n+1} = P_{C \cap C_n^*}(\bar{x}_n)$. If $x_{n+1} = x_n$, then stop. Otherwise, let $n := n+1$ and return Step 1.

A question of interest in the study of variational inequalities and related optimization problems is how to increase the convergence speed of iterative methods. Lots of efforts are being made in constructing iterative methods that improve and speed up the convergence of sequences. Among these, incorporating the inertial extrapolation term in algorithms (resulting into inertial extrapolation methods) is hot. The research on inertial methods is a very large topic in optimization. The method was first considered in [37] for solving the smooth convex minimization problems. It was later made very popular by Nesterov's acceleration gradient method [34], and was further developed by Beck and Teboulle [4] in the case of structured convex minimization [3]. Since then, many researchers have incorporated the inertial extrapolation terms in iterative methods to solve the classical VIPs (see, for example, [10, 13, 26, 41, 44, 47] and the references therein). However, some of the attractive properties these methods with inertial terms for solving the classical VIPs and many other related problems are lost as a result of incorporating the inertial term to these algorithms. For instance, the Fejér monotonicity (with respect to the solution set) of the sequence $\{x_n\}$ generated by these methods is lost and this makes the sequence to move or swing back and forth around the solution set of the problem. This also makes these methods to sometimes not converge faster than the corresponding algorithms without the inertial term.

Our aim in this paper is to design two projection-type methods with inertial extrapolation steps for solving the MVIP (1.1) with the following properties:

- Different from the famous vanilla inertial extrapolation step proposed in [10, 13, 26, 41, 44, 47] for solving variational inequalities in which the Fejér monotonicity property is lost, the proposed methods in this paper involve alternated inertial step and recover the Fejér monotonicity property of the non-inertial projection method to some extent (see [19, 20, 33]).
- The sequence generated by our proposed methods converge to a solution of the MVIP (1.1) under much relaxed assumptions. In particular,
  - the multivalued mapping A is only assumed to be locally bounded and continuous without any monotonicity assumption,
  - the inertial extrapolation factor varies freely in $[0,1]$ without any additional conditions like the need for it to be monotonically decreasing and/or bounded. These features are new to projection-type methods for solving MVIPs. Thus, they bring novelty and state-of-the-art contributions to projection-type methods for solving MVIP (1.1).
- Our proposed inertial methods have fewer evaluations of the multivalued mappings A compared to Algorithms 1.1-1.4.
- Convergence rate analyses of the proposed methods are given even with the presence of inertial term in the algorithms.
- Several computational experiments show that our proposed methods are efficient and outperform many related methods in the literature for solving the MVIP (1.1).

The organization of this paper is as follows. We give some definitions and basic results in Section 2. Some discussions about our proposed methods are given in Section 3. The convergence analysis of our first proposed method is then investigated in Section 4. The convergence of our second proposed method is analyzed in Section 5. In Section 6, we provide some analyses on the convergence rate of our methods. Some numerical experiments are given in Section 7. Lastly, we conclude with some final remarks in Section 8.

2. Preliminaries

The metric projection, denoted by P_C, is a map defined on $\mathbb{R}^m$, which assigns each $x \in \mathbb{R}^m$, to the unique point in C, denoted by $P_C x$,

$$\|x - P_C x\| = \inf\{\|x-y\| : \ y \in C\}.$$

It is well known that P_C is nonexpansive, and characterized by the inequality

$$\langle x - P_C x, y - P_C x \rangle \leq 0 \; \forall y \in C. \tag{2.1}$$

Furthermore, the P_C is known to possess the following property

$$\|P_C x - y\|^2 \leq \|x - y\|^2 - \|P_C x - x\|^2, \; \forall y \in C. \tag{2.2}$$

It is also known that P_C satisfies

$$\langle x - \bar{z}, x - P_C \bar{z} \rangle \geq \|x - P_C \bar{z}\|^2, \; \forall x \in C, \; \bar{z} \in \mathbb{R}^m. \tag{2.3}$$

For more information and properties on P_C, we refer to [21, 23].

Definition 2.1. A multivalued mapping $A : C \rightrightarrows \mathbb{R}^m$ is said to be

- outer-semicontinuous at $x \in C$ if and only if the graph of A is closed;
- inner-semicontinuous at $x \in C$ if, for any sequence $\{x_n\}$ converging to x and $y \in A(x)$, there exists a sequence $\{y_n\}$ in $A(x_n)$ such that $\{y_n\}$ converges to y;
- continuous at $x \in C$ if it is both outer-semicontinuous and inner-semicontinuous at x;
- locally bounded on C if, for every $x \in C$, there exists a neighborhood V of x such that $A(V)$ is bounded, where $A(V) = \cup_{x \in V} A(x)$.

Definition 2.2. A multivalued mapping $A : C \rightrightarrows \mathbb{R}^m$ is said to be

- monotone on C if, for any $x, y \in C$,
$$\langle u - v, x - y \rangle \geq 0, \; \forall u \in A(x), \; v \in A(y);$$
- pseudomonotone on C if, for any $x, y \in C$,
$$\text{there exists } u \in A(x) : \langle u, y - x \rangle \geq 0 \text{ implies } \forall v \in A(y) \; : \; \langle v, y - x \rangle \geq 0;$$
- quasimonotone on C if, for any $x, y \in C$,
$$\text{there exists } u \in A(x) : \langle u, y - x \rangle > 0 \text{ implies } \forall v \in A(y) \; : \; \langle v, y - x \rangle \geq 0.$$

Proposition 2.3. [39] *A multivalued mapping $A : C \rightrightarrows \mathbb{R}^m$ is said to be locally bounded if and only if, for any bounded sequence $\{x_n\}$ with $u_n \in A(x_n)$, the sequence $\{u_n\}$ is bounded.*

Proposition 2.4. [25] *Assume that the solution set of problem (1.1) Γ is nonempty and that $A : C \rightrightarrows \mathbb{R}^m$ is continuous. If either*

(i) A is monotone or pseudomonotone on C;

(ii) A is quasimonotone on C and for any $x^* \in \Gamma$ with $u^* \in A(x^*)$ satisfying (1.1) such that
$$\text{there exists } y^* \in C \; : \; \langle u^*, y^* - x^* \rangle \neq 0;$$

(iii) A is quasimonotone on C with int $C \neq \emptyset$ and $0 \notin A(x^*)$ for all $x^* \in \Gamma$,

then

$$\langle u, y - x^* \rangle \geq 0, \; \forall y \in C, \; u \in A(y), \; x^* \in \Gamma. \tag{2.4}$$

Remark 2.5. We can see from Proposition 2.4 that condition (2.4) is weaker than various monotonicity conditions. Thus, we shall assume, in the rest of this paper, that the solution set of problem (1.1) Γ is nonempty and that (2.4) is satisfied.

Lemma 2.6. [14] *A point $x^* \in \Gamma$ if and only if $r_\rho(x^*, u) := x^* - P_C(x^* - \rho u) = 0$ for some $u \in A(x^*)$ and $\rho > 0$.*

Lemma 2.7. [5] *For any $x \in \mathbb{R}^m$, $u \in A(x)$ and $\rho > 0$,*

$$\min\{1, \rho\}||r_1(x,u)|| \leq ||r_\rho(x,u)|| \leq \max\{1, \rho\}||r_1(x,u)||.$$

Lemma 2.8. *The following is well-known:*

$$2\langle x,y\rangle = \|x\|^2 + \|y\|^2 - \|x-y\|^2 = \|x+y\|^2 - \|x\|^2 - \|y\|^2 \; \forall\, x,y \in \mathbb{R}^m.$$

Definition 2.9. Let C be a nonempty, closed and convex subset of $\mathbb{R}^m$. The sequence $\{x_n\}$ in $\mathbb{R}^m$ is said to be Fejér monotone with respect to C if

$$||x_{n+1} - x|| \leq ||x_n - x|| \; \forall n \in \mathbb{N}, \; x \in C.$$

Lemma 2.10. [23] *Let $C \subseteq \mathbb{R}^m$ be a closed convex set, and let h be a real-valued function on $\mathbb{R}^m$. Define $K := \{x \in C : h(x) \leq 0\}$. If K is nonempty and h is Lipschitz continuous on C with modulus $M > 0$, then*

$$dist(x,K) \geq M^{-1}\max\{h(x),0\}, \forall x \in C,$$

where $dist(x,K)$ denotes the distance function from x to K.

Procedure A [27]
Input: a point $x \in \mathbb{R}^m$.
Output: a point $R(x) \in C$, where $C := \{x \in \mathbb{R}^m \mid g(x) \leq 0\}$, and $g : \mathbb{R}^m \to \mathbb{R}$ is a convex function.
Step 1: set $n = 0$ and $x_n = x$.
Step 2: if $g(x_n) \leq 0$, then stop and set $R(x) = x_n$. Otherwise, go to Step 3.
Step 3: choose a point $w_n \in \partial g(x_n)$, where $\partial g(x)$ denotes the subdifferential of g at x, set

$$x_{n+1} = x_n - 2g(x_n)\frac{w_n}{\|w_n\|^2}, \tag{2.5}$$

and set $n := n+1$ go back to Step 2.

Lemma 2.11. [27] *The number of iterations in* **Procedure A** *is finite and for any given $x \in \mathbb{R}^m$, it holds that*

$$\|R(x) - y\| \leq \|x - y\|, \; \forall y \in C.$$

3. Proposed Methods

In this section, we present our proposed methods and discuss their features. We begin with the following assumptions under which our convergence results are obtained.

Assumption 3.1. *Suppose that the following hold:*

(a) the feasible set C is nonempty, closed and convex subset of $\mathbb{R}^m$;
(b) $A : \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ is locally bounded and continuous;
(c) Γ is nonempty and satisfies condition (2.4).

We now present the first alternating inertial projection-type method of this paper.

Algorithm 3.2.

Step 0: *Choose $x_0, x_1 \in \mathbb{R}^m$ as an initial point and the sequence $\{\theta_n\}$ in $[0,1]$. Fix $\gamma, \sigma \in (0,1)$ and $0 < \rho_0 \leq \rho_1 < \infty$. Set $C_1 = \mathbb{R}^m$ and $n = 1$.*

Step 1. *Set*

$$v_n = \begin{cases} x_n, & \text{if } n \text{ is even}, \\ x_n + \theta_n(x_n - x_{n-1}), & \text{if } n \text{ is odd}; \end{cases}$$

choose $u_n \in A(v_n)$ *and* $\rho_n \in [\rho_0, \rho_1]$. *Then, compute*
$y_n = P_C(v_n - \rho_n u_n)$. *If* $r_{\rho_n}(v_n, u_n) := v_n - P_C(v_n - \rho_n u_n) = 0$: *then STOP. Otherwise, go to Step 2.*

Step 2. *Compute*

$$z_n = v_n - \alpha_n r_{\rho_n}(v_n, u_n),$$

and choose the largest $\alpha \in \{\gamma^0, \gamma, \gamma^2, \gamma^3, \dots\}$ *such that there exists a point* $w_n \in A(z_n)$ *satisfying*

$$\langle w_n, r_{\rho_n}(v_n, u_n)\rangle \geq \sigma \langle u_n, r_{\rho_n}(v_n, u_n)\rangle. \tag{3.1}$$

Step 3. *Set* $C_n = \{y \in \mathbb{R}^m : \langle w_n, y - z_n\rangle \leq 0\}$ *for* $n \geq 2$ *and* $C_n^* = \cap_{i=1}^n C_i$. *Then, compute*

$$x_{n+1} = P_{C_n^*}(v_n).$$

Set $n := n+1$ *and go back to* **Step 1.**

In the following, we present the second alternating inertial projection-type method with a new linesearch, which is different from (3.1).

Algorithm 3.3.

Step 0: *Choose* $x_0, x_1 \in \mathbb{R}^m$ *as an initial point and the sequence* $\{\theta_n\}$ *in* $[0,1]$. *Fix* $\gamma, \sigma \in (0,1)$ *and* $\rho > 0$. *Set* $C_1 = \mathbb{R}^m$ *and* $n = 1$.

Step 1. *Set*

$$v_n = \begin{cases} x_n, & \text{if } n \text{ is even}, \\ x_n + \theta_n(x_n - x_{n-1}), & \text{if } n \text{ is odd}; \end{cases}$$

choose $u_n \in A(v_n)$ *and* $\rho_n \in [\rho, \frac{1}{\sigma}]$. *Then, compute*
$y_n = P_C(v_n - \rho_n u_n)$. *If* $r_{\rho_n}(v_n, u_n) := v_n - P_C(v_n - \rho_n u_n) = 0$, *then STOP. Otherwise, go to Step 2.*

Step 2. *Compute*

$$z_n = v_n - \alpha_n r_{\rho_n}(v_n, u_n),$$

and choose the largest $\alpha \in \{\gamma^0, \gamma, \gamma^2, \gamma^3, \dots\}$ *such that there exists a point* $w_n \in A(z_n)$ *satisfying*

$$\langle w_n, r_{\rho_n}(v_n, u_n)\rangle \geq \frac{\sigma}{2} \|r_{\rho_n}(v_n, u_n)\|^2.$$

Step 3. *Set* $C_n = \{y \in \mathbb{R}^m : \langle w_n, y - z_n\rangle \leq 0\}$ *for* $n \geq 2$ *and* $C_n^* = \cap_{i=1}^n C_i$. *Then, compute*

$$x_{n+1} = P_{C_n^*}(v_n).$$

Set $n := n+1$ *and go back to* **Step 1.**

Remark 3.4. (a) In Algorithms 3.2 and 3.3, the inertial extrapolation factor θ_n varies freely in $[0,1]$ without any extra assumptions like the requirement of the monotonicity of the inertial factor and/or the boundedness of the inertial factor away from 1. Therefore, the condition on the inertial factor in our methods is significantly much more relaxed than many other inertial methods for solving VIPs (see, for example, [10, 13, 26, 41, 47]).

(b) Compared to the Nesterov's method [34] (see also [4, 8, 11, 31, 32]), the method does not allow the case where $\theta_n = 1$, which is achievable in our methods. In fact, in most methods (see, for example, [30, 41, 44]), the inertial factor is restricted in $[0, \frac{1}{3})$. Thus, we have enlarged the inertial factor from $[0, \frac{1}{3})$ to $[0,1]$.

(c) Compared to the relaxed inertial methods recently considered in [1, 2], the inertial factor in these methods is allowed to be in $[0,1]$ but with some additional conditions on both the relaxation and inertial factors (see, for example, condition (K_1) in [1, Page 6]) and [2, Page 554]). In fact, when the inertial factor increases to 1, the relaxation factor must be assumed to decrease rapidly to 0, otherwise the additional conditions will be violated (see, for example, [1, Section 2.5] and [2, Section 3.3]). In other words, in order not to violate the additional conditions in these methods, the inertial factor tends to 1 only if the relaxation factor tends to 0. Note also that, according to condition (K_1) of [1, 2], the relaxation factor cannot be 0. Therefore, the inertial factor cannot be 1. This is in contrast to our methods as we do not require any additional condition on the inertial factor.

(d) As we shall see in Lemma 4.1 (a), the *even* sequence generated by our methods is Fejér monotone with respect to the solution set Γ, which is not obtainable in other inertial methods for solving VIPs, as well as the Nesterov's method and the relaxed inertial methods studied in [1, 2]. Thus, our inertial projection methods partially inherit the Fejér monotonicity property (an attractive property) of the classical projection method for solving VIPs.

(e) Compared to Algorithms 1.1-1.4, our methods have fewer evaluations of the multivalued mapping A.

Lemma 3.5. *The Step 2 of Algorithm 3.2 is well-defined.*

Proof. Let $u \in A(v)$. Then, define $r_\rho(v,u) := v - y$ and $y := P_C(v - \rho u)$ with $\rho > 0$. If $r_\rho(v,u) = 0$, then we obtain by Lemma 2.6 that v is a solution. Otherwise (that is, $r_\rho(v,u) \neq 0$), we obtain by the characterization of P_C that

$$\langle u, r_\rho(v,u)\rangle = \frac{1}{\rho}\langle y - (v - \rho u) + r_\rho(v,u), r_\rho(v,u)\rangle \geq \frac{1}{\rho}\langle r_\rho(v,u), r_\rho(v,u)\rangle > 0. \tag{3.2}$$

Now, let us suppose on the contrary that the Step 2 of Algorithm 3.2 is not well-defined. Then, we will have that, for any $\alpha > 0$ and $w \in A(z)$ with $z = v - \alpha r_\rho(v,u)$,

$$\langle w, r_\rho(v,u)\rangle < \sigma\langle u, r_\rho(v,u)\rangle. \tag{3.3}$$

In particular, for $\alpha_n = \frac{1}{n}$ with $z_n = v - \alpha_n r_\rho(v,u)$, we have $z_n \to v$ as $n \to \infty$. Since A is continuous, it is inner-semicontinuous. Thus, there exists $w_n \in A(z_n)$ such that $w_n \to u$ with $u \in A(v)$. Taking w as w_n in (3.3), and taking limit as $n \to \infty$, we obtain

$$(1-\sigma)\langle u, r_\rho(v,u)\rangle \leq 0,$$

which contradicts (3.2). Hence, the Step 2 of Algorithm 3.2 is well-defined. □

Remark 3.6. The Step 2 of Algorithm 3.3 is also well-defined. Indeed, define $r_{\bar\rho}(x,u) := x - y$. Then, we observe that, if we choose a point $u \in A(x)$ with $y := P_C(x - \bar\rho u)$, $\bar\rho \in [\rho, \frac{1}{\sigma}]$ and set $\bar z = x - \bar\rho u$ in (2.3), then

$$\langle u, r_{\bar\rho}(x,u)\rangle \geq \frac{\sigma}{2}\|r_{\bar\rho}(x,u)\|^2. \tag{3.4}$$

Thus, using (3.4) and the continuity of A, we can see that tge Step 2 of Algorithm 3.3 is well-defined.

Remark 3.7. Observe that Assumption 3.1 (c) ensures that the Step 3 of Algorithms 3.2 and 3.3 is well-defined since $\Gamma \subset C_n^*$. Hence $C_n^* \neq \emptyset$ for all $n \geq 1$. Indeed, for $z \in \Gamma$, we obtain from Assumption 3.1 (c) that $\langle w_n, z - z_n \rangle \leq 0 \; \forall n \geq 1$. Thus, $z \in C_n \; \forall n \geq 1$, which follows that $z \in C_n^* \; \forall n \geq 1$.

4. Convergence Analysis for Algorithm 3.2

Lemma 4.1. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.2. Then, under Assumption 3.1, we have that*

(a) $\|x_{2n+2} - x^*\|^2 \leq \|x_{2n} - x^*\|^2 - \|x_{2n+2} - v_{2n+1}\|^2$ *for all* $x^* \in \Gamma$,

(b) $\lim\limits_{n\to\infty} \|x_{2n} - x^*\|$ *exists for all* $x^* \in \Gamma$,

(c) $\lim\limits_{n\to\infty} \|x_{2n+2} - v_{2n+1}\| = 0$.

Proof. (a) Let $x^* \in \Gamma$. Then, from (2.2), Step 1 and Step 3 of Algorithm 3.2, we obtain, for all $n \geq 1$, that

$$\begin{aligned} \|x_{2n+1} - x^*\|^2 &= \|P_{C_n^*}(v_{2n}) - x^*\|^2 \\ &\leq \|x_{2n} - x^*\|^2 - \|x_{2n+1} - x_{2n}\|^2 \end{aligned} \tag{4.1}$$

and

$$\begin{aligned} \|x_{2n+2} - x^*\|^2 &= \|P_{C_n^*}(v_{2n+1}) - x^*\|^2 \\ &\leq \|x_{2n+1} + \theta_{2n+1}(x_{2n+1} - x_{2n}) - x^*\|^2 \\ &\quad - \|x_{2n+2} - (x_{2n+1} + \theta_{2n+1}(x_{2n+1} - x_{2n}))\|^2. \end{aligned} \tag{4.2}$$

From (4.2) and (2.1), we obtain

$$\begin{aligned} \|x_{2n+2} - x^*\|^2 &\leq \|x_{2n+1} - x^*\|^2 + 2\theta_{2n+1} \langle x_{2n+1} - x^*, x_{2n+1} - x_{2n} \rangle + \theta_{2n+1}^2 \|x_{2n+1} - x_{2n}\|^2 \\ &\quad - \|x_{2n+2} - x_{2n+1}\|^2 + 2\theta_{2n+1} \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n} \rangle - \theta_{2n+1}^2 \|x_{2n+1} - x_{2n}\|^2 \\ &= \|x_{2n+1} - x^*\|^2 + 2\theta_{2n+1} \langle x_{2n+1} - x^*, x_{2n+1} - x_{2n} \rangle - \|x_{2n+2} - x_{2n+1}\|^2 \\ &\quad + 2\theta_{2n+1} \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n} \rangle \\ &\leq \|x_{2n+1} - x^*\|^2 - \|x_{2n+2} - x_{2n+1}\|^2 + 2\theta_{2n+1} \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n} \rangle . \end{aligned} \tag{4.3}$$

Combining (4.1) and (4.3), and using $\theta_{2n+1} \in [0,1]$, we obtain that

$$\begin{aligned} \|x_{2n+2} - x^*\|^2 &\leq \|x_{2n} - x^*\|^2 - \|x_{2n+2} - x_{2n+1}\|^2 + 2\theta_{2n+1} \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n} \rangle \\ &\quad - \|x_{2n+1} - x_{2n}\|^2 \\ &\leq \|x_{2n} - x^*\|^2 - \|x_{2n+2} - x_{2n+1}\|^2 + 2\theta_{2n+1} \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n} \rangle \\ &\quad - \theta_{2n+1}^2 \|x_{2n+1} - x_{2n}\|^2. \end{aligned} \tag{4.4}$$

Since,

$$\begin{aligned}&\|x_{2n+2}-(x_{2n+1}+\theta_{2n+1}(x_{2n+1}-x_{2n}))\|^2\\&=\|x_{2n+2}-x_{2n+1}\|^2-2\theta_{2n+1}\langle x_{2n+2}-x_{2n+1},x_{2n+1}-x_{2n}\rangle\\&+\theta_{2n+1}^2\|x_{2n+1}-x_{2n}\|^2,\end{aligned}$$

we obtain from (4.4) that

$$\begin{aligned}\|x_{2n+2}-x^*\|^2&\leq\|x_{2n}-x^*\|^2-\|x_{2n+2}-x_{2n+1}\|^2+\|x_{2n+2}-x_{2n+1}\|^2+\theta_{2n+1}^2\|x_{2n+1}-x_{2n}\|^2\\&-\|x_{2n+2}-(x_{2n+1}+\theta_{2n+1}(x_{2n+1}-x_{2n}))\|^2-\theta_{2n+1}^2\|x_{2n+1}-x_{2n}\|^2\\&=\|x_{2n}-x^*\|^2-\|x_{2n+2}-v_{2n+1}\|^2,\end{aligned}\tag{4.5}$$

which gives part (a).

Next, we prove part (b). From (4.5), we obtain that

$$\|x_{2n+2}-x^*\|\leq\|x_{2n}-x^*\|.$$

Therefore, the *even* sequence $\{x_{2n}\}$ is Fejér monotone with respect to Γ. Hence, $\lim_{n\to\infty}\|x_{2n}-x^*\|$ exists.

Finally, we prove part (c). Letting $j\geq 1$, we obtain from (4.5) that

$$\sum_{n=1}^{j}\|x_{2n+2}-v_{2n+1}\|^2\leq\|x_2-x^*\|^2.\tag{4.6}$$

As $j\to\infty$ in (4.6), we get that

$$\sum_{n=1}^{\infty}\|x_{2n+2}-v_{2n+1}\|^2\leq\|x_2-x^*\|^2<\infty,$$

which implies part (c). □

Lemma 4.2. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.2 under Assumption 3.1. Then, $\lim_{n\to\infty}\alpha_{2n+1}\|y_{2n+1}-v_{2n+1}\|^2=0$. Moreover, if there exists a subsequence $\{x_{2n_k}\}$ of $\{x_{2n}\}$ such that $\{x_{2n_k}\}$ converges to $\bar{x}$ and $\bar{x}\notin\Gamma$, then*

(a) $\liminf_{k\to\infty}\alpha_{2n_k+1}>0$*;*

(b) $\lim_{k\to\infty}\|v_{2n_k+1}-y_{2n_k+1}\|=0.$

Proof. From Step 1, Step 2 and the characterization of P_C, we obtain that

$$\begin{aligned}&\alpha_{2n+1}\|v_{2n+1}-y_{2n+1}\|^2\\\leq\;&\alpha_{2n+1}\langle r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1}),\ r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1})\rangle\\&+\alpha_{2n+1}\langle y_{2n+1}-(v_{2n+1}-\rho_{2n+1}u_{2n+1}),v_{2n+1}-y_{2n+1}\rangle\\\leq\;&\alpha_{2n+1}\rho_{2n+1}\langle u_{2n+1},r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1})\rangle\qquad(4.7)\\\leq\;&\frac{\alpha_{2n+1}\rho_{2n+1}}{\sigma}\langle w_{2n+1},r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1})\rangle\\=\;&\frac{\rho_{2n+1}}{\sigma}\langle w_{2n+1},v_{2n+1}-z_{2n+1}\rangle\\\leq\;&\frac{\rho_{2n+1}}{\sigma}\left(\langle w_{2n+1},v_{2n+1}-x_{2n+2}\rangle+\langle w_{2n+1},x_{2n+2}-z_{2n+1}\rangle\right).\qquad(4.8)\end{aligned}$$

Since $x_{2n+2} = P_{C^*_{2n+1}}(v_{2n+1})$, it follows that $\langle w_{2n+1}, x_{2n+2} - z_{2n+1}\rangle \leq 0$. Hence, (4.8) becomes

$$\begin{aligned} \alpha_{2n+1}\|v_{2n+1} - y_{2n+1}\|^2 &\leq \frac{\rho_{2n+1}}{\sigma}\langle w_{2n+1}, v_{2n+1} - x_{2n+2}\rangle \\ &\leq \frac{\rho_1}{\sigma}||w_{2n+1}||||v_{2n+1} - x_{2n+2}||. \end{aligned} \tag{4.9}$$

By Lemma 4.1, the *even* sequence $\{x_{2n}\}$ is bounded, so is $\{x_{2n+2}\}$. Hence $\{v_{2n+1}\}$ is also bounded. Thus, $\{z_{2n+1}\}$ is bounded as well. Since A is locally bounded, we obtain from Proposition 2.3 that $\{w_{2n+1}\}$ is also bounded. Hence, we obtain from (4.9) and Lemma 4.1 (c) that

$$\lim_{n\to\infty} \alpha_{2n+1}\|y_{2n+1} - v_{2n+1}\|^2 = 0. \tag{4.10}$$

We now prove part (a). We know from Step 2 that $\{\alpha_{2n+1}\} \subset [0,1]$ is bounded. Thus, there exists a subsequence $\{\alpha_{2n_k+1}\}$ of $\{\alpha_{2n+1}\}$ such that $\liminf\limits_{k\to\infty} \alpha_{2n_k+1} \geq 0$. In fact, we claim that $\liminf\limits_{k\to\infty} \alpha_{2n_k+1} > 0$. Suppose otherwise (that is, $\liminf\limits_{k\to\infty} \alpha_{2n_k+1} = 0$). Then, without loss of generality, we can choose a subsequence of $\{\alpha_{2n_k+1}\}$ still denoted by $\{\alpha_{2n_k+1}\}$ such that $\lim\limits_{k\to\infty} \alpha_{2n_k+1} = 0$. Now, we define $\bar{\alpha}_{2n_k+1} := \frac{\alpha_{2n_k+1}}{\gamma}$, $\bar{z}_{2n_k+1} := v_{2n_k+1} - \bar{\alpha}_{2n_k+1} r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})$. Then, by the boundedness of $\{r(v_{2n_k+1})\} = \{y_{2n_k+1} - v_{2n_k+1}\}$ and $\alpha_{2n_k+1} \to 0$ as $k \to \infty$, we obtain that

$$\lim_{k\to\infty} \|\bar{z}_{2n_k+1} - v_{2n_k+1}\| = 0. \tag{4.11}$$

Now, let $\{x_{2n_k}\}$ be a subsequence of $\{x_{2n}\}$ such that $x_{2n_k} \to \bar{x}$. Using Lemma 4.1 (c), we obtain that $v_{2n_k-1} \to \bar{x}$. Using Assumption 3.1 (b), the boundedness of $\{v_{2n_k-1}\}$ and Proposition 2.3, we obtain that $\{u_{2n_k-1}\}$ is also bounded. Thus, we can choose a subsequence of $\{u_{2n_k-1}\}$ still denoted by $\{u_{2n_k-1}\}$ such that $u_{2n_k-1} \to \bar{u}$. Since A is continuous, it is outer-semicontinuous. Hence, $\bar{u} \in A(\bar{x})$. Furthermore, we can also assume without loss of generality that $\rho_{2n_k-1} \to \rho \in [\rho_0, \rho_1]$. Therefore, we obtain from the continuity of P_C that $y_{2n_k-1} \to \bar{y}$ as $k \to \infty$, where $\bar{y} = P_C(\bar{x} - \rho\bar{u})$. Again, from (4.11), we obtain that $\bar{z}_{2n_k-1} \to \bar{x}$. Since A is inner-semicontinuous and $\bar{u} \in A(\bar{x})$, we can choose a subsequence $w_{2n_k-1} \in A(\bar{z}_{2n_k-1})$ such that $\bar{w}_{2n_k-1} \to \bar{u}$. From the definition of $\bar{z}_{2n_k+1}$ and Step 2, we obtain that

$$\langle \bar{w}_{2n_k-1},\ r_{\rho_{2n_k-1}}(v_{2n_k-1}, u_{2n_k-1})\rangle < \sigma\langle u_{2n_k-1},\ r_{\rho_{2n_k-1}}(v_{2n_k-1}, u_{2n_k-1})\rangle. \tag{4.12}$$

Taking limit as $k \to \infty$ in (4.12), we obtain that

$$\langle \bar{u}, \bar{x} - \bar{y}\rangle \leq 0. \tag{4.13}$$

On the other hand, since $\bar{x} \notin \Gamma$, we have from Lemma 2.6 that $\bar{x} \neq \bar{y}$. Hence,

$$\langle \bar{u}, \bar{x} - \bar{y}\rangle = \frac{1}{\rho}\langle \bar{y} - (\bar{x} - \rho\bar{u}) + (\bar{x} - \bar{y}), \bar{x} - \bar{y}\rangle > \frac{1}{\rho}\langle \bar{x} - \bar{y}, \bar{x} - \bar{y}\rangle > 0, \tag{4.14}$$

which is a contradiction to (4.13). Therefore, our claim holds. That is, $\liminf\limits_{k\to\infty} \alpha_{2n_k+1} > 0$.

Next, we prove part (b). From part (a), we have that $\liminf_{k\to\infty}\alpha_{2n_k+1} > 0$. Therefore, we obtain from (4.10) that

$$\begin{aligned}
0 &\leq \limsup_{k\to\infty}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\|^2\\
&\leq \left(\limsup_{k\to\infty}\alpha_{2n_k+1}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\|^2\right)\left(\limsup_{k\to\infty}\frac{1}{\alpha_{2n_k+1}}\right)\\
&= \left(\limsup_{k\to\infty}\alpha_{2n_k+1}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\|^2\right)\left(\frac{1}{\liminf_{k\to\infty}\alpha_{2n_k+1}}\right)\\
&= 0.
\end{aligned}$$

Therefore, we obtain that

$$\lim_{k\to\infty}\|v_{2n_k+1}-y_{2n_k+1}\| = \lim_{k\to\infty}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\| = 0.$$

□

We are now in position to give the first main theorem of this section.

Theorem 4.3. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.2. Then, under Assumption 3.1, we have that $\{x_n\}$ converges to an element of Γ.*

Proof. From Lemma 4.1, we find that $\{x_{2n}\}$ is bounded. Thus, there exists a subsequence $\{x_{2n_k}\}$ of $\{x_{2n}\}$ such that $\{x_{2n_k}\}$ converges to some point $\bar{x}$. Hence, it follows from Lemma 4.1 (c) that $\{v_{2n_k-1}\}$ also converges to $\bar{x}$.

Claim: $\bar{x}\in\Gamma$.

Suppose on the contrary that $\bar{x}\notin\Gamma$. Then, it follows from Lemma 4.2 (b) that $\{y_{2n_k-1}\}$ also converges to $\bar{x}$. Now, without loss of generality, we may assume that $\rho_{2n_k-1}\to\bar{\rho}$ and $u_{n_k}\to\bar{u}$. Since A is continuous, it is outer-semicontinuous. Thus, $\bar{u}\in A(\bar{x})$ and

$$P_C(\bar{x}-\bar{\rho}\bar{u}) = \lim_{k\to\infty}P_C(v_{2n_k-1}-\rho_{2n_k-1}u_{2n_k-1}) = \lim_{k\to\infty}y_{2n_k-1} = \bar{x},$$

which implies by Lemma 2.6 that $\bar{x}\in\Gamma$. This leads to a contraction. Hence, our claim holds.

We now show that $\{x_n\}$ converges to $\bar{x}$. Since x^* is arbitrarily chosen in Γ, we can replace x^* by $\bar{x}$ in Lemma 4.1 (b) to get that $\lim_{n\to\infty}\|x_{2n}-\bar{x}\|^2$ exists. Since $\bar{x}$ is an accumulation point of $\{x_{2n}\}$, we obtain that $\{x_{2n}\}$ converges to $\bar{x}$. Furthermore, we have from (4.1) that

$$\|x_{2n+1}-\bar{x}\| \leq ||x_{2n}-\bar{x}||.$$

Thus, we obtain that $\{x_{2n+1}\}$ converges to $\bar{x}$. Therefore, the whole sequence $\{x_n\}$ converges to $\bar{x}\in\Gamma$. □

Remark 4.4. In the case that $\theta_n = 0$ for all $n\geq 1$ in Algorithm 3.2, we have $v_n = x_n$ for all $n\geq 1$. In this case, we can employ **Procedure A** (see (2.5)) to obtain the following algorithm.

Algorithm 4.5.

Step 0: *Let $x_0,x_1\in\mathbb{R}^m$ be given arbitrary and fix $\gamma,\sigma\in(0,1), 0<\rho_0\leq\rho_1<\infty$. Set $C_1=\mathbb{R}^m$, $\tilde{x}_1 = x_1$ and $n=1$.*

Step 1. *Apply* **Procedure A** *to obtain $x_n = R(\tilde{x}_n)$.*

Step 2. *Choose $u_n\in A(x_n)$ and $\rho_n\in[\rho_0,\rho_1]$. Then, compute*

$y_n = P_C(x_n - \rho_n u_n)$. If $r_{\rho_n}(x_n,u_n) := x_n - P_C(x_n - \rho_n u_n) = 0$, then STOP. Otherwise, go to Step 2.

***Step 3.** Compute*

$$z_n = x_n - \alpha_n r_{\rho_n}(x_n,u_n)$$

and choose the largest $\alpha \in \{\gamma^0, \gamma, \gamma^2, \gamma^3, \dots\}$ such that there exists a point $w_n \in A(z_n)$ satisfying

$$\langle w_n, r_{\rho_n}(x_n,u_n)\rangle \geq \sigma\langle u_n, r_{\rho_n}(x_n,u_n)\rangle. \tag{4.15}$$

***Step 4.** Set $C_n = \{y \in \mathbb{R}^m : \langle w_n, y - z_n\rangle \leq 0\}$ for $n \geq 2$ and $C_n^* = \cap_{i=1}^n C_i$. Then, compute*

$$\tilde{x}_{n+1} = P_{C_n^*}(x_n).$$

If $\tilde{x}_{n+1} = x_n$, then stop. Otherwise, let $n = n+1$ and return to Step 1.

Corollary 4.6. *Let $\{x_n\}$ be a sequence generated by Algorithm 4.5 such that the following assumptions hold:*

(a) the set C is described as in **procedure A***,*
(b) $A : C \rightrightarrows \mathbb{R}^m$ is locally bounded and continuous,
(c) Γ is nonempty and satisfies condition (2.4).

Then, $\{x_n\}$ converges to an element of Γ.

Proof. It directly follows from Lemma 2.11 and Theorem 4.3. □

Remark 4.7. Under the settings of Remark 4.4, we can obtain, in general, the following algorithm without **Procedure A**.

Algorithm 4.8.

Step 0: *Let $x_0, x_1 \in C$ be given arbitrary and fix $\gamma, \sigma \in (0,1), 0 < \rho_0 \leq \rho_1 < \infty$. Set $C_1 = \mathbb{R}^m$ and $n = 1$.*
Step 1. *Choose $u_n \in A(x_n)$ and $\rho_n \in [\rho_0, \rho_1]$. Then, compute*
$y_n = P_C(x_n - \rho_n u_n)$. If $r_{\rho_n}(x_n,u_n) := x_n - P_C(x_n - \rho_n u_n) = 0$: STOP. Otherwise, go to Step 2.

***Step 2.** Compute*

$$z_n = x_n - \alpha_n r_{\rho_n}(x_n,u_n)$$

and choose the largest $\alpha \in \{\gamma^0, \gamma, \gamma^2, \gamma^3, \dots\}$ such that there exists a point $w_n \in A(z_n)$ satisfying

$$\langle w_n, r_{\rho_n}(x_n,u_n)\rangle \geq \sigma\langle u_n, r_{\rho_n}(x_n,u_n)\rangle. \tag{4.16}$$

***Step 3.** Set $C_n = \{y \in \mathbb{R}^m : \langle w_n, y - z_n\rangle \leq 0\}$ for $n \geq 2$ and $C_n^* = \cap_{i=1}^n C_i$. Then, compute*

$$x_{n+1} = P_{C \cap C_n^*}(x_n).$$

If $x_{n+1} = x_n$, then stop. Otherwise, let $n = n+1$ and return to Step 1.

Corollary 4.9. *Let $\{x_n\}$ be a sequence generated by Algorithm 4.8 such that the following assumptions hold:*

(a) The feasible set C is a nonempty closed and convex subset of $\mathbb{R}^m$,
(b) $A : C \rightrightarrows \mathbb{R}^m$ is locally bounded and continuous,
(c) Γ is nonempty and satisfies condition (2.4).

Then, $\{x_n\}$ converges to an element of Γ.

Proof. It directly follows from Corollary 4.6. □

Remark 4.10. Note that the linesearch procedure in Algorithm 4.5 requires to calculate the projection onto C only one time in each search which is in contrast to Algorithm 1.1 and Algorithm 1.2 of Fang and Chen [15] and Dong *et al.* [12] respectively, where the Armijo-type linesearch procedures in these algorithms require to compute the projection onto C multiple times in each search. Also, compared to Algorithm 1.3 and Algorithm 1.4 of He *et al.* [25], Algorithm 4.5 and Algorithm 4.8 have fewer evaluations of the multivalued mapping A than these algorithms since the Step 3 of Algorithm 1.3 and the Step 2 of Algorithm 1.4 are dispensed in Algorithm 4.5 and Algorithm 4.8, respectively. Thus, Corollary 4.6 improves the main theorems in Fang and Chen [15] and Dong *et al.* [12], while Corollary 4.6 and Corollary 4.9 improve the main theorems in He *et al.* [25].

5. Convergence Analysis for Algorithm 3.3

Note that the Step 2 (the linesearch procedure) of Algorithm 3.2 was not utilized in the proof of Lemma 4.1. Thus, Lemma 4.1 also holds if $\{x_n\}$ is generated by Algorithm 3.3. Therefore, we only need to present and prove the version of Lemma 4.2 and Theorem 4.3 corresponding to Algorithm 3.3 in this section.

Lemma 5.1. *Let the sequence $\{x_n\}$ be generated by Algorithm 3.3 such that Assumption 3.1 is satisfied. Then,*

(a) $\lim_{n\to\infty} \alpha_{2n+1}\|y_{2n+1}-v_{2n+1}\|^2 = 0$;

(b) if there exists a subsequence $\{x_{2n_k}\}$ of $\{x_{2n}\}$ such that $\{x_{2n_k}\}$ converges to $\bar{x}$, then $\lim_{k\to\infty}\|v_{2n_k+1}-y_{2n_k+1}\| = 0$.

Proof. (a) From (2.1), the Step 2 of Algorithm 3.3 and the fact that $x_{2n+2}\in C^*_{2n+1}$, we obtain that

$$\begin{aligned}\alpha_{2n+1}\|r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1})\|^2 &\le \frac{2\alpha_{2n+1}}{\sigma}\langle w_{2n+1}, r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1})\rangle\\ &\le \frac{2}{\sigma}\langle w_{2n+1}, v_{2n+1}-z_{2n+1}\rangle\\ &\le \frac{2}{\sigma}\left(\langle w_{2n+1}, v_{2n+1}-x_{2n+2}\rangle+\langle w_{2n+1}, x_{2n+2}-z_{2n+1}\rangle\right)\\ &\le \frac{2}{\sigma}\|w_{2n+1}\|\|v_{2n+1}-x_{2n+2}\|. \qquad (5.1)\end{aligned}$$

From Lemma 4.1, the *even* sequence $\{x_{2n}\}$ is bounded, so is $\{v_{2n+1}\}$. Thus, $\{z_{2n+1}\}$ is also bounded. Since A is locally bounded, we obtain from Proposition 2.3 that $\{w_{2n+1}\}$ is also bounded. Hence, we obtain from (5.1) and Lemma 4.1 (c) that

$$\lim_{n\to\infty}\alpha_{2n+1}\|v_{2n+1}-y_{2n+1}\|^2 = \lim_{n\to\infty}\alpha_{2n+1}\|r_{\rho_{2n+1}}(v_{2n+1},u_{2n+1})\|^2 = 0. \qquad (5.2)$$

(b) Since $\{\alpha_{2n+1}\}\subset[0,1]$ is bounded, we have that $\liminf_{n\to\infty}\alpha_{2n+1}\ge 0$.

We now consider two possible cases:

Case 1. Suppose that $\liminf_{n\to\infty}\alpha_{2n+1} = 0$. Then, we can choose a subsequence of $\{\alpha_{2n+1}\}$, denoted by $\{\alpha_{2n_k+1}\}$, such that $\lim\limits_{k\to\infty} \alpha_{2n_k+1} = 0$ and

$$\lim_{k\to\infty} \|v_{2n_k+1} - y_{2n_k+1}\| = t \geq 0. \tag{5.3}$$

Now, define $\bar{\alpha}_{2n_k+1} := \frac{\alpha_{2n_k+1}}{\gamma}$. Then, $\bar{z}_{2n_k+1} := v_{2n_k+1} - \bar{\alpha}_{2n_k+1} r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})$. Since $\alpha_{2n_k+1} \to 0$ as $k \to \infty$, we obtain that $\bar{\alpha}_{2n_k+1} \to 0$ as $k \to \infty$. Hence,

$$\lim_{k\to\infty} \|\bar{z}_{2n_k+1} - v_{2n_k+1}\| = 0. \tag{5.4}$$

Now, from the definition of $\bar{z}_{2n_k+1}$ and the Step 2, we obtain that

$$\langle \bar{w}_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle < \frac{\sigma}{2} \|r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\|^2,$$

which implies that

$$\begin{aligned} &2\langle \bar{w}_{2n_k+1} - u_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle + 2\langle u_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle \\ &< \sigma \|r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\|^2. \end{aligned} \tag{5.5}$$

Set $s_{2n_k+1} := v_{2n_k+1} - \rho_{2n_k+1} u_{2n_k+1}$. Then, (5.5) becomes

$$\begin{aligned} &2\langle \bar{w}_{2n_k+1} - u_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle + \frac{2}{\rho_{2n_k+1}} \langle v_{2n_k+1} - s_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle \\ &< \sigma \|r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\|^2, \end{aligned}$$

which implies that

$$\begin{aligned} &2\langle \bar{w}_{2n_k+1} - u_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle \\ &+ \frac{1}{\rho_{2n_k+1}} \left(\|r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\|^2 + \|s_{2n_k+1} - v_{2n_k+1}\|^2 - \|s_{2n_k+1} - y_{2n_k+1}\|^2 \right) \\ &< \sigma \|r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\|^2. \end{aligned}$$

That is,

$$\begin{aligned} &\frac{1}{\rho_{2n_k+1}} \left(\|s_{2n_k+1} - v_{2n_k+1}\|^2 - \|s_{2n_k+1} - y_{2n_k+1}\|^2 \right) \\ &< (\sigma - \frac{1}{\rho_{2n_k+1}}) \|r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\|^2 \\ &-2\langle \bar{w}_{2n_k+1} - u_{2n_k+1}, r_{\rho_{2n_k+1}}(v_{2n_k+1}, u_{2n_k+1})\rangle. \end{aligned} \tag{5.6}$$

Let $\{x_{2n_k}\}$ be a subsequence of $\{x_{2n}\}$ such that $x_{2n_k} \to \bar{x}$. Using Lemma 4.1 (c), we obtain that $v_{2n_k-1} \to \bar{x}$. Using Assumption 3.1 (b), the boundedness of $\{v_{2n_k-1}\}$ and Proposition 2.3, we obtain that $\{u_{2n_k-1}\}$ is also bounded. Thus, we can choose a subsequence of $\{u_{2n_k-1}\}$, still denoted by $\{u_{2n_k-1}\}$, such that $u_{2n_k-1} \to \bar{u}$. Since A is continuous, it is outer-semicontinuous. Hence, $\bar{u} \in A(\bar{x})$. We also assume without loss of generality that $\rho_{2n_k-1} \to \bar{\rho} \in [\rho, \frac{1}{\sigma})$. Again, from (5.4), we obtain that $\bar{z}_{2n_k-1} \to \bar{x}$. Since A is inner-semicontinuous and $\bar{u} \in A(\bar{x})$, we can choose a subsequence $w_{2n_k-1} \in A(\bar{z}_{2n_k-1})$ such that $\bar{w}_{2n_k-1} \to \bar{u}$. Replacing "$2n_k + 1$" by "$2n_k -$

1” in (5.6) and noting that $\{v_{2n_k-1}\}$, $\{u_{2n_k-1}\}$, $\{y_{2n_k-1}\}$ and $\{\bar{w}_{2n_k-1}\}$ are bounded, we can choose a subsequence $\{k_j\}$ of $\{k\}$ such that

$$\begin{aligned}
&\frac{1}{\bar{\rho}}\Big[\limsup_{k\to\infty}\big(\|s_{2n_k-1}-v_{2n_k-1}\|^2-\|s_{2n_k-1}-y_{2n_k-1}\|^2\big)\Big]\\
&\le \limsup_{k\to\infty}\Big[\big(\sigma-\frac{1}{\rho_{2n_k-1}}\big)\|r_{\rho_{2n_k-1}}(v_{2n_k-1},u_{2n_k-1})\|^2\\
&\quad -2\langle \bar{w}_{2n_k-1}-u_{2n_k-1}, r_{\rho_{2n_k-1}}(v_{2n_k-1},u_{2n_k-1})\rangle\Big]\\
&= \lim_{j\to\infty}\Big[\big(\sigma-\frac{1}{\rho_{2n_{k_j}-1}}\big)\|r_{\rho_{2n_{k_j}-1}}(v_{2n_{k_j}-1},u_{2n_{k_j}-1})\|^2\\
&\quad -2\langle \bar{w}_{2n_{k_j}-1}-u_{2n_{k_j}-1}, r_{\rho_{2n_{k_j}-1}}(v_{2n_{k_j}-1},u_{2n_{k_j}-1})\rangle\Big].
\end{aligned}$$

Thus, we obtain from (5.3) that

$$\limsup_{k\to\infty}\big(\|s_{2n_k-1}-v_{2n_k-1}\|^2-\|s_{2n_k-1}-y_{2n_k-1}\|^2\big) \quad\le\quad \bar{\rho}(\sigma-\frac{1}{\bar{\rho}})t. \tag{5.7}$$

At this point, we claim that $t=0$. Otherwise, (5.7) will become

$$\limsup_{k\to\infty}\big(\|s_{2n_k-1}-v_{2n_k-1}\|^2-\|s_{2n_k-1}-y_{2n_k-1}\|^2\big)\le \bar{\rho}(\sigma-\frac{1}{\bar{\rho}})t<0.$$

For $\varepsilon=\frac{-\bar{\rho}\left(\sigma-\frac{1}{\bar{\rho}}\right)}{2}t>0$, there exists $N\in\mathbb{N}$ such that

$$\|s_{2n_k-1}-v_{2n_k-1}\|^2-\|s_{2n_k-1}-y_{2n_k-1}\|^2\le\bar{\rho}\left(\sigma-\frac{1}{\bar{\rho}}\right)+\varepsilon=\frac{\bar{\rho}\left(\sigma-\frac{1}{\bar{\rho}}\right)}{2}<0$$

$\forall k\in\mathbb{N}, k\ge N$. Thus, we obtain that

$$\|v_{2n_k-1}-s_{2n_k-1}\|<\|y_{2n_k-1}-s_{2n_k-1}\|,\ \forall k\in\mathbb{N},$$

which is a contradiction to the definition of $y_{2n_k-1}=P_C(v_{2n_k-1}-\rho_{2n_k-1}u_{2n_k-1})=P_C(s_{2n_k-1})$. Therefore, $t=0$. Hence, (5.3) becomes

$$\lim_{k\to\infty}\|v_{2n_k+1}-y_{2n_k+1}\|=0.$$

Case 2. Suppose that $\liminf_{n\to\infty}\alpha_{2n+1}>0$. Then, we obtain from (5.2) that

$$\begin{aligned}
0&\le\limsup_{k\to\infty}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\|^2\\
&\le\limsup_{k\to\infty}\left(\alpha_{2n_k+1}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\|^2\right)\left(\limsup_{k\to\infty}\frac{1}{\alpha_{2n_k+1}}\right)\\
&=\left(\limsup_{k\to\infty}\alpha_{2n_k+1}\|r_{\rho_{2n_k+1}}(v_{2n_k+1},u_{2n_k+1})\|^2\right)\left(\frac{1}{\liminf_{k\to\infty}\alpha_{2n_k+1}}\right)\\
&=0.
\end{aligned}$$

Therefore, we obtain that

$$\lim_{k\to\infty}\|v_{2n_k+1}-y_{2n_k+1}\|=0.$$

□

Theorem 5.2. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.3. Then, under Assumption 3.1, we have that $\{x_n\}$ converges to an element of* Γ.

Proof. From Lemma 4.1, we have that $\{x_{2n}\}$ is bounded. Thus, there exists a subsequence $\{x_{2n_k}\}$ of $\{x_{2n}\}$ such that $\{x_{2n_k}\}$ converges to some point $\bar{x}$. Hence, we obtain from Lemma 4.1 (c) that $\{v_{2n_k-1}\}$ also converges to $\bar{x}$. It then follows from Lemma 5.1 (b) that $\{y_{2n_k-1}\}$ converges to $\bar{x}$. Also, without loss of generality, we may assume that $\rho_{2n_k-1} \to \bar{\rho}$ and $u_{2n_k-1} \to \bar{u}$. Since A is continuous, we obtain that $\bar{u} \in A(\bar{x})$. Therefore, we have

$$P_C(\bar{x}-\bar{\rho}\bar{u}) = \lim_{k\to\infty} P_C(v_{2n_k-1}-\rho_{2n_k-1}u_{2n_k-1}) = \lim_{k\to\infty} y_{2n_k-1} = \bar{x},$$

which follows from Lemma 2.6 that $\bar{x} \in \Gamma$.

We now show that $\{x_n\}$ converges to $\bar{x}$. Since x^* is arbitrarily chosen in Γ, we can replace x^* by $\bar{x}$ in Lemma 4.1 (b) to get $\lim\limits_{n\to\infty} ||x_{2n}-\bar{x}||^2$ exists. Since $\bar{x}$ is an accumulation point of $\{x_{2n}\}$, we obtain that $\{x_{2n}\}$ converges to $\bar{x}$. Furthermore, we have from (4.1) that

$$||x_{2n+1}-\bar{x}|| \leq ||x_{2n}-\bar{x}||.$$

Thus, $\{x_{2n+1}\}$ converges to $\bar{x}$. Therefore, the whole sequence $\{x_n\}$ converges to $\bar{x} \in \Gamma$. □

Remark 5.3. Following Remark 4.4 and Remark 4.10, we can obtain some corollaries of Theorem 5.2 corresponding to Corollaries 4.6 and 4.9, which also improve the main theorems in Dong *et al.* [12], Fang and Chen [15], and He *et al.* [25].

6. Convergence rate

In this section, we provide some results on the convergence rate of the subsequences generated by Algorithm 3.2 and Algorithm 3.3. To establish these results, we need the following crucial lemmas.

Lemma 6.1. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.2 under Assumption 3.1. Then, there exist $t \geq 0$ and $M > 0$ such that*

$$||x_{2n+2}-x^*||^2 \leq ||x_{2n}-x^*||^2 - (1-\theta_{2n+1}^2)\left(\frac{\sigma t}{M\rho_1}||r_{\rho_{2n}}(x_{2n},u_{2n})||^2\right)^2.$$

Proof. Let $F_n(y) := \langle w_n, y-z_n\rangle$, $\forall y \in \mathbb{R}^m$, where $w_n \in A(z_n)$. Then, in Step 3, $C_n = \{y \in \mathbb{R}^m : F_n(y) \leq 0\}$. Recall that $\{w_n\}$ is bounded. Thus, there exists $M > 0$ such that $||w_n|| \leq M, \forall n \geq 1$. Hence, for all $x, y \in \mathbb{R}^m$, we obtain

$$\begin{aligned}|F_n(x)-F_n(y)| &= |\langle w_n, x-y\rangle| \\ &\leq ||w_n||||x-y|| \leq M||x-y||.\end{aligned}$$

Therefore, F_n is Lipschitz continuous on $\mathbb{R}^m$ with modulus $M > 0$. Thus, by Lemma 2.10 and Step 3, we obtain

$$\begin{aligned}M^{-1}\max\{F_n(v_n),0\} &\leq \operatorname{dist}(v_n,C_n) \\ &= ||x_{n+1}-v_n|| \;\forall n \geq 1. \end{aligned} \tag{6.1}$$

From Step 2 and (4.7), we obtain that

$$\begin{aligned} F_n(v_n) &= \langle w_n, v_n - z_n\rangle \\ &= \alpha_n \langle w_n, r_{\rho_n}(v_n,u_n)\rangle \\ &\geq \alpha_n \sigma \langle u_n, r_{\rho_n}(v_n,u_n)\rangle \\ &\geq \frac{\alpha_n \sigma}{\rho_n}\|r_{\rho_n}(v_n,u_n)\|^2 \geq 0. \end{aligned} \tag{6.2}$$

Hence, we obtain from (6.1) and (6.2) that

$$\begin{aligned} \|x_{n+1} - v_n\| &\geq M^{-1}\max\ \{F_n(v_n), 0\} \\ &\geq \frac{M^{-1}\alpha_n\sigma}{\rho_n}\|r_{\rho_n}(v_n,u_n)\|^2. \end{aligned} \tag{6.3}$$

Since

$$\begin{aligned} \|x_{2n+2} - v_{2n+1}\|^2 = \|x_{2n+2} - x_{2n+1}\|^2 - 2\theta_{2n+1}\ \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n}\rangle \\ + \theta_{2n+1}^2\|x_{2n+1} - x_{2n}\|^2, \end{aligned}$$

we obtain from (4.3) and (4.1) that

$$\begin{aligned} \|x_{2n+2} - x^*\|^2 &\leq \|x_{2n+1} - x^*\|^2 - \|x_{2n+2} - x_{2n+1}\|^2 + 2\theta_{2n+1}\ \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n}\rangle \\ &\leq ||x_{2n} - x^*||^2 - \|x_{2n+1} - x_{2n}\|^2 - \|x_{2n+2} - x_{2n+1}\|^2 \\ &\quad + 2\theta_{2n+1}\ \langle x_{2n+2} - x_{2n+1}, x_{2n+1} - x_{2n}\rangle \\ &= ||x_{2n} - x^*||^2 - (1-\theta_{2n+1}^2)\|x_{2n+1} - x_{2n}\|^2 - \|x_{2n+2} - v_{2n+1}\|^2. \end{aligned} \tag{6.4}$$

Since $\{\alpha_n\} \subset [0,1]$ is bounded, then its lower limit must exists. Let $\alpha := \liminf\limits_{n\to\infty} \alpha_n$. Clearly, $\alpha \geq 0$, and there exists a positive integer N such that $\alpha_n \geq \frac{1}{2}\alpha$, $\forall n \geq N$. Now, Lemma 3.5 implies that the Step 2 of Algorithm 3.2 has finite termination. This implies that $\alpha_n \geq 0$, $\forall n < N$. Let $t^0 := \min\limits_{n<N}\{\alpha_n\}$, and define $t := \min\{\frac{1}{2}\alpha,\ t^0\}$. Clearly $t^0 \geq 0$ and so, $t \geq 0$. Also, for any $n > 0$, we see that $\alpha_n \geq t$. Therefore, putting $n = 2n$ in (6.3) and combining with (6.4) (noting that $x_{2n} = v_{2n}$), we get that

$$\begin{aligned} &\|x_{2n+2} - x^*\|^2 \\ &\leq ||x_{2n} - x^*||^2 - (1-\theta_{2n+1}^2)\left(\frac{\sigma t}{M\rho_1}\|r_{\rho_{2n}}(v_{2n},u_{2n})\|^2\right)^2 - \|x_{2n+2} - v_{2n+1}\|^2 \\ &\leq ||x_{2n} - x^*||^2 - (1-\theta_{2n+1}^2)\left(\frac{\sigma t}{M\rho_1}\|r_{\rho_{2n}}(x_{2n},u_{2n})\|^2\right)^2. \end{aligned}$$

□

We now present the corresponding result for Algorithm 3.3.

Lemma 6.2. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.3 under Assumption 3.1. Then, there exist $t \geq 0$ and $M > 0$ such that*

$$\|x_{2n+2} - x^*\|^2 \leq ||x_{2n} - x^*||^2 - (1-\theta_{2n+1}^2)\left(\frac{\sigma t}{2M}\|r_{\rho_{2n}}(x_{2n},u_{2n})\|^2\right)^2.$$

Proof. By the similar argument as in (6.2), we obtain in the case of Algorithm 3.3 that

$$F_n(v_n) \geq \frac{\alpha_n \sigma}{2} \|r_{\rho_n}(v_n, u_n)\|^2 \geq 0 \; \forall n \geq 1.$$

Hence, the rest of the proof follows from the proof of Lemma 6.1. □

We now state and prove the theorems concerning the convergence rate of the sequence generated by Algorithm 3.2 and Algorithm 3.3, respectively.

Theorem 6.3. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.2 under Assumption 3.1. If $0 \leq \theta_n \leq \tilde{\theta} < 1$, $\forall n < 1$, and there exist positive constants c and θ such that*

$$dist\ (x, \Gamma) \leq c \|r_1(x, w)\|^{\frac{1}{2}} \; \forall (x, w) \in P(\theta), \tag{6.5}$$

where $P(\theta) := \{(x, w) \in \mathbb{R}^m \times \mathbb{R}^m : w \in A(x),\ \|r_1(x, w)\| \leq \theta\}$, then there is a constant $q \geq 0$ such that for sufficiently large n,

$$dist(x_n, \Gamma) \leq \frac{dist(x_2, \Gamma)}{\sqrt{q^2 n\ dist^2(x_2, \Gamma) + 1}} = \frac{1}{\sqrt{q^2 n + dist^{-2}(x_2, \Gamma)}}.$$

Proof. Let $x^* \in P_\Gamma(x_n)$. Then, by (6.5), Lemma 6.1 and Lemma 2.7, we obtain

$$\begin{aligned}
\mathrm{dist}^2(x_{2n+2}, \Gamma) &\leq \|x_{2n+2} - x^*\|^2 \\
&\leq \|x_{2n} - x^*\|^2 - (1 - \theta_{2n+1}^2) \left(\frac{\sigma t}{M\rho_1} \|r_{\rho_{2n}}(x_{2n}, u_{2n})\|^2 \right)^2 \\
&\leq \|x_{2n} - x^*\|^2 - (1 - \theta_{2n+1}^2) \left(\frac{\sigma t}{M\rho_1} \min\{1,\ \rho_{2n}^2\} \|r_1(x_{2n}, u_{2n})\|^2 \right)^2 \\
&\leq \|x_{2n} - x^*\|^2 - (1 - \theta_{2n+1}^2) \left(\frac{\sigma t}{M\rho_1} \min\{1,\ \rho_0^2\} \|r_1(x_{2n}, u_{2n})\|^2 \right)^2 \\
&\leq \mathrm{dist}^2(x_{2n}, \Gamma) - \left((1 - \tilde{\theta}^2)^{\frac{1}{2}} \frac{\sigma t}{M\rho_1} \min\{1,\ \rho_0^2\} c^{-4} \mathrm{dist}^4(x_{2n}, \Gamma) \right)^2 \\
&= \mathrm{dist}^2(x_{2n}, \Gamma) - \left(q\ \mathrm{dist}^4(x_{2n}, \Gamma) \right)^2,
\end{aligned}$$

where $q = (1 - \tilde{\theta}^2)^{\frac{1}{2}} \frac{\sigma t}{M\rho_1} \min\{1,\ \rho_0^2\} c^{-4}$. Thus, applying Lemma 2.2 of [36, Chapter 2], we obtain that

$$\mathrm{dist}(x_{2n}, \Gamma) \leq \frac{\mathrm{dist}(x_2, \Gamma)}{\sqrt{q^2 n\ \mathrm{dist}^2(x_2, \Gamma) + 1}} = \frac{1}{\sqrt{q^2 n + \mathrm{dist}^{-2}(x_2, \Gamma)}}.$$

Thus, it follows from (4.1) that

$$\mathrm{dist}(x_{2n+1}, \Gamma) \leq \mathrm{dist}(x_{2n}, \Gamma) \leq \frac{\mathrm{dist}(x_2, \Gamma)}{\sqrt{q^2 n\ \mathrm{dist}^2(x_2, \Gamma) + 1}} = \frac{1}{\sqrt{q^2 n + \mathrm{dist}^{-2}(x_2, \Gamma)}}.$$

Hence, the proof is complete. □

Theorem 6.4. *Let $\{x_n\}$ be a sequence generated by Algorithm 3.3 under Assumption 3.1. If $0 \leq \theta_n \leq \tilde{\theta} < 1$, $\forall n \geq 1$, and there exist positive constants c and θ such that condition (6.5) holds, then there is a constant $\tilde{q} \geq 0$ such that for sufficiently large n,*

$$dist(x_n,\Gamma) \leq \frac{dist(x_2,\Gamma)}{\sqrt{\tilde{q}^2 n\, dist^2(x_2,\Gamma)+1}} = \frac{1}{\sqrt{\tilde{q}^2 n + dist^{-2}(x_2,\Gamma)}}.$$

Proof. Similar to the proof of Theorem 6.3, by employing Lemma 6.2 in place of Lemma 6.1, we obtain the desired conclusion. □

Remark 6.5. The error bound (6.5) playes a crucial role in our convergence rate analyses. Similar conditions to (6.5) have also been used in [15, 42] for convergence rate analyses. In [15], it was shown that (6.5) possibly holds when A is pseudomonotone. In fact, the research on error bound is a very wide and interesting topic in mathematics programming. See, for example, [14, Chapter 6] and [35], for the role played by error bounds in the convergence analyses of iterative schemes.

7. Numerical Experiments

In this section, we discuss the numerical behavior of our proposed methods considered in Section 3 using many test examples taken from the literature. In all the examples considered in this section, we give numerical comparison of our methods with the methods of Dong *et al.* [12, Algorithm 3.1] (see Algorithm 1.2 of this paper), Fang and Chen [15, Algorithm 2.1] (see Algorithm 1.1 of this paper), He *et al.* [25, Algorithm 1 and Algorithm 2] (see Algorithms 1.3 and 1.4, respectively of this paper) and Ye and He [48, Algorithm 2.1].

All codes are written in Matlab 2016 (b) and performed on a personal computer with an Intel(R) Core(TM) i5-2600 CPU at 2.30GHz and 8.00 Gb-RAM. In Tables 1-4, "Iter." means the number of iterations while "CPU" means the CPU time in seconds.

Example 7.1. Let $\tilde{t} = (x_1 + \sqrt{x_1^2 + 4x_2})/2$. We define

$$A(x_1,x_2) = \begin{cases} \left(\frac{-\tilde{t}}{(1+\tilde{t})}, \frac{-1}{(1+\tilde{t})}\right), & \text{if } (x_1,x_2) \neq (0,0), \\ \\ (0, \ -1), & \text{if } (x_1,x_2) = (0,0). \end{cases}$$

This example has also been considered in [22, 48], where A is quasimonotone and $C := [0,1] \times [0,1]$.

For the parameters, we choose $\sigma = 0.99$, $\rho_n \in (0, \ 1]$ and $\gamma = 0.4$ for Algorithm 3.2, Algorithm 3.3 and He *et al.* [25, Algorithm 1 and Algorithm 2]; $\delta = 0.8$, $\gamma = 0.4, \tau = 2.0$ and $\alpha = 1.1$ for Dong *et al.* [12, Algorithm 3.1]; $\delta = 0.8$ and $\gamma = 0.4$ for Fang and Chen [15, Algorithm 2.1]; and $\sigma = 0.99$, $\gamma = 0.4$ for Ye and He [48, Algorithm 2.1]. Also, in this example, we use the stopping criterion $\|v_n - y_n\| \leq 10^{-4}$ and obtain the numerical results listed in Table 1 and Figure 1. We stress that this stopping criterion and the choices for σ and γ are the same as in Ye and He [48, Section 4].

We consider the following cases for the numerical experiments of this example.

Case 1: $x_1 = (0.5, 0.25)$, $x_0 = (0.1, 0.1)$ and $\theta_n = \frac{n}{3n+1}$.

Case 2: $x_1 = (0.5, 1)$, $x_0 = (0.3, 0.3)$ and $\theta_n = \frac{1}{10n+3}$.

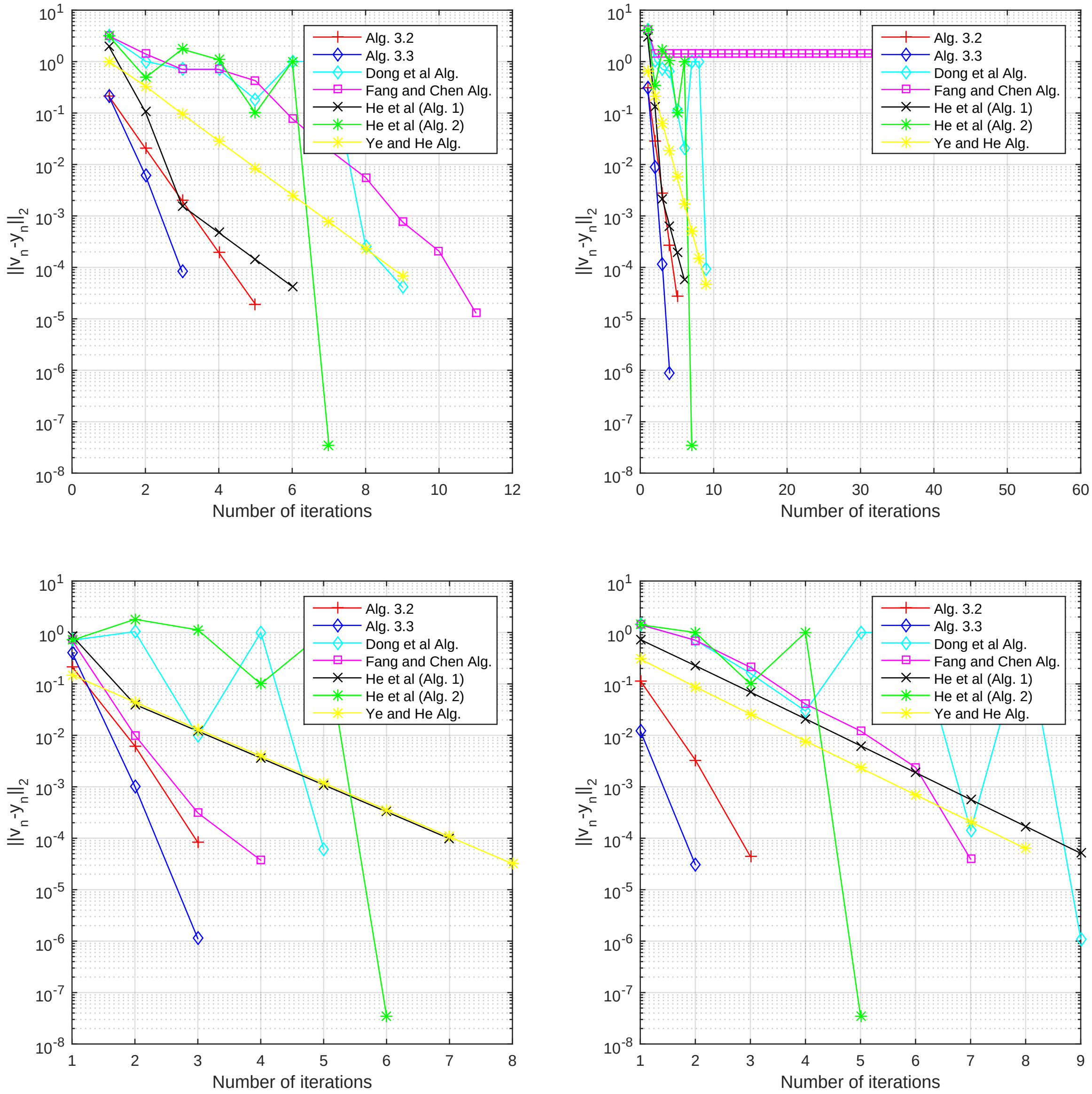

FIGURE 1. Example 7.1 with $\|v_n - y_n\| \leq 10^{-4}$: Top Left: **Case 1**; Top Right: **Case 2**; Bottom Left: **Case 3**; Bottom Right: **Case 4**.

Case 3: $x_1 = (1, 0.8)$, $x_0 = (0.1, 0.1)$ and $\theta_n = \frac{n}{2n+5}$.
Case 4: $x_1 = (0, 0.9)$, $x_0 = (0.3, 0.3)$ and $\theta_n = \frac{n+1}{n+4}$.

Table 1. Numerical results for Example 7.1 with $\|v_n - y_n\| \leq 10^{-4}$.

Cases		Alg. 3.2	Alg. 3.3	Dong et al.	Fang & Chen	He et al. (1)	He et al. (2)	Ye & He
1:	CPU	0.0116	0.0098	0.0193	0.0262	0.0173	0.0213	0.0124
	Iter.	5	3	9	11	6	7	9
2:	CPU	0.0116	0.0080	0.0161	0.1129	0.0157	0.0200	0.0127
	Iter.	5	4	8	53	6	7	9
3:	CPU	0.0118	0.0076	0.0155	0.0142	0.0216	0.0240	0.0128
	Iter.	3	3	5	4	7	6	8
4:	CPU	0.0124	0.0068	0.0177	0.0153	0.0167	0.0197	0.0128
	Iter.	3	2	9	7	9	5	8

Example 7.2. Define

$$A(x_1,x_2,x_3) := \{(t,\ t-x_1,\ t-x_2) : t \in [0,1]\}$$

and $C = \{x \in \mathbb{R}^3_+ : \sum_{i=1}^{3} x_i = 1\}$. Then A satisfies Assumption 3.1, with $x^* := (0,0,1)$ as a solution of the MVIP (1.1) (see, for example, [15, Example 4.1]). This example has also been considered by many other authors (see, for example [17]).

For the parameters, we choose $\sigma = 0.6$, $\rho_n \in (0,\ 1.6]$ and $\gamma = 0.8$ for Algorithm 3.2, Algorithm 3.3 and He *et al.* [25, Algorithm 1 and Algorithm 2]; $\delta = 0.6$, $\gamma = 0.8, \tau = 1.0$ and $\alpha = 1.5$ for Dong *et al.* [12, Algorithm 3.1]; $\delta = 0.6$ and $\gamma = 0.8$ for Fang and Chen [15, Algorithm 2.1]; and $\sigma = 0.6$, $\gamma = 0.8$ for Ye and He [48, Algorithm 2.1]. Also, in this example, we use the stopping criterion $\|x_n - x^*\| \leq 10^{-7}$ and obtain the numerical results listed in Table 2 and Figure 2. We also stress that this stopping criterion and the choices for δ and γ are the same as in Fang and Chen [15, Example 4.1].

Furthermore, we consider the following cases for the numerical experiments of this example.
Case 1: $x_1 = (0.5,0.25,0.25)$, $x_0 = (0.1,0.1,0.8)$ and $\theta_n = \frac{n}{3n+1}$.
Case 2: $x_1 = (0.1,0.7,0.2)$, $x_0 = (0.1,0.4,0.5)$ and $\theta_n = \frac{1}{10n+3}$.
Case 3: $x_1 = (0.3,0.2,0.5)$, $x_0 = (0.5,0.4,0.1)$ and $\theta_n = \frac{n}{2n+5}$.
Case 4: $x_1 = (0.1,0.4,0.5)$, $x_0 = (0.5,0.1,0.4)$ and $\theta_n = \frac{n+1}{n+4}$.

Table 2. Numerical results for Example 7.2 with $\|x_n - x^*\| \leq 10^{-7}$.

Cases		Alg. 3.2	Alg. 3.3	Dong et al.	Fang & Chen	He et al. (1)	He et al. (2)
1:	CPU	0.0121	0.0077	0.0153	0.0149	0.0219	0.0248
	Iter.	5	3	21	34	40	45
2:	CPU	0.0113	0.0084	0.0150	0.0158	0.0358	0.0259
	Iter.	5	3	17	34	43	48
3:	CPU	0.0229	0.0206	0.0247	0.0280	0.0345	0.0245
	Iter.	5	4	13	34	44	49
4:	CPU	0.0123	0.0085	0.0143	0.0161	0.0227	0.0281
	Iter.	5	4	15	34	45	50

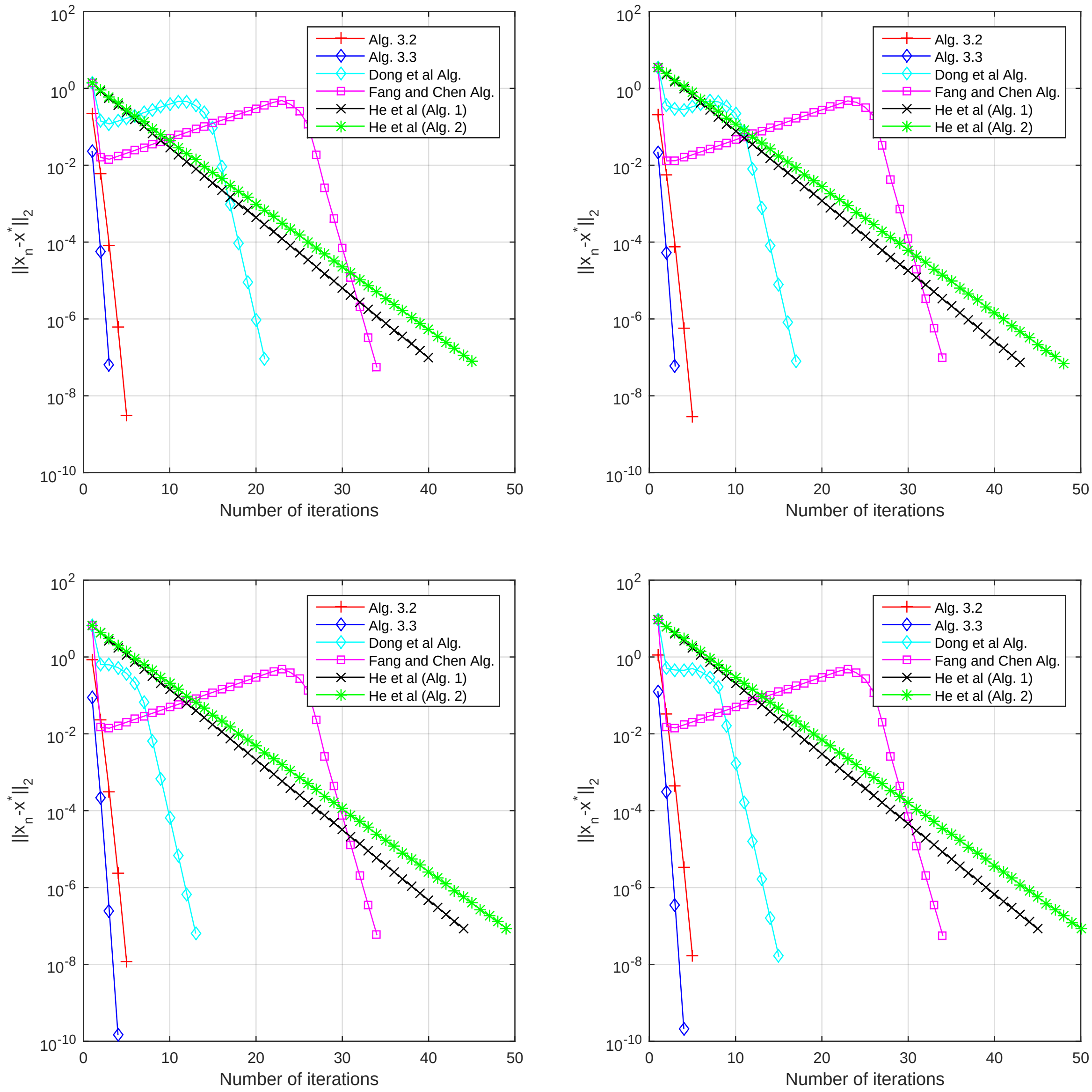

FIGURE 2. Example 7.2 with $\|x_n - x^*\| \leq 10^{-7}$: Top Left: **Case 1**; Top Right: **Case 2**; Bottom Left: **Case 3**; Bottom Right: **Case 4**.

Example 7.3. Define

$$A(x_1, x_2, x_3) := \{(t,\ t - x_1,\ t - x_2) : t \in [0,1]\}$$

and $C = \{x \in \mathbb{R}^3 : g(x) \leq 0\}$, where $g(x) = \max_{1 \leq i \leq 4} \{g_i(x)\}$, $g_i(x) = -x_i (i = 1,2,3)$ and $g_4(x) = x_1 + x_2 + x_3 - 1$. Then A satisfies Assumption 3.1, with $x^* := (0,0,0)$ as a solution of the MVIP (1.1) (see, for example [15, Example 4.1]).

We choose $\sigma = 0.6$, $\rho_n \in (0,\ 1.6]$ and $\gamma = 0.8$ for Algorithm 3.2, Algorithm 3.3 and He, Huang and Li [25, Algorithm 1 and Algorithm 2]; $\delta = 0.6$, $\gamma = 0.8, \tau = 1.0$ and $\alpha = 1.5$ for Dong *et al.* [12, Algorithm 3.1]; $\delta = 0.6$ and $\gamma = 0.8$ for Fang and Chen [15, Algorithm 2.1];

and $\sigma = 0.6,\ \gamma = 0.8$ for Ye and He [48, Algorithm 2.1]. Also, we use the stopping criterion $\|x_n - x^*\| \leq 10^{-7}$ and obtain the numerical results listed in Table 2 and Figure 2. This stopping criterion is the same as in Fang and Chen [15, Example 4.2].

Furthermore, we consider the following cases for the numerical experiments of this example.
Case 1: $x_1 = (-0.5, -0.25, 0),\ x_0 = (-0.5, 0.5, -0.5)$ and $\theta_n = \frac{n}{3n+1}$.
Case 2: $x_1 = (0.7, 0.3, 1),\ x_0 = (-1, 2, 0.8)$ and $\theta_n = \frac{1}{10n+3}$.
Case 3: $x_1 = (0.1, 0.4, -0.5),\ x_0 = (1, -1, 0.8)$ and $\theta_n = \frac{n}{2n+5}$.
Case 4: $x_1 = (-0.1, 0.4, -0.5),\ x_0 = (0.1, 0.4, -0.5)$ and $\theta_n = \frac{n+1}{n+4}$.

Table 3. Numerical results for Example 7.3 with $\|x_n\| \leq 10^{-7}$.

Cases		Alg. 3.2	Alg. 3.3	Dong et al.	Fang & Chen	He et al. (1)	He et al. (2)
1:	CPU	0.0205	0.0098	0.0155	0.0214	0.0292	0.0483
	Iter.	5	4	9	34	46	51
2:	CPU	0.0238	0.0141	0.0307	0.0333	0.0359	0.0601
	Iter.	5	4	9	55	46	52
3:	CPU	0.0122	0.0076	0.0142	0.0160	0.0212	0.0261
	Iter.	5	4	27	34	47	52
4:	CPU	0.0119	0.0082	0.0140	0.0168	0.0209	0.0281
	Iter.	5	4	21	34	52	57

Finally, we consider the following optimization problem which has also been considered in many papers (see, for example, [25, 48]).

Example 7.4.

$$\min_{x \in C} \varphi(x),$$

where $C = \left\{x \in \mathbb{R}^5 : x_i \geq 0,\ i = 1, 2, \cdots, 5,\ \sum_{i=1}^5 x_i = a,\ a > 0\right\}$ and $\varphi(x) = \frac{0.5\langle Hx, x\rangle + \langle q, x\rangle + 1}{\sum\limits_{i=1}^{5} x_i}$.
Here, H denotes a positive diagonal matrix with elements h uniformly taken from the interval $(0.1,\ 1.6)$, and $q = (-1, -1, -1, -1, -1)$. Clearly, this problem is equivalent to MVIP (1.1) with solution set $\Gamma = \{\frac{1}{5}(a, \cdots, a)\}$, where $A(x) = (A_1(x), \cdots, A_5(x))$ and

$$A_i(x) = \frac{\partial \varphi(x)}{\partial x_i} = \frac{hx_i \sum\limits_{i=1}^{5} x_i - 0.5h \sum_{i=1}^5 x_i^2 - 1}{\left(\sum\limits_{i=1}^{5} x_i\right)^2}.$$

We choose $\sigma = 0.99,\ \rho_n \in (0,\ 1]$ and $\gamma = 0.4$ for Algorithm 3.2, Algorithm 3.3 and He *et al.* [25, Algorithm 2]; $\sigma = 0.99,\ \gamma = 0.4$ for Ye and He [48, Algorithm 2.1], and randomly choose values of a (as in the cases below). Furthermore, we use the stopping criterion $\|x_n - x^*\| \leq 10^{-4}$ and obtain the numerical results reported in Table 4 and Figure 4. We emphasis that this stopping criterion and the choices of the parameters are the same as in Ye and He [48, Section 4] and He *et al.* [25, Example 4.2].

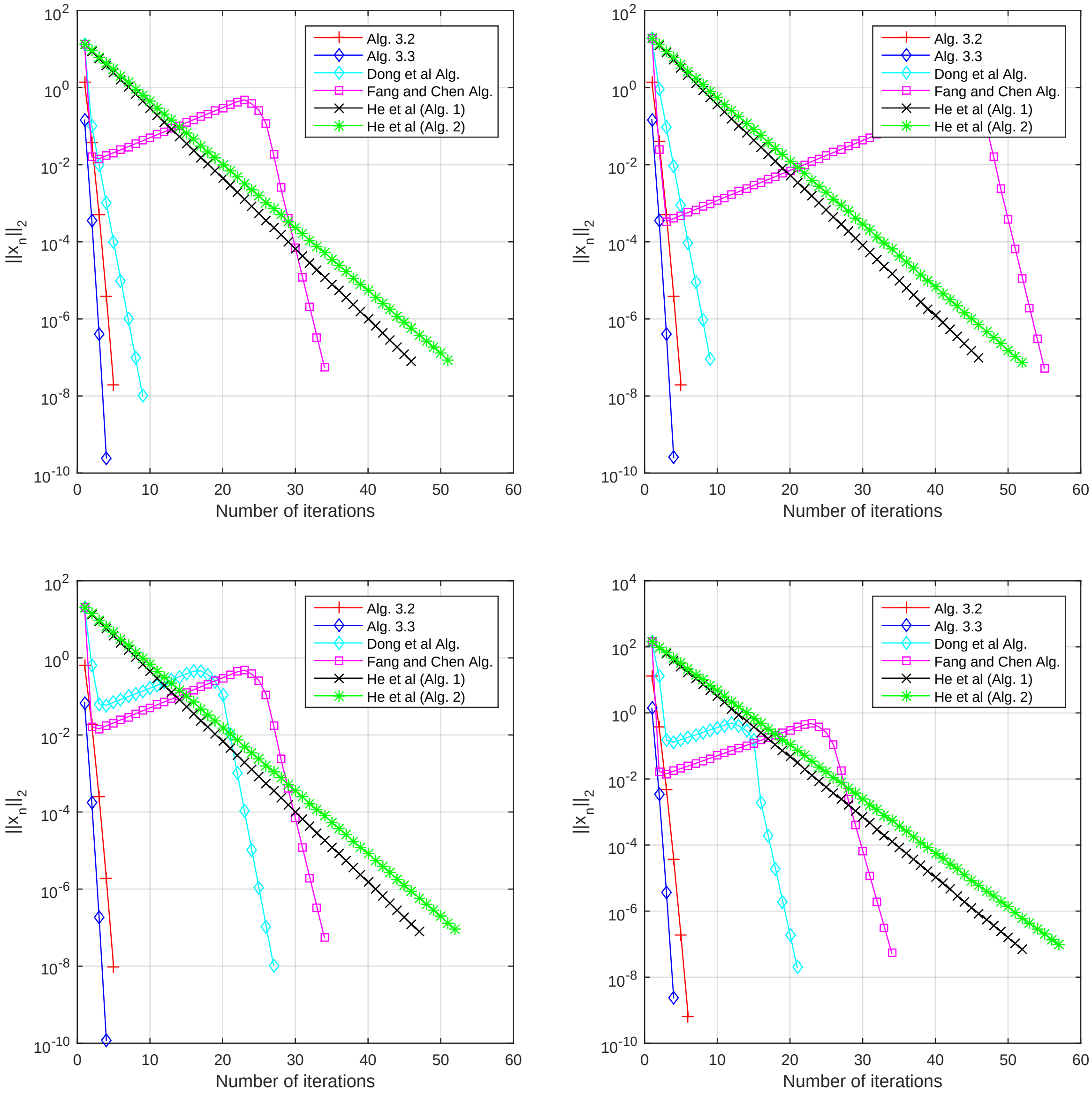

FIGURE 3. Example 7.3 with $\|x_n\| \le 10^{-7}$: Top Left: **Case 1**; Top Right: **Case 2**; Bottom Left: **Case 3**; Bottom Right: **Case 4**.

We consider the following cases for the numerical experiments.

Case 1: $x_1 = (0.5, 1.5, 0.5, 1.5, 1)$, $x_0 = (1, 0.5, 1, 1.5, 1)$, $a = 5$ and $\theta_n = \frac{n}{3n+1}$.
Case 2: $x_1 = (4.3, 2.5, 2.2, 0.3, 0.7)$, $x_0 = (4, 3, 2, 0.3, 0.7)$, $a = 10$ and $\theta_n = \frac{1}{10n+3}$.
Case 3: $x_1 = (0.3, 0.5, 1.2, 2.5, 0.5)$, $x_0 = (0.3, 0.5, 1.2, 2.5, 0.5)$, $a = 5$ and $\theta_n = \frac{n}{2n+5}$.
Case 4: $x_1 = (1.3, 1.5, 2.2, 3.5, 1.5)$, $x_0 = (1.3, 1.5, 2.2, 3.5, 1.5)$, $a = 10$ and $\theta_n = \frac{n+1}{n+4}$.

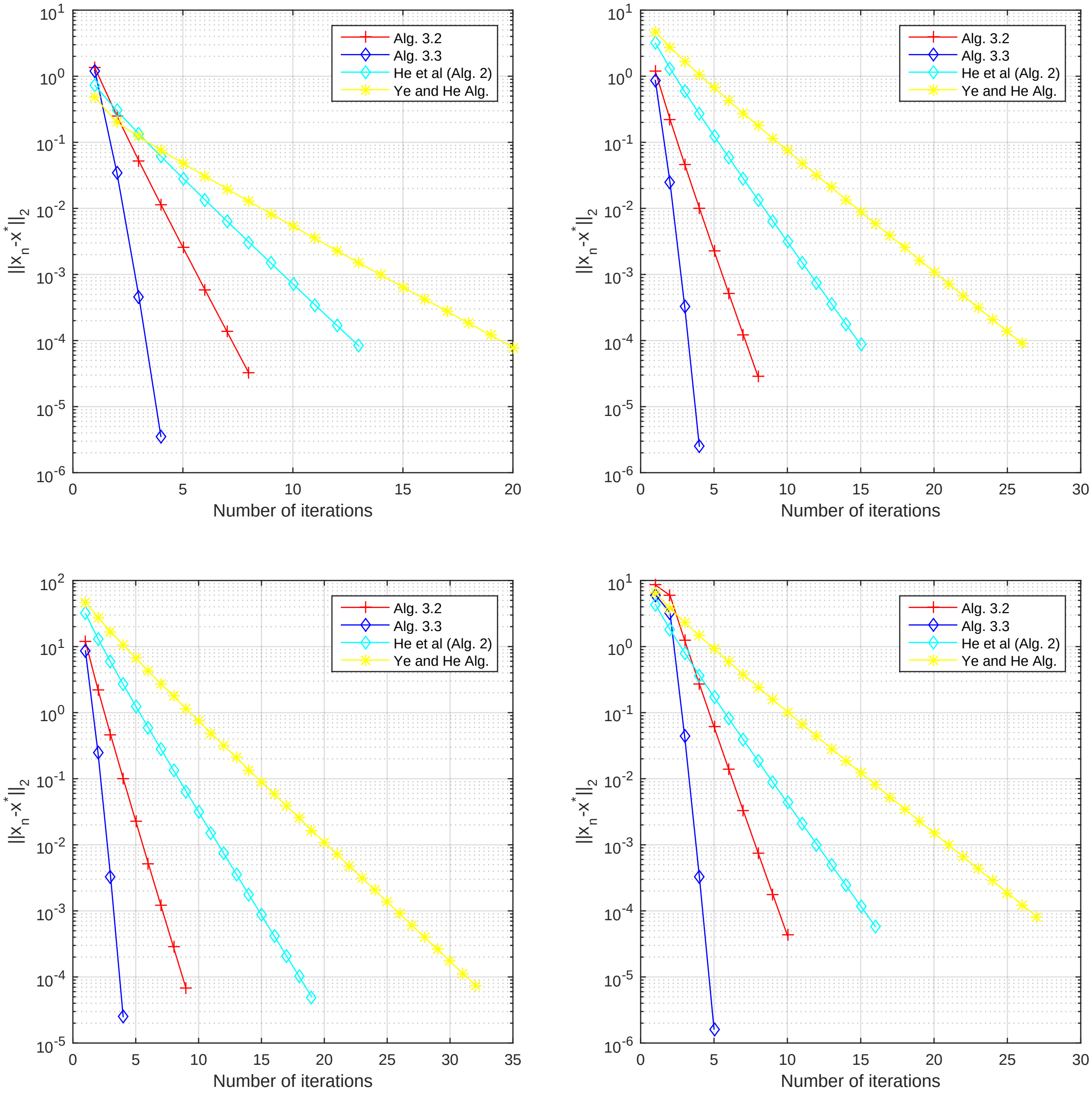

FIGURE 4. Example 7.4 with $\|x_n - x^*\| \le 10^{-4}$: Top Left: **Case 1**; Top Right: **Case 2**; Bottom Left: **Case 3**; Bottom Right: **Case 4**.

Table 4. Numerical results for Example 7.4 with $\|x_n - x^*\| \le 10^{-4}$.

Cases		Alg. 3.2	Alg. 3.3	He et al. (2)	Ye & He
1:	CPU Iter.	0.0237 8	0.0132 4	0.0255 13	0.0355 20
2:	CPU Iter.	0.0249 8	0.0163 4	0.0264 15	0.0384 26
3:	CPU Iter.	0.0132 9	0.0097 4	0.0203 19	0.0305 32
4:	CPU Iter.	0.0174 10	0.0171 5	0.0261 16	0.0372 27

Remark 7.5. By using different starting points and varying the inertial extrapolation factor θ_n in $[0,1]$ (see **Case 1-Case 4**) in each example, we obtain the numerical results displayed in Tables 1-4 and Figures 1-4. We compared our methods with the methods of Dong *et al.* [12, Algorithm 3.1], Fang and Chen [15, Algorithm 2.1], He *et al.* [25, Algorithm 1 and Algorithm 2] and Ye and He [48, Algorithm 2.1]. The following are observed from our numerical experiments:

- In Example 7.4, the methods of Dong *et al.* [12, Algorithm 3.1], Fang and Chen [15, Algorithm 2.1] and He *et al.* [25, Algorithm 1] do not work well. We suspect that this may be as a result of the presence of **Procedure A** in these algorithms. Thus, we only compared our methods with the methods of He *et al.* [25, Algorithm 2] and Ye and He [48, Algorithm 2.1] for this example.

- We see from Table 2 that the number of iterations for our proposed Algorithm 3.2 remains consistent for all starting points while every other methods are dependent on the different starting points. Also, from Table 3, we see that the number of iterations for our proposed Algorithm 3.2 and Algorithm 3.3 are consistent for all starting points while every other methods depend on the different starting points. Furthermore, the method of Fang and Chen [15, Algorithm 2.1] and our proposed Algorithm 3.3 are almost consistent (in terms of number of iterations) in Example 7.3 and Example 7.4 (see Table 3 and Table 4), respectively. Finally, none of the methods are consistent in Example 7.1 (see Table 1).

- It can easily be inferred from the Tables and Figures that in terms of CPU time and number of iterations, our proposed Algorithm 3.3 outperforms the proposed Algorithm 3.2 while Algorithm 3.2 outperforms the methods of Dong *et al.* [12, Algorithm 3.1], Fang and Chen [15, Algorithm 2.1], He *et al.* [25, Algorithm 1 and Algorithm 2] and Ye and He [48, Algorithm 2.1].

Therefore, in all the test examples, our proposed Algorithm 3.2 is the most consistent (well-behaved) method (in terms of number of iterations) while our proposed Algorithm 3.3 is the fastest convergent method (in terms of CPU time). Thus, our methods are more efficient than these other methods.

8. Conclusion

Two projection-type methods with alternated inertial extrapolation steps were introduced and studied for solving multivalued variational inequality problems. Our analyses shown that these proposed methods partially inherit the Fejér monotonicity property, which is lost in many projection methods with inertial extrapolation steps for solving variational inequalities and many related problems. Moreover, we proved the convergence of the sequence generated by these methods under much relaxed assumptions on the inertial extrapolation factor θ_n and the mutivalued mapping A. In particular, the inertial extrapolation factor varies freely in $[0,1]$ without any additional condition, the multivalued mapping A is only required to be locally bounded and continuous without any monotonicity assumption, and the methods have fewer evaluations of A compared to many existing methods for solving the multivalued variational inequality problems. Finally, we established the convergence rate of our methods and also considered some numerical implementations of our proposed methods in comparison with other state-of-the-art methods for solving multivalued variational inequality problems. In all our comparisons, the numerical results demonstrate that our methods perform better than these other methods.

Acknowledgments

The research of the first author was supported by the National Research Foundation (NRF) South Africa (S& F-DSI/NRF Free Standing Postdoctoral Fellowship; Grant Number: 120784). The first author acknowledges the financial support from DSI/NRF, South Africa Center of Excellence in Mathematical and Statistical Sciences (CoE-MaSS) Postdoctoral Fellowship.

REFERENCES

[1] H. Attouch, A. Cabot, Convergence of a relaxed inertial proximal algorithm for maximally monotone operators, Math. Program. 184 (2020), 243-287.

[2] H. Attouch, A. Cabot, Convergence of a relaxed inertial forward–backward algorithm for structured monotone inclusions, Appl. Math. Optim. 80 (2019), 547-598.

[3] H. Attouch, X. Goudon, P. Redont, The heavy ball with friction. I. The continuous dynamical system, Commun. Contemp. Math. 2 (2000) 1-34.

[4] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (2009), 183-202.

[5] A. Bnouhachem, A self-adaptive method for solving general mixed variational inequalities, J. Math. Anal. Appl. 309 (2005), 136-150.

[6] F.E. Browder, Multi-valued monotone nonlinear mappings and duality mappings in Banach spaces, Trans. Am. Math. Soc. 18 (1965), 338-351.

[7] Y. Censor, A. Gibali, S. Reich, The subgradient extragradient method for solving variational inequalities in Hilbert space, J. Optim. Theory Appl. 148 (2011), 318-335.

[8] A. Chambolle, Ch. Dossal, On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm", J. Optim. Theory Appl. 166 (2015), 968-982.

[9] S.Y. Cho, X. Qin, L. Wang, Strong convergence of a splitting algorithm for treating monotone operators, Fixed Point Theory Appl. 2014 (2014), Article ID 94.

[10] S.Y. Cho, A convergence theorem for generalized mixed equilibrium problems and multivalued asymptotically nonexpansive mappings, J. Nonlinear Convex Anal. 21 (2020), 1017-1026.

[11] C. S. Chuang, Hybrid inertial proximal algorithm for the split variational inclusion problem in Hilbert spaces with applications, Optimization, 66 (2017), 777-792.

[12] Q.L. Dong, Y.Y. Lu, J. Yang, S. He, Approximately solving multi-valued variational inequalities by using a projection and contraction algorithm, Numer. Algo. 76 (2017), 799-812.

[13] Q.L. Dong, Y.J. Cho, L.L. Zhong, Th.M. Rassias, Inertial projection and contraction algorithms for variational inequalities, J. Glob. Optim. 70 (2018), 687-704.

[14] F. Facchinei, J.S. Pang, Finite-dimensional variational inequalities and complementarity problems, vol 1 & 2. Springer, Berlin (2003).

[15] C.J. Fang, S.L. Chen, A subgradient extragradient algorithm for solving multi-valued variational inequality, Appl. Math. Comput. 229 (2014), 123-130.

[16] C.J. Fang, Y.R. He, An extragradient method for generalized variational inequality, Pac. J. Optim. 9 (2013), 47-59.

[17] C.J. Fang, Y.R. He, A double projection algorithm for multi-valued variational inequalities and a unified framework of the method, Appl. Math. Comput. 217 (2011), 9543-9511.

[18] S.C. Fang, E.L. Peterson, Generalized variational inequalities, J. Optim. Theory Appl. 38 (1982), 363-383.

[19] F. Iutzeler, J.M. Hendrickx, A generic online acceleration scheme for optimization algorithms via relaxation and inertia, Optim. Methods Softw. 34 (2019), 383-405.

[20] F. Iutzeler, J. Malick, On the proximal gradient algorithm with alternated inertia, J. Optim. Theory Appl. 176 (2018), 688-710.

[21] K. Goebel, S. Reich, Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings, Marcel Dekker, New York, 1984.

[22] N. Hadjisavvas, S. Schaible, Quasimonotone variational inequalities in Banach spaces, J. Optim. Theory Appl. 90 (1996), 95-111.

[23] Y.R. He, A new double projection algorithm for variational inequalities, J. Comput. Appl. Math. 185 (2006), 66-173.

[24] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Appl. Math. Optim. 35 (1997), 69-76.

[25] X. He, N. Huang, X. Li, Modified projection methods for solving multi-valued variational inequality without monotonicity, *Netw. Spat. Econ.*, (2019), https://doi.org/10.1007/s11067-019-09485-2.

[26] D.V. Hieu, J.J. Strodiot, L.D. Muu, An explicit extragradient algorithm for solving variational inequalities, J. Optim. Theory Appl. 185 (2020), 476-503.

[27] I.V. Konnov, A combined relaxation method for variational inequalities with nonlinear constraints, Math. Program. 80 (1998), 239-252.

[28] G.M. Korpelevich, The extragradient method for finding saddle points and other problems, Matecon. 12 (1976), 747-756.

[29] F.L. Li, Y.R. He, An algorithm for generalized variational inequality with pseudomonotone mapping, J. Comput. Appl. Math. 228 (2009), 212-218.

[30] D.A. Lorenz, T. Pock, An inertial forward–backward algorithm for monotone inclusions, J. Math. Imaging Vis. 51 (2015), 311-325.

[31] P.E. Mainge, Convergence theorems for inertial KM-type algorithms, J. Comput. Appl. Math. 219 (2008), 223-236.

[32] A. Moudafi, M. Oliny, Convergence of a splitting inertial proximal method for monotone operators, J. Comput. Appl. Math. 155 (2003), 447-454.

[33] Z. Mu, Y. Peng, A note on the inertial proximal point method, Stat. Optim. Inf. Comput. 3 (2015), 241-248.

[34] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, Soviet Math. Dokl 27 (1983), 372-376.

[35] J.S. Pang, Error bounds in mathematical programming, Math. Program. 79 (1997), 299-332.

[36] B.T. Polyak, Introduction to Optimization, Optimization Software Inc., Publications Division, New York, 1987.

[37] B.T. Polyak, Some methods of speeding up the convergence of iterates methods, U.S.S.R. Comput. Math. and Math. Phys. 4 (1964), 1-17.

[38] X. Qin, L. Wang, J.C. Yao, Inertial splitting method for maximal monotone mappings, J. Nonlinear Convex Anal. 21 (2020), 2325-2333.

[39] R.T. Rockafellar, J.B. Wets, Variational Analysis, Springer, New York, (2004).

[40] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, (1970).

[41] Y. Shehu, X.H. Li, Q.L. Dong, An efficient projection-type method for monotone variational inequalities in Hilbert spaces, Numer. Algo. 84 (2020), 365-388.

[42] M.V. Solodov, Convergence rate analysis of iterative algorithms for solving variational inequality problems, Math. Program. 96 (2003), 513-528.

[43] D.F. Sun, A class of iterative methods for solving nonlinear projection equations, J. Optim. Theory Appl. 91 (1996), 123-140.

[44] D.V. Thong, D.V. Hieu, Inertial subgradient extragradient algorithms with line-search process for solving variational inequality problems and fixed point problems, Numer. Algo. 80 (2019), 1283-1307.

[45] P. Tseng, A modified forward-backward splitting method for maximal monotone mappings, SIAM J. Control Optim. 38 (2000), 431-446.

[46] J.C. Yao, The generalized quasi-variational inequality problem with applications, J. Math. Anal. Appl. 158 (1991), 139-160.

[47] Y. Yang, Self-adaptive inertial subgradient extragradient algorithm for solving pseudomonotone variational inequalities, Appl. Anal. (2019), https://doi.org/10.1080/00036811.2019.1634257.

[48] M.L. Ye, Y.R. He, A double projection method for solving variational inequalities without monotonicity, Comput. Optim. Appl. 60 (2015), 141-150.