OPTIMIZATION
https://doi.org/10.1080/02331934.2022.2123705

Taylor & Francis
Taylor & Francis Group

Modified inertial projection and contraction algorithms with non-monotonic step sizes for solving variational inequalities and their applications

Bing Tan [a,b] and Songxiao Li [a]

[a]Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, People's Republic of China; [b]Department of Mathematics, University of British Columbia, Kelowna, BC, Canada

ABSTRACT

We present two adaptive inertial projection and contraction algorithms to discover the minimum-norm solutions of pseudomonotone variational inequality problems in real Hilbert spaces. The suggested algorithms employ two different step sizes in each iteration and use a non-monotone step size criterion without any line search allowing them to work adaptively. The strong convergence of the iterative sequences formed by the proposed algorithms is established under some mild conditions. Several numerical experiments occurring in finite- and infinite-dimensional Hilbert spaces and applications to optimal control problems as well as signal processing problems are given. Performance profiles are used to verify the computational efficiency and advantages of the proposed algorithms with respect to some known ones.

ARTICLE HISTORY
Received 27 October 2021
Accepted 26 August 2022

KEYWORDS
Variational inequality problem; projection and contraction method; subgradient extragradient method; optimal control problem; signal processing problem

MATHEMATICS SUBJECT CLASSIFICATIONS (2010)
47J20; 47J25; 47J30; 68W10; 65K15

1. Introduction

The goal of this paper is to introduce two new adaptive iterative algorithms for finding solutions of pseudomonotone variational inequality problems in real Hilbert spaces. Let C be a nonempty, closed, and convex subset of a real Hilbert space $\mathcal{H}$ embedded with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$, and $A : C \to \mathcal{H}$ be a nonlinear mapping. Recall that the classical variational inequality problem is stated as follows

$$\text{find } x^* \in C \text{ such that } \langle Ax^*, x - x^* \rangle \geq 0, \quad \forall x \in C. \tag{VIP}$$

The solution set of (VIP) is denoted as $\mathrm{VI}(C, A)$ and is assumed to be nonempty. Variational inequalities construct a unified framework for many optimization problems and have a wide range of application in many fields; see, for example, [1–5].

CONTACT Songxiao Li jyulsx@163.com

In the last few decades, a large number of efficient numerical algorithms were proposed to discover the solutions of variational inequality problems. In this paper we are concerned with projection-based methods. Korpelevich [6] proposed an algorithm (now known as the extragradient method) that requires the computation of the projection on the feasible set twice in each iteration to solve the variational inequality problem (VIP). In order to reduce the projection computation on the feasible set and thus improve the computational efficiency of the extragradient method. Recently, many extensions of the extragradient method, which compute the projection onto the feasible set only once in each iteration, were proposed to solve variational inequality problems; see, for example, the Tseng's extragradient method [7], the projection and contraction method [8], the subgradient extragradient method [9–11], and the projected reflected gradient method [12]. Recently, by combining the advantages of the subgradient extragradient method and the projection and contraction method, Dong et al. [13] proposed a modified subgradient extragradient method using two different step sizes in each iteration. The computational advantages of their method were verified in some basic numerical experiments. However, the method introduced in [13] can only obtain weak convergence in an infinite-dimensional Hilbert space. It is known that strong convergence is preferable to weak convergence in infinite-dimensional spaces (see [14]). Recently, some strongly convergent iterative algorithms were proposed by combining the modified subgradient extragradient method [13] with the Mann method and the voscosity method; see, for instance, [15–19]. On the other hand, the inertial technique have attracted extensive research by scholars as one of the techniques to accelerate the convergence speed of algorithms. The basic idea of inertial-type methods is that the next iteration is determined by the combination of the previous two (or more) iterations. It should be mentioned that Beck and Teboulle [20] proposed a well-known Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) to solve linear inverse problems in image processing, which combines the inertial method with the classical iterative shrinkage thresholding algorithm. FISTA was shown to have a global second-order convergence rate through a clever choice of the inertial coefficient. Moreover, they provide numerical experiments on image deblurring which show that the proposed algorithm converges faster than some known algorithms in the literature. FISTA received a great deal of attention and research from scholars as soon as it came up, and related work can be found at [21,22]. Recently, many inertial-type methods are proposed to solve variational inequalities, equilibrium problems, split feasibility problem, and various optimization problems; see, e.g.[23–29] and the references therein.

Very recently, inspired by the works in [13,15–17], Tan et al. [29] introduced several modified inertial subgradient extragradient methods for solving the variational inequality problem (VIP). Some of their iterative schemes are shown in Algorithms 1.1 and 1.2 below.

Algorithm 1.1 The Algorithm 3.1 of Tan et al. [29]

Initialization: Take $\tau > 0$, $\delta > 0$, $\zeta \in (0,1)$, $\mu \in (0,1)$, and $\sigma \in (0,2)$. Let $x_0, x_1 \in \mathcal{H}$.

Iterative Steps: Given the iterates x_{n-1} and x_n $(n \geq 1)$. Calculate the iterate x_{n+1} as follows:

Step 1. Compute $w_n = x_n + \tau_n(x_n - x_{n-1})$, where

$$\tau_n = \begin{cases} \min\left\{\dfrac{\epsilon_n}{\|x_n - x_{n-1}\|}, \tau\right\}, & \text{if } x_n \neq x_{n-1}; \\ \tau, & \text{otherwise.} \end{cases} \tag{1}$$

Step 2. Compute $y_n = P_C(w_n - \chi_n A w_n)$, where the step size χ_n is chosen to be the largest $\chi \in \{\delta, \delta\zeta, \delta\zeta^2, \ldots\}$ satisfying

$$\chi \|Aw_n - Ay_n\| \leq \mu \|w_n - y_n\|. \tag{2}$$

If $w_n = y_n$, then stop and y_n is a solution of (VIP). Otherwise, go to **Step 3**.

Step 3. Compute $z_n = P_{T_n}(w_n - \sigma \chi_n d_n A y_n)$, where

$$T_n = \left\{x \in \mathcal{H} : \langle w_n - \chi_n A w_n - y_n, x - y_n \rangle \leq 0\right\},$$

and

$$d_n = (1-\mu)\frac{\|w_n - y_n\|^2}{\|\eta_n\|^2}, \quad \eta_n = w_n - y_n - \chi_n(Aw_n - Ay_n). \tag{3}$$

Step 4. Compute $x_{n+1} = (1 - \theta_n - \alpha_n)w_n + \alpha_n z_n$. Set $n := n+1$ and go to **Step 1**.

Algorithm 1.2 The Algorithm 3.4 of Tan et al. [29]

Initialization: Take $\tau > 0$, $\delta > 0$, $\zeta \in (0,1)$, $\mu \in (0,1)$, and $\sigma \in (0,2)$. Let $x_0, x_1 \in \mathcal{H}$.

Iterative Steps: Given the iterates x_{n-1} and x_n $(n \geq 1)$. Calculate the iterate x_{n+1} as follows:

Step 1. Compute $w_n = x_n + \tau_n(x_n - x_{n-1})$, where τ_n is defined in (1).

Step 2. Compute $y_n = P_C(w_n - \chi_n A w_n)$, where the step size χ_n is chosen to be the largest $\chi \in \{\delta, \delta\zeta, \delta\zeta^2, \ldots\}$ satisfying (2). If $w_n = y_n$, then stop and y_n is a solution of (VIP). Otherwise, go to **Step 3**.

Step 3. Compute $z_n = w_n - \sigma d_n \eta_n$, where d_n and η_n are defined in (2).

Step 4. Compute $x_{n+1} = (1 - \theta_n - \alpha_n)w_n + \alpha_n z_n$. Set $n := n+1$ and go to **Step 1**.

Under some suitable conditions, the strong convergence of Algorithms 1.1 and 1.2 is established. Notice that the Armijo-type step criterion (2) used in Algorithms 1.1 and 1.2 leads to the evaluation of the projection on the feasible set and the value of the operator A multiple times in each iteration. This can further affect the computational efficiency of these methods. To overcome this drawback, some adaptive step size criteria that update the step size of each iteration with a simple computation using some previously known information are introduced; see, e.g. the works in [25,30–34]. It is worth noting that the methods presented in [25,30,32,33] generate a non-increasing sequence of step sizes, which may affect the efficiency of this type of algorithms. On the other hand, the pseudomonotone mappings are widely studied by scholars as a broader class of mappings than the monotone mappings. Recently, a large number of numerical methods were introduced to solve pseudomonotone variational inequality problems in real Hilbert spaces; see, e.g.[15,18,25,29] and the references therein.

Inspired by the above work and some recent ongoing efforts in this area, we introduce in this paer a more general step size criterion than the methods in [19,25,30–32] and propose two adaptively modified inertial subgradient extragradient algorithms without any line search procedure for solving pseudomonotone variational inequalities in real Hilbert spaces. Our contributions to this paper are summarized below.

(1) Notice that the algorithms presented in [9,13,23,25] can only obtain weak convergence in an infinite-dimensional Hilbert space. The strong convergence theorems of the proposed algorithms are established in real Hilbert spaces (see Theorems 3.1 and 3.2). Therefore, the iterative schemes proposed in this paper are preferable to the weak convergence results in the literature in infinite-dimensional Hilbert spaces.
(2) Note that the prerequisite for availability of fixed-step algorithms introduced in [15,18,23] is that the prior information of the Lipschitz constant of the mapping needs to be known. This paper introduces a new non-monotonic step size criterion (8) without any line search procedure. Our algorithms with the step size rule (8) can work without the prior knowledge of the Lipschitz constant of the operator. The computational efficiency of our algorithms under different step size choices is shown in Section 4.
(3) We introduce a new parameter β to modify the subgradient extragradient method presented in [9–11] and the projection and contraction method introduced in [8]. Moreover, our algorithms also incorporate an inertial term to speed up the convergence of the algorithms. Numerical experiments will show that suitable parameters τ_n and β have a significant improvement on the convergence speed and accuracy of our algorithms (see Section 4).
(4) Our two algorithms can solve pseudomonotone variational inequalities, which extends many results in the literature (e.g.[13,16,17,23]) for solving monotone variational inequalities.

(5) The computational efficiency and stability of our algorithms compared to the ones in the literature [19,29,34] are verified by using the performance profiles introduced by Dolan and Moré [35]; see Sections 4 and 5 for more details.

The rest of this paper is organized as follows. In the next section, we collect some important definitions and lemmas for further use. In Section 3, we introduce two iterative schemes with non-monotonic step sizes to solve variational inequality problems and analyze their convergence. We provide three examples of constrained and unconstrained variational inequality problems in Section 4 and two practical applications in Section 5 to verify the computational efficiency and stability of the algorithms proposed in this paper. Finally, we conclude the paper in Section 6, the last section.

2. Preliminaries

In this section, we give some definitions and lemmas that need to be used in the sequel. Recall that an operator $A : \mathcal{H} \to \mathcal{H}$ is said to be:

(1) *L-Lipschitz continuous* with $L > 0$ if $\|Ax - Ay\| \leq L\|x - y\|$, $\forall x, y \in \mathcal{H}$;
(2) *monotone* if $\langle Ax - Ay, x - y\rangle \geq 0$, $\forall x, y \in \mathcal{H}$;
(3) *pseudomonotone* if $\langle Ax, y - x\rangle \geq 0 \Rightarrow \langle Ay, y - x\rangle \geq 0$, $\forall x, y \in \mathcal{H}$;
(4) *sequentially weakly continuous* if for each sequence $\{x_n\}$ converges weakly to x implies that $\{Ax_n\}$ converges weakly to Ax.

The weak convergence and strong convergence of $\{x_n\}$ to x are represented by $x_n \rightharpoonup x$ and $x_n \to x$, respectively. For each $x, y \in \mathcal{H}$, we have the following inequality

$$\|x + y\|^2 \leq \|x\|^2 + 2\langle y, x + y\rangle. \tag{4}$$

For every point $x \in \mathcal{H}$, there exists a unique nearest point in C, denoted by $P_C(x)$, such that $P_C(x) = \arg\min\{\|x - y\|, y \in C\}$. P_C is called the *metric projection* of $\mathcal{H}$ onto C. It is known that P_C has the following basic properties (see, e.g. Chapter 20 of the monograph by Bauschke and Combettes [36]):

$$\langle x - P_C(x), y - P_C(x)\rangle \leq 0, \quad \forall x \in \mathcal{H}, \forall y \in C, \tag{5}$$

and

$$\|P_C(x) - P_C(y)\|^2 \leq \langle P_C(x) - P_C(y), x - y\rangle, \quad \forall x, y \in \mathcal{H}. \tag{6}$$

The projection can be computed explicitly in some special closed and convex sets such as half spaces, balls, and box constraints; see, e.g.[36, Chapter 29] and [37].

The following two lemmas are useful for our main results.

Lemma 2.1 ([38, Lemma 1]): *Let* $\{\chi_n\}$, $\{\xi_n\}$ *and* $\{\rho_n\}$ *be sequences of nonnegative numbers such that*

$$\chi_{n+1} \leq \xi_n \chi_n + \rho_n, \quad \forall n \geq 1.$$

If $\{\xi_n\} \subset [1, +\infty)$, $\sum_{n=1}^{\infty}(\xi_n - 1) < \infty$, *and* $\sum_{n=1}^{\infty} \rho_n < \infty$, *then* $\lim_{n\to\infty} \chi_n$ *exists.*

Lemma 2.2 ([39, Lemma 2.6]): *Let* $\{x_n\}$ *be a positive sequence,* $\{r_n\}$ *be a sequence of real numbers, and* $\{\theta_n\}$ *be a sequence in* $(0, 1)$ *such that* $\sum_{n=1}^{\infty} \theta_n = \infty$. *Suppose that*

$$x_{n+1} \leq (1 - \theta_n) x_n + \theta_n r_n, \quad \forall n \geq 1.$$

If $\limsup_{k\to\infty} r_{n_k} \leq 0$ *for any subsequence* $\{x_{n_k}\}$ *of* $\{x_n\}$ *satisfying* $\liminf_{k\to\infty} (x_{n_k+1} - x_{n_k}) \geq 0$, *then* $\lim_{n\to\infty} x_n = 0$.

3. Main results

In this section, we introduce two new modified inertial extragradient methods for finding the minimum-norm solutions of pseudomonotone variational inequalities in real Hilbert spaces. The advantage of the suggested algorithms is that they can work without the prior information of the Lipschitz constant of the operator. First, we assume that our iterative schemes satisfy the following conditions in order to perform their convergence analysis.

(C1) The feasible set C is a nonempty, closed, and convex subset of a real Hilbert space $\mathcal{H}$, and the solution set of the (VIP) is nonempty, i.e. $\mathrm{VI}(C, A) \neq \emptyset$.

(C2) The mapping $A : \mathcal{H} \to \mathcal{H}$ is pseudomonotone, L-Lipschitz continuous on $\mathcal{H}$, and sequentially weakly continuous on C.

(C3) Assume $\chi_1 > 0$, $\mu \in (0, 1)$, $\{\delta_n\} \subset [1, \infty)$ such that $\lim_{n\to\infty} \delta_n = 1$, $\{\xi_n\} \subset [1, \infty)$ such that $\sum_{n=0}^{\infty}(\xi_n - 1) < \infty$, and $\{\rho_n\} \subset [0, \infty)$ such that $\sum_{n=0}^{\infty} \rho_n < \infty$.

(C4) Let $\{\epsilon_n\}$ be a positive sequence such that $\lim_{n\to\infty} \frac{\epsilon_n}{\theta_n} = 0$, where $\{\theta_n\} \subset (0, 1)$ satisfies $\lim_{n\to\infty} \theta_n = 0$ and $\sum_{n=1}^{\infty} \theta_n = \infty$.

Now, we are in a position to state our Algorithm 3.1.

Remark 3.1: We have the following comments on the proposed Algorithm 3.1.

(i) Note that the inertial criterion (7) is easy to implement because $\|x_n - x_{n-1}\|$ is known before the calculation of τ_n. Furthermore, the parameter τ in (7) can be either a positive constant or a positive sequence; special attention is drawn to the case $\tau = \frac{n-1}{n+\alpha-1}$, where $\alpha > 0$ (see [19,21–23] for more details).

Algorithm 3.1 The modified subgradient extragradient method with non-monotonic step sizes

Initialization: Take $\tau > 0$, $\chi_1 > 0$, $\mu \in (0,1)$, $\sigma \in (0, 2/\mu)$, and $\beta \in (\sigma/2, 1/\mu)$. Choose $\{\epsilon_n\}$, $\{\delta_n\}$, $\{\xi_n\}$, $\{\rho_n\}$, and $\{\theta_n\}$ satisfies Conditions (C3) and (C4). Let $x_0, x_1 \in \mathcal{H}$.

Iterative Steps: Given the iterates x_{n-1} and x_n $(n \geq 1)$. Calculate the iterate x_{n+1} as follows:

Step 1. Compute $w_n = (1-\theta_n)(x_n + \tau_n(x_n - x_{n-1}))$, where

$$\tau_n = \begin{cases} \min\left\{\dfrac{\epsilon_n}{\|x_n - x_{n-1}\|}, \tau\right\}, & \text{if } x_n \neq x_{n-1}; \\ \tau, & \text{otherwise.} \end{cases} \tag{7}$$

Step 2. Compute $y_n = P_C(w_n - \beta\chi_n A w_n)$, where the next step size χ_{n+1} is updated by

$$\chi_{n+1} = \begin{cases} \min\left\{\dfrac{\mu\delta_n \|w_n - y_n\|}{\|Aw_n - Ay_n\|}, \xi_n\chi_n + \rho_n\right\}, & \text{if } Aw_n \neq Ay_n; \\ \xi_n\chi_n + \rho_n, & \text{otherwise.} \end{cases} \tag{8}$$

If $w_n = y_n$, then stop and y_n is a solution of (VIP). Otherwise, go to **Step 3**.

Step 3. Compute $x_{n+1} = P_{H_n}(w_n - \sigma\chi_n d_n A y_n)$, where

$$H_n = \left\{x \in \mathcal{H} : \langle w_n - \beta\chi_n A w_n - y_n, x - y_n\rangle \leq 0\right\},$$

and

$$d_n = \frac{\langle w_n - y_n, \eta_n\rangle}{\|\eta_n\|^2}, \quad \eta_n = w_n - y_n - \beta\chi_n(Aw_n - Ay_n). \tag{9}$$

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

(ii) It is not necessary to impose the sequentially weak continuity of the operator A in Condition (C2) when A is monotonic (see [40]).

(iii) It is easy to verify that Conditions (C3) and (C4) can be satisfied; for example, take

$$\delta_n = 1 + \frac{1}{(n+1)^a}\ (a \geq 1), \quad \xi_n = 1 + \frac{1}{(n+1)^b}\ (b > 1),$$
$$\rho_n = \frac{1}{(n+1)^c}\ (c > 1), \quad \theta_n = \frac{1}{(n+1)^d}\ (0 < d \leq 1),$$
$$\epsilon_n = \frac{1}{(n+1)^e}\ (e > d).$$

(iv) Note that $C \subset H_n$. Indeed, combining the definition of y_n and (5), we have

$$\langle w_n - \beta\chi_n A w_n - y_n, x - y_n\rangle \leq 0, \quad \forall x \in C.$$

This together with the definition of H_n gives the required conclusion.

The following lemma demonstrates that the step size criterion (8) is well defined.

Lemma 3.1: *Suppose that Condition* (C3) *holds. Then the step size sequence* $\{\chi_n\}$ *formed by* (8) *is well defined and* $\lim_{n\to\infty}\chi_n$ *exists.*

Proof: Since mapping A is L-Lipschitz continuous and $\delta_n \geq 1$, one has

$$\frac{\mu\delta_n\|w_n - y_n\|}{\|Aw_n - Ay_n\|} \geq \frac{\mu\delta_n\|w_n - y_n\|}{L\|w_n - y_n\|} \geq \frac{\mu}{L}.$$

Note that $\xi_n \geq 1$ and $\theta_n > 0$. Thus

$$\chi_{n+1} = \min\left\{\frac{\mu\delta_n\left\|w_n - y_n\right\|}{\left\|Aw_n - Ay_n\right\|}, \xi_n\chi_n + \rho_n\right\} \geq \min\left\{\frac{\mu}{L}, \chi_n\right\}.$$

By induction, we obtain that the sequence $\{\chi_n\}$ has a lower bound $\{\mu/L, \chi_1\}$. From the definition of (8), one sees that $\chi_{n+1} \leq \xi_n\chi_n + \rho_n$, which together with Condition (C3), in the light of Lemma 2.1, implies that $\lim_{n\to\infty}\chi_n$ exists. This completes the proof. ■

Remark 3.2: We show that if $w_n = y_n$ or $\eta_n = 0$ in Algorithm 3.1, then $y_n \in \mathrm{VI}(C, A)$. From the definition of η_n and (8), one has

$$\begin{aligned}\|\eta_n\| &\geq \|w_n - y_n\| - \beta\chi_n\|Aw_n - Ay_n\| \\ &\geq \left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\|w_n - y_n\|.\end{aligned}$$

We can also show that

$$\|\eta_n\| \leq \left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\|w_n - y_n\|. \tag{10}$$

Therefore we conclude that

$$\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\|w_n - y_n\| \leq \|\eta_n\| \leq \left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\|w_n - y_n\|.$$

It follows from Lemma 3.1 that $\lim_{n\to\infty}\chi_n$ exists, which together with $\lim_{n\to\infty}\delta_n = 1$ gives

$$\lim_{n\to\infty}\frac{\delta_n\chi_n}{\chi_{n+1}} = 1.$$

Consequently, there exists a constant n_0 such that $1-\frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}>0$ for all $n\geq n_0$ (noting that $\beta<1/\mu$). Hence we have that $w_n=y_n$ if and only if $\eta_n=0$. If $w_n=y_n$, then $y_n=P_C(y_n-\beta\chi_nAy_n)$. This means that $y_n\in \mathrm{VI}(C,A)$ by means of (5).

The following lemmas are very helpful in analyzing the convergence of Algorithm 3.1.

Lemma 3.2: *Suppose that Conditions* (C1) *and* (C2) *hold. Let* $\{w_n\}$ *and* $\{y_n\}$ *be two sequences generated by Algorithm* 3.1. *If there exists a subsequence* $\{w_{n_k}\}$ *of* $\{w_n\}$ *such that* $\{w_{n_k}\}$ *converges weakly to* $z\in\mathcal{H}$ *and* $\lim_{k\to\infty}\|w_{n_k}-y_{n_k}\|=0$, *then* $z\in\mathrm{VI}(C,A)$.

Proof: The proof follows the proof in [41, Lemma 3.3] and thus it is omitted. ■

Lemma 3.3: *Suppose that Conditions* (C1) *and* (C2) *hold. Let* $\{x_{n+1}\}$, $\{y_n\}$ *and* $\{w_n\}$ *be three sequences formed by Algorithm* 3.1. *Then, for every* $x^*\in\mathrm{VI}(C,A)$, *there exists* $n_0>0$ *such that*

$$\begin{aligned}\|x_{n+1}-x^*\|^2\leq{}&\|w_n-x^*\|^2-\|w_n-x_{n+1}-\frac{\sigma}{\beta}d_n\eta_n\|^2\\&-\frac{\sigma}{\beta^2}(2\beta-\sigma)\frac{\left(1-\frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}{\left(1+\frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}\|w_n-y_n\|^2,\quad\forall n\geq n_0.\end{aligned}$$

Proof: From the definition of x_{n+1}, the property of projection (6), and $x^*\in\mathrm{VI}(C,A)\subset C\subset H_n$, we have

$$\begin{aligned}&2\|x_{n+1}-x^*\|^2\\&\quad=2\|P_{H_n}(w_n-\sigma\chi_nd_nAy_n)-P_{H_n}(x^*)\|^2\\&\quad\leq 2\langle x_{n+1}-x^*,w_n-\sigma\chi_nd_nAy_n-x^*\rangle\\&\quad=\|x_{n+1}-x^*\|^2+\|w_n-\sigma\chi_nd_nAy_n-x^*\|^2-\|x_{n+1}-w_n+\sigma\chi_nd_nAy_n\|^2\\&\quad=\|x_{n+1}-x^*\|^2+\|w_n-x^*\|^2+\sigma^2\chi_n^2d_n^2\|Ay_n\|^2-2\langle w_n-x^*,\sigma\chi_nd_nAy_n\rangle\\&\qquad-\|x_{n+1}-w_n\|^2-\sigma^2\chi_n^2d_n^2\|Ay_n\|^2-2\langle x_{n+1}-w_n,\sigma\chi_nd_nAy_n\rangle\\&\quad=\|x_{n+1}-x^*\|^2+\|w_n-x^*\|^2-\|x_{n+1}-w_n\|^2-2\langle x_{n+1}-x^*,\sigma\chi_nd_nAy_n\rangle.\end{aligned}$$

Thus

$$\|x_{n+1}-x^*\|^2\leq\|w_n-x^*\|^2-\|x_{n+1}-w_n\|^2-2\sigma\chi_nd_n\langle x_{n+1}-x^*,Ay_n\rangle.\tag{11}$$

In view of $x^*\in\mathrm{VI}(C,A)$ and $y_n\in C$, one obtains $\langle Ax^*,y_n-x^*\rangle\geq 0$, which combining with the pseudomonotonicity of mapping A gives that $\langle Ay_n,y_n-x^*\rangle\geq 0$.

This is equivalent to

$$\langle Ay_n, x_{n+1} - x^* \rangle \geq \langle Ay_n, x_{n+1} - y_n \rangle. \tag{12}$$

Note that $d_n > 0$ for all $n \geq n_0$. Indeed, by the definitions of d_n, η_n, and (8), we have

$$\begin{aligned} d_n = \frac{\langle w_n - y_n, \eta_n \rangle}{\|\eta_n\|^2} &= \frac{\|w_n - y_n\|^2 - \langle w_n - y_n, \beta \chi_n (Aw_n - Ay_n) \rangle}{\|\eta_n\|^2} \\ &\geq \frac{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right) \|w_n - y_n\|^2}{\|\eta_n\|^2}. \end{aligned} \tag{13}$$

It follows from Remark 3.2 that $1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}} > 0$ for all $n \geq n_0$. Combining (10) and (13), we deduce

$$d_n \geq \frac{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)}{\left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2} > 0, \quad \forall n \geq n_0. \tag{14}$$

From (12) and (14) (noting that $\sigma \in (0, 2/\mu)$), one sees that

$$-2\sigma\chi_n d_n \langle Ay_n, x_{n+1} - x^* \rangle \leq -2\sigma\chi_n d_n \langle Ay_n, x_{n+1} - y_n \rangle. \tag{15}$$

Using the definition of H_n and $x_{n+1} \in H_n$, one obtains

$$\langle w_n - \beta\chi_n Aw_n - y_n, x_{n+1} - y_n \rangle \leq 0.$$

This shows that

$$\langle w_n - y_n - \beta\chi_n (Aw_n - Ay_n), x_{n+1} - y_n \rangle \leq \beta\chi_n \langle Ay_n, x_{n+1} - y_n \rangle. \tag{16}$$

From the definitions of η_n, d_n, (15), and (16), we have

$$\begin{aligned} -2\sigma\chi_n d_n \langle Ay_n, x_{n+1} - x^* \rangle &\leq -2\frac{\sigma}{\beta} d_n \langle \eta_n, x_{n+1} - y_n \rangle \\ &= -2\frac{\sigma}{\beta} d_n \langle \eta_n, w_n - y_n \rangle + 2\frac{\sigma}{\beta} d_n \langle \eta_n, w_n - x_{n+1} \rangle \\ &= -2\frac{\sigma}{\beta} d_n^2 \|\eta_n\|^2 + 2\frac{\sigma}{\beta} d_n \langle \eta_n, w_n - x_{n+1} \rangle. \end{aligned} \tag{17}$$

Now, we estimate $2\frac{\sigma}{\beta} d_n \langle \eta_n, w_n - x_{n+1} \rangle$. According to the formula $2ab = a^2 + b^2 - (a-b)^2$, we arrive

$$\begin{aligned} &2\frac{\sigma}{\beta} d_n \langle \eta_n, w_n - x_{n+1} \rangle \\ &\quad = \|w_n - x_{n+1}\|^2 + \frac{\sigma^2}{\beta^2} d_n^2 \|\eta_n\|^2 - \left\| w_n - x_{n+1} - \frac{\sigma}{\beta} d_n \eta_n \right\|^2. \end{aligned} \tag{18}$$

It follows from (13) that

$$d_n\|\eta_n\|^2 \geq \left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\left\|w_n - y_n\right\|^2.$$

This together with (10) yields

$$d_n^2\|\eta_n\|^2 \geq \left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2 \frac{\|w_n - y_n\|^4}{\|\eta_n\|^2} \geq \frac{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}{\left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}\|w_n - y_n\|^2. \quad (19)$$

Combining (11), (17), (18), and (19), we conclude that

$$\begin{aligned}\|x_{n+1} - x^*\|^2 \leq \|w_n - x^*\|^2 - \left\|w_n - x_{n+1} - \frac{\sigma}{\beta}d_n\eta_n\right\|^2 \\ - \frac{\sigma}{\beta^2}(2\beta - \sigma)\frac{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}{\left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}\|w_n - y_n\|^2, \ \forall n \geq n_0.\end{aligned}$$

This completes the proof. ■

Now, we are ready to analyze the strong convergence of Algorithm 3.1.

Theorem 3.1: *Suppose that Conditions* (C1)–(C4) *hold. Then the sequence* $\{x_n\}$ *formed by Algorithm* 3.1 *converges strongly to* $x^* \in \mathrm{VI}(C, A)$*, where* $\|x^*\| = \min\{\|z\| : z \in \mathrm{VI}(C, A)\}$.

Proof: First, we show that the sequence $\{x_n\}$ is bounded. Note that $2\beta - \sigma > 0$. It follows from Lemma 3.3 that

$$\|x_{n+1} - x^*\| \leq \|w_n - x^*\|, \quad \forall n \geq n_0. \quad (20)$$

By the definition of w_n, one has

$$\begin{aligned}\left\|w_n - x^*\right\| &= \left\|(1-\theta_n)\left(x_n + \tau_n\left(x_n - x_{n-1}\right)\right) - x^*\right\| \\ &= \left\|(1-\theta_n)\left(x_n - x^*\right) + (1-\theta_n)\tau_n\left(x_n - x_{n-1}\right) - \theta_n x^*\right\| \\ &\leq (1-\theta_n)\left\|x_n - x^*\right\| + (1-\theta_n)\tau_n\left\|x_n - x_{n-1}\right\| + \theta_n\|x^*\| \\ &= (1-\theta_n)\left\|x_n - x^*\right\| + \theta_n\left[(1-\theta_n)\frac{\tau_n}{\theta_n}\left\|x_n - x_{n-1}\right\| + \|x^*\|\right].\end{aligned} \quad (21)$$

From (7), one sees that $\tau_n\|x_n - x_{n-1}\| \leq \epsilon_n, \forall n \geq 1$, which together with $\lim_{n\to\infty}\frac{\epsilon_n}{\theta_n} = 0$ implies that

$$\lim_{n\to\infty}\frac{\tau_n}{\theta_n}\|x_n - x_{n-1}\| \leq \lim_{n\to\infty}\frac{\epsilon_n}{\theta_n} = 0. \quad (22)$$

Thus we have

$$\lim_{n\to\infty}\left[(1-\theta_n)\frac{\tau_n}{\theta_n}\|x_n-x_{n-1}\|+\|x^*\|\right]=\|x^*\|. \tag{23}$$

Therefore, there exists a constant $Q_1>0$ such that

$$(1-\theta_n)\frac{\tau_n}{\theta_n}\|x_n-x_{n-1}\|+\|x^*\|\le Q_1,\quad \forall n\ge 1.$$

It follows from (21) and (23) that

$$\|w_n-x^*\|\le(1-\theta_n)\|x_n-x^*\|+\theta_n Q_1,\quad \forall n\ge 1. \tag{24}$$

Combining (20) and (24), we have

$$\begin{aligned}\|x_{n+1}-x^*\|&\le(1-\theta_n)\|x_n-x^*\|+\theta_n Q_1\\&\le\max\{\|x_n-x^*\|,Q_1\},\quad \forall n\ge n_0\\&\le\cdots\le\max\{\|x_{n_0}-x^*\|,Q_1\}.\end{aligned}$$

That is, the sequence $\{x_n\}$ is bounded. So are $\{w_n\}$ and $\{y_n\}$.

By (24), one has

$$\begin{aligned}\|w_n-x^*\|^2&\le\left[(1-\theta_n)\|x_n-x^*\|+\theta_n Q_1\right]^2\\&=(1-\theta_n)^2\|x_n-x^*\|^2+\theta_n\left[2(1-\theta_n)Q_1\|x_n-x^*\|+\theta_n Q_1^2\right]\\&\le\|x_n-x^*\|^2+\theta_n Q_2,\end{aligned} \tag{25}$$

where $Q_2:=\sup_{n\in\mathbb{N}}\{2(1-\theta_n)Q_1\|x_n-x^*\|+\theta_n Q_1^2\}>0$. Combining Lemma 3.3 and (25), we deduce

$$\begin{aligned}&\|w_n-x_{n+1}-\frac{\sigma}{\beta}d_n\eta_n\|^2+\frac{\sigma}{\beta^2}(2\beta-\sigma)\frac{\left(1-\frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}{\left(1+\frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}\|w_n-y_n\|^2\\&\le\|x_n-x^*\|^2-\|x_{n+1}-x^*\|^2+\theta_n Q_2,\quad \forall n\ge n_0.\end{aligned} \tag{26}$$

From the definition of w_n, (4), and (20), we have

$$\begin{aligned}&\|x_{n+1}-x^*\|^2\\&\quad\le\|w_n-x^*\|^2\\&\quad=\|(1-\theta_n)(x_n-x^*)+(1-\theta_n)\tau_n(x_n-x_{n-1})-\theta_n x^*\|^2\end{aligned}$$

$$
\begin{aligned}
&\leq \|(1-\theta_n)(x_n - x^*) + (1-\theta_n)\tau_n(x_n - x_{n-1})\|^2 + 2\theta_n \left\langle -x^*, w_n - x^* \right\rangle \\
&\leq (1-\theta_n)^2\|x_n - x^*\|^2 + 2(1-\theta_n)\tau_n\|x_n - x^*\|\|x_n - x_{n-1}\| \\
&\quad + \tau_n^2\|x_n - x_{n-1}\|^2 + 2\theta_n \left\langle -x^*, w_n - x_{n+1} \right\rangle + 2\theta_n \left\langle -x^*, x_{n+1} - x^* \right\rangle \\
&\leq (1-\theta_n)\|x_n - x^*\|^2 + \theta_n \Bigg[2(1-\theta_n)\|x_n - x^*\| \frac{\tau_n}{\theta_n}\|x_n - x_{n-1}\| \\
&\quad + \tau_n\|x_n - x_{n-1}\| \frac{\tau_n}{\theta_n}\|x_n - x_{n-1}\| + 2\|x^*\|\|w_n - x_{n+1}\| \\
&\quad + 2\langle x^*, x^* - x_{n+1} \rangle \Bigg], \quad \forall n \geq n_0. \qquad (27)
\end{aligned}
$$

Finally, we show that $\{\|x_n - x^*\|\}$ converges to zero. By Lemma 2.2, we assume that $\{\|x_{n_k} - x^*\|^2\}$ is a subsequence of $\{\|x_n - x^*\|^2\}$ such that

$$
\liminf_{k\to\infty} \left(\|x_{n_k+1} - x^*\|^2 - \|x_{n_k} - x^*\|^2\right) \geq 0. \qquad (28)
$$

Note that $\sigma \in (0, 2/\mu)$ and $\beta \in (\sigma/2, 1/\mu)$. Combining (26), (28), and Condition (C4), we obtain

$$
\begin{aligned}
&\limsup_{k\to\infty} \left(\frac{\sigma}{\beta^2}(2\beta - \sigma) \frac{\left(1 - \frac{\beta\mu\delta_{n_k}\chi_{n_k}}{\chi_{n_k+1}}\right)^2}{\left(1 + \frac{\beta\mu\delta_{n_k}\chi_{n_k}}{\chi_{n_k+1}}\right)^2} \|w_{n_k} - y_{n_k}\|^2 + \|w_{n_k} - x_{n_k+1} - \frac{\sigma}{\beta} d_{n_k}\eta_{n_k}\|^2 \right) \\
&\leq \limsup_{k\to\infty} \left[\|x_{n_k} - x^*\|^2 - \|x_{n_k+1} - x^*\|^2\right] + \limsup_{k\to\infty} \theta_{n_k} Q_2 \\
&= -\liminf_{k\to\infty} \left[\|x_{n_k+1} - x^*\|^2 - \|x_{n_k} - x^*\|^2\right] \leq 0,
\end{aligned}
$$

which implies that

$$
\lim_{k\to\infty} \|y_{n_k} - w_{n_k}\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \left\| w_{n_k} - x_{n_k+1} - \frac{\sigma}{\beta} d_{n_k}\eta_{n_k} \right\| = 0.
$$

From the definition of d_n, we obtain

$$
\begin{aligned}
\|w_{n_k} - x_{n_k+1}\| &\leq \left\| w_{n_k} - x_{n_k+1} - \frac{\sigma}{\beta} d_{n_k}\eta_{n_k} \right\| + \frac{\sigma}{\beta} d_{n_k}\|\eta_{n_k}\| \\
&= \left\| w_{n_k} - x_{n_k+1} - \frac{\sigma}{\beta} d_{n_k}\eta_{n_k} \right\| + \frac{\sigma}{\beta} \frac{\left\langle w_{n_k} - y_{n_k}, \eta_{n_k} \right\rangle}{\|\eta_{n_k}\|} \\
&\leq \left\| w_{n_k} - x_{n_k+1} - \frac{\sigma}{\beta} d_{n_k}\eta_{n_k} \right\| + \frac{\sigma}{\beta}\|w_{n_k} - y_{n_k}\|.
\end{aligned}
$$

Hence we have that $\lim_{k\to\infty} \|x_{n_k+1} - w_{n_k}\| = 0$. This together with the boundedness of $\{x_n\}$ gives

$$
\lim_{k\to\infty} \|w_{n_k} - x_{n_k+1}\|\|x^*\| = 0. \qquad (29)
$$

It follows from the definition of w_n that

$$\begin{aligned}\|x_{n_k}-w_{n_k}\| &= \|(1-\theta_{n_k})\tau_{n_k}(x_{n_k}-x_{n_k-1})-\theta_{n_k}x_{n_k}\| \\ &\le \|(1-\theta_{n_k})\tau_{n_k}(x_{n_k}-x_{n_k-1})\|+\|\theta_{n_k}x_{n_k}\| \\ &= \theta_{n_k}\left[(1-\theta_{n_k})\frac{\tau_{n_k}}{\theta_{n_k}}\|x_{n_k}-x_{n_k-1}\|+\|x_{n_k}\|\right],\end{aligned}$$

which combining with (22) and Condition (C4) means that $\lim_{k\to\infty}\|x_{n_k}-w_{n_k}\|=0$. Moreover, one finds that

$$\lim_{k\to\infty}\|x_{n_k+1}-x_{n_k}\| \le \lim_{k\to\infty}\|x_{n_k+1}-w_{n_k}\|+\lim_{k\to\infty}\|w_{n_k}-x_{n_k}\|.$$

From the above facts, we deduce that

$$\lim_{k\to\infty}\|x_{n_k+1}-x_{n_k}\|=0. \tag{30}$$

Since the sequence $\{x_{n_k}\}$ is bounded, there exists a subsequence $\{x_{n_{k_j}}\}$ of $\{x_{n_k}\}$ such that $x_{n_{k_j}} \rightharpoonup z$ when $j\to\infty$. Furthermore,

$$\limsup_{k\to\infty}\left\langle x^*, x^*-x_{n_k}\right\rangle = \lim_{j\to\infty}\langle x^*, x^*-x_{n_{k_j}}\rangle = \langle x^*, x^*-z\rangle. \tag{31}$$

Since $\lim_{k\to\infty}\|x_{n_k}-w_{n_k}\|=0$, one obtains $w_{n_k}\rightharpoonup z$. This together with $\lim_{k\to\infty}\|w_{n_k}-y_{n_k}\|=0$, in the light of Lemma 3.2, gives that $z\in \mathrm{VI}(C,A)$. From the definition of x^*, (5), and (31), we have

$$\limsup_{k\to\infty}\left\langle x^*, x^*-x_{n_k}\right\rangle = \langle x^*, x^*-z\rangle \le 0. \tag{32}$$

By (30) and (32), we obtain

$$\limsup_{k\to\infty}\left\langle x^*, x^*-x_{n_k+1}\right\rangle \le \limsup_{k\to\infty}\left\langle x^*, x^*-x_{n_k}\right\rangle \le 0. \tag{33}$$

Combining (22), (27), (29), (33), and Lemma 2.2, we conclude that $x_n\to x^*$ as $n\to\infty$. The proof is completed. ■

Next, we present a new modified inertial projection and contraction method (see Algorithm 3.2 below) for finding the minimum-norm solutions of pseudomonotone variational inequalities in real Hilbert spaces.

Algorithm 3.2 The modified projection and contraction method with non-monotonic step sizes

Initialization: Take $\tau > 0$, $\chi_1 > 0$, $\mu \in (0,1)$, $\sigma \in (0,2)$, and $\beta \in (0, 1/\mu)$. Choose $\{\epsilon_n\}$, $\{\delta_n\}$, $\{\xi_n\}$, $\{\rho_n\}$, and $\{\theta_n\}$ satisfies Conditions (C3) and (C4). Let $x_0, x_1 \in \mathcal{H}$.

Iterative Steps: Given the iterates x_{n-1} and x_n ($n \geq 1$). Calculate the iterate x_{n+1} as follows:

Step 1. Compute $w_n = (1-\theta_n)(x_n + \tau_n(x_n - x_{n-1}))$, where τ_n is defined in (7).

Step 2. Compute $y_n = P_C(w_n - \beta\chi_n A w_n)$, where the next step size χ_{n+1} is updated by (8). If $w_n = y_n$, then stop and y_n is a solution of (VIP). Otherwise, go to **Step 3**.

Step 3. Compute $x_{n+1} = w_n - \sigma d_n \eta_n$, where d_n and η_n are defined in (9). Set $n := n+1$ and go to **Step 1**.

The following lemma plays a crucial role in studying the convergence of Algorithm 3.2.

Lemma 3.4: *Suppose that Conditions (C1) and (C2) hold. Let $\{w_n\}$, $\{y_n\}$, and $\{x_{n+1}\}$ be three sequences generated by Algorithm 3.2. Then*

$$\|x_{n+1} - x^*\|^2 \leq \|w_n - x^*\|^2 - \frac{2-\sigma}{\sigma}\|w_n - x_{n+1}\|^2, \quad \forall x^* \in \mathrm{VI}(C, A),$$

and

$$\|w_n - y_n\|^2 \leq \left[\frac{\left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)}{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\sigma}\right]^2 \|w_n - x_{n+1}\|^2.$$

Proof: It follows from the definition of x_{n+1} that

$$\begin{aligned}\|x_{n+1} - x^*\|^2 &= \|w_n - \sigma d_n \eta_n - x^*\|^2 \\ &= \|w_n - x^*\|^2 - 2\sigma d_n \langle w_n - x^*, \eta_n\rangle + \sigma^2 d_n^2 \|\eta_n\|^2. \end{aligned} \tag{34}$$

By the definition of η_n, one sees that

$$\begin{aligned}\langle w_n - x^*, \eta_n\rangle &= \langle w_n - y_n, \eta_n\rangle + \langle y_n - x^*, \eta_n\rangle \\ &= \langle w_n - y_n, \eta_n\rangle + \langle y_n - x^*, w_n - y_n - \beta\chi_n(Aw_n - Ay_n)\rangle. \end{aligned} \tag{35}$$

In view of $y_n = P_C(w_n - \beta\chi_n A w_n)$ and (5), we have

$$\langle w_n - y_n - \beta\chi_n A w_n, y_n - x^*\rangle \geq 0. \tag{36}$$

By using $x^* \in \mathrm{VI}(C,A)$, $y_n \in C$, and the pseudomonotonicity of the mapping A, we deduce

$$\langle Ay_n, y_n - x^* \rangle \geq 0. \tag{37}$$

From (35), (36), and (37), we have

$$\langle w_n - x^*, \eta_n \rangle \geq \langle w_n - y_n, \eta_n \rangle. \tag{38}$$

Note that $x_{n+1} - w_n = \sigma d_n \eta_n$. From the definition of d_n, one obtains $\langle w_n - y_n, \eta_n \rangle = d_n \|\eta_n\|^2$. Combining (34) and (38), we obtain

$$\begin{aligned}
\|x_{n+1} - x^*\|^2 &\leq \|w_n - x^*\|^2 - 2\sigma d_n \langle w_n - y_n, \eta_n \rangle + \sigma^2 d_n^2 \|\eta_n\|^2 \\
&= \|w_n - x^*\|^2 - 2\sigma d_n^2 \|\eta_n\|^2 + \sigma^2 d_n^2 \|\eta_n\|^2 \\
&= \|w_n - x^*\|^2 - \frac{2-\sigma}{\sigma} \|\sigma d_n \eta_n\|^2 \\
&= \|w_n - x^*\|^2 - \frac{2-\sigma}{\sigma} \|w_n - x_{n+1}\|^2.
\end{aligned}$$

On the other hand, by the definition of x_{n+1} and (19), we have

$$\|x_{n+1} - w_n\|^2 = \sigma^2 d_n^2 \|\eta_n\|^2 \geq \sigma^2 \frac{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2}{\left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)^2} \|w_n - y_n\|^2.$$

Thus we obtain

$$\|w_n - y_n\|^2 \leq \left[\frac{\left(1 + \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)}{\left(1 - \frac{\beta\mu\delta_n\chi_n}{\chi_{n+1}}\right)\sigma} \right]^2 \|w_n - x_{n+1}\|^2.$$

The proof is completed. ■

Now, we are in a position to analyze the convergence of Algorithm 3.2.

Theorem 3.2: *Suppose that Conditions (C1)–(C4) hold. Then the sequence $\{x_n\}$ generated by Algorithm 3.2 converges strongly to $x^* \in \mathrm{VI}(C,A)$, where $\|x^*\| = \min\{\|z\| : z \in \mathrm{VI}(C,A)\}$.*

Proof: The proof is similar to that in Theorem 3.1. Therefore we omit some details of the proof. From Lemma 3.4 and $\sigma \in (0,2)$, we have

$$\|x_{n+1} - x^*\| \leq \|w_n - x^*\|, \quad \forall n \geq 1. \tag{39}$$

Using the same arguments as declared in Theorem 3.1, we have that the sequences $\{x_n\}$, $\{w_n\}$, and $\{y_n\}$ are bounded. By Lemma 3.4 and (25), we deduce

$$\frac{2-\sigma}{\sigma} \|w_n - x_{n+1}\|^2 \leq \|x_n - x^*\|^2 - \|x_{n+1} - x^*\|^2 + \theta_n Q_2. \tag{40}$$

Moreover, we can obtain (27) by using the same arguments as stated in Theorem 3.1. Finally, we show that $\{\|x_n - x^*\|\}$ converges to zero. By Lemma 2.2, we assume that $\{\|x_{n_k} - x^*\|^2\}$ is a subsequence of $\{\|x_n - x^*\|^2\}$ such that (28) holds. From (40) and Condition (C4), we have

$$\begin{aligned}\limsup_{k\to\infty} \frac{2-\sigma}{\sigma}\|w_{n_k} - x_{n_k+1}\|^2 \\ \leq \limsup_{k\to\infty}\left[\|x_{n_k} - x^*\|^2 - \|x_{n_k+1} - x^*\|^2 + \theta_{n_k} Q_2\right] \\ \leq 0,\end{aligned}$$

which means that $\lim_{k\to\infty}\|x_{n_k+1} - w_{n_k}\| = 0$ since $\sigma \in (0,2)$. In view of Lemma 3.4, we observe that $\lim_{k\to\infty}\|y_{n_k} - w_{n_k}\| = 0$. As stated in Theorem 3.1, we can obtain the same result as (29)–(33). Therefore we conclude that $x_n \to x^*$ as $n \to \infty$. This completes the proof. ■

4. Numerical experiments

In this section, we present three numerical examples occurring in finite- and infinite-dimensional Hilbert spaces to illustrate the computational efficiency and robustness of the proposed algorithms over some existing methods in [19,29,34]. All programs are executed on MATLAB 2018a on a personal computer with Intel(R) Core(TM) i5-8250U CPU @1.60GHz 1.80 GHz and RAM 8.00 GB.

Before starting our numerical experiments we introduce the algorithms and convergence theorems in [19,34]. Based on the inertial method, the Armijo-type line search method, the projection contraction algorithm, and the viscosity method, Jolaoso [19] presents an iterative scheme to solve monotone variational inequality problems in real Hilbert spaces. The proposed scheme and the convergence results are shown in Algorithm 4.1 and Theorem 4.1.

Theorem 4.1 ([19]): *Let C be a nonempty, closed, and convex subset of a real Hilbert space $\mathcal{H}$. Let $A : C \to \mathcal{H}$ be a monotone and L-Lipschitz continuous operator, and $f : \mathcal{H} \to \mathcal{H}$ be a contraction mapping with coefficient $\rho \in (0,1)$. Let $\{\theta_n\}$ be a sequence in $(0,1)$ such that $\lim_{n\to\infty}\theta_n = 0$ and $\sum_{n=1}^{\infty}\theta_n = \infty$, and $\{\epsilon_n\}$ be a positive sequence satisfying $\lim_{n\to\infty}\frac{\epsilon_n}{\theta_n} = 0$. Suppose that $\mathrm{VI}(C,A) \neq \emptyset$. Then the sequence $\{x_n\}$ generated by Algorithm 4.1 converges strongly to a point $x^* \in \mathrm{VI}(C,A)$, where $x^* = P_{\mathrm{VI}(C,A)}(f(x^*))$.*

Recently, Thong et al. [34] introduced an inertial subgradient extragradient algorithm to discover solutions of variational inequality problems with pseudomonotone operators in infinite-dimensional Hilbert spaces. Their scheme and weak convergence results are shown in Algorithm 4.2 and Theorem 4.2.

Algorithm 4.1 The Algorithm 3.11 of Jolaoso [19]

Initialization: Take $\alpha > 0$, $\delta > 0$, $\zeta \in (0,1)$, $\mu \in (0,1)$, and $\sigma \in (0,2)$. Let $x_0, x_1 \in \mathcal{H}$.

Iterative Steps: Given the iterates x_{n-1} and x_n ($n \geq 1$). Calculate the iterate x_{n+1} as follows:

Step 1. Compute $w_n = x_n + \tau_n(x_n - x_{n-1})$, where

$$\tau_n = \begin{cases} \min\left\{\dfrac{n-1}{n+\alpha-1}, \dfrac{\epsilon_n}{\|x_n - x_{n-1}\|}\right\}, & \text{if} x_n \neq x_{n-1}; \\ \dfrac{n-1}{n+\alpha-1}, & \text{otherwise.} \end{cases}$$

Step 2. Compute $y_n = P_C(w_n - \chi_n A w_n)$, where the step size χ_n is chosen to be the largest $\chi \in \{\delta, \delta\zeta, \delta\zeta^2, \ldots\}$ satisfying

$$\chi \|Aw_n - Ay_n\| \leq \mu \|w_n - y_n\|.$$

If $w_n = y_n$, then stop and y_n is a solution of (VIP). Otherwise, go to **Step 3**.

Step 3. Compute $z_n = w_n - \sigma d_n \eta_n$, where

$$d_n = \frac{\langle w_n - y_n, \eta_n \rangle}{\|\eta_n\|^2}, \quad \eta_n = w_n - y_n - \chi_n(Aw_n - Ay_n).$$

Step 4. Compute $x_{n+1} = \theta_n f(x_n) + (1-\theta_n) z_n$. Set $n := n+1$ and go to **Step 1**.

Algorithm 4.2 The Algorithm 3.1 of Thong et al. [19]

Initialization: Take $\mu \in (0,1)$, $\tau \in [0, 1 - \frac{3-\sqrt{5-4\mu}}{1+\mu})$, and $\chi_1 > 0$. Let $x_0, x_1 \in \mathcal{H}$.

Iterative Steps: Given the iterates x_{n-1} and x_n ($n \geq 1$). Calculate the iterate x_{n+1} as follows:

Step 1. Compute $w_n = x_n + \tau(x_n - x_{n-1})$.

Step 2. Compute $y_n = P_C(w_n - \chi_n A w_n)$. If $w_n = y_n$, then stop and y_n is a solution of (VIP). Otherwise, go to **Step 3**.

Step 3. Compute $x_{n+1} = P_{T_n}(w_n - \chi_n A y_n)$, where $T_n := \{x \in \mathcal{H} : \langle w_n - \chi_n A w_n - y_n, x - y_n \rangle \leq 0\}$. Set $\Delta_n = \langle Aw_n - Ay_n, x_{n+1} - y_n \rangle$ and update

$$\chi_{n+1} = \begin{cases} \min\left\{\mu \dfrac{\|w_n - y_n\|^2 + \|x_{n+1} - y_n\|^2}{2\Delta_n}, \chi_n + \rho_n\right\}, & \text{if} \Delta_n > 0; \\ \chi_n + \rho_n, & \text{otherwise.} \end{cases}$$

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

Theorem 4.2 ([34]): *Let C be a nonempty, closed, and convex subset of a real Hilbert space $\mathcal{H}$. Let operator $A : \mathcal{H} \to \mathcal{H}$ be pseudomonotone and L-Lipschitz continuous. Let A satisfies the following condition*

$$\text{whenever } \{x_n\} \subset C,\ x_n \rightharpoonup z,\ \text{one has } \|Az\| \leq \liminf_{n\to\infty} \|Ax_n\|.$$

Let $\{\rho_n\}$ be a nonnegative real numbers sequence such that $\sum_{n=1}^{\infty} \rho_n < +\infty$. Assume that $\mathrm{VI}(C, A) \neq \emptyset$. Then $\{x_n\}$ formed by Algorithm 4.2 converges weakly to $x^* \in \mathrm{VI}(C, A)$.

4.1. Performance profiles

To measure the computational efficiency and stability of the proposed algorithms, we employ the performance profiles introduced by Dolan and Moré [35], which are widely used in the numerical optimization community. We would like to thank the reviewer for pointing out this tool to attract our attention. Let $G = \{s \mid s = 1, 2, 3, \dots, n_s\}$ denote the set of algorithms and $B = \{p \mid p = 1, 2, 3, \dots, n_p\}$ the set of problems. Suppose there is a set of benchmark tests for n_s algorithms solving n_p problems. The performance profile is used to evaluate and compare the performance of the algorithm set G on the problem set B, where the performance metrics can be some information of interest such as number of iterations, computation time, function value evaluation, error evaluation, etc. It is assumed that computation time is used as the metric to be measured. We define

$$t_{p,s} = \text{computation time required by algorithm } s \text{ to solve problem } p.$$

If algorithm s fails to solve problem p, we define $t_{p,s} = +\infty$. A similar definition is used if the measure is number of iterations. The performance ratio defined by Dolan and Moré [35] is expressed as follows

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,s} : s \in G\right\}}.$$

This represents the performance of algorithm s on problem p compared to the best performance of any algorithm on this problem. Notice that $r_{p,s} \geq 1$ for any algorithm s in sloving problem p and that $r_{p,s} = 1$ indicates (one of) the best performance ratio. To obtain an evaluation of the overall performance of algorithm s on problem set B, define the scaled performance profile as follows

$$P\left(\log_2\left(r_{p,s}\right) \leq \omega : 1 \leq s \leq n_s\right) = \rho_s(\omega) = \frac{1}{n_p}\text{size}\left\{p \in B : \log_2(r_{p,s}) \leq \omega\right\}.$$

Then $\rho_s(\omega)$ is the ratio of the performance of algorithm s with respect to the scaled performance ratio $\log_2(r_{p,s})$ within the best possible ratio factor $\omega \in \mathbb{R}$. The function ρ_s is the (cumulative) distribution function for the performance ratio of algorithm s on problem set B. Clearly, $\rho_s(0)$ denotes the rate by which algorithm s outperforms other algorithms on problem set B. Therefore, we can choose the maximum of $\rho_s(0)$ among all algorithms if we are only interested in the number of wins of the algorithm on problem set B. Moreover, $\rho_s(\omega_{\max})$ denotes the number of problems solved by algorithm s, where $\omega_{\max} = \max_{s\in G, p\in B} \log_2(r_{p,s})$. That is, $\rho_s(\omega_{\max})$ can be used to measure the stability of algorithm s on problem set B. Hence, if $\rho_s(\omega)$ is large with respect to ω being small, then the algorithm $s \in G$ is 'fast'; if $\rho_s(\omega)$ is large with respect to ω being large, then s is 'robust'.

4.2. Theoretical examples

In this subsection, we consider three numerical tests known in the literature, where the feasible sets in Examples 4.1 and 4.2 are bounded and the feasible set in Example 4.3 is unbounded.

Example 4.1: The first example is the HP-Hard problem which is considered by much of the literature; see, e.g.[12,19,29,42]. Let the linear operator $A : \mathbb{R}^m \to \mathbb{R}^m$ be given by

$$A(x) = Gx,$$

where $G = BB^\mathsf{T} + S + E$, matrix $B \in \mathbb{R}^{m\times m}$, matrix $S \in \mathbb{R}^{m\times m}$ is skew-symmetric, and matrix $E \in \mathbb{R}^{m\times m}$ is diagonal matrix whose diagonal terms are non-negative (hence G is positive symmetric definite). Let the feasible set C be a box constraint with the form

$$C = \{x \in \mathbb{R}^m : -2 \le x_i \le 5,\ i = 1, 2, 3, \ldots, m\}.$$

It is easy to see that A is monotone, Lipschitz continuous and its Lipschitz constant $L = \|G\|$. In this example, all entries of B, S are generated randomly in $[-2, 2]$, and E is generated randomly in $[0, 2]$. The solution set of the (VIP) is $x^* = \{\mathbf{0}\}$. We use $D_n = \|x_n - x^*\|$ to measure the nth iteration error of all algorithms. The maximum number of iterations of 200 as a common stopping criterion. Next, we test the performance of the proposed algorithms under different parameters. Specifically, we consider the following three cases.

Case 1: *Compare inertial parameters* τ_n. Take $\sigma = 1.5$, $\theta_n = 1/(n+1)$, $\tau = \{0.2, 0.4, 0.6, 0.8\}$, $\epsilon_n = 100/(n+1)^2$, $\beta = 0.8$, $\chi_1 = 0.6$, $\mu = 0.6$, $\delta_n = 1 + 1/n$, $\xi_n = 1 + 1/(n+1)^{1.1}$ and $\rho_n = 1/(n+1)^{1.1}$ for the proposed Algorithms 3.1 and 3.2. The numerical performance of the proposed algorithms with different parameters τ is given in Figure 1.

Case 2: *Compare the new parameter* β. Choose $\tau = 0.6$, $\beta = \{0.8, 0.9, 1.0, 1.1\}$ and let the values of parameters σ, θ_n, ϵ_n, χ_1, μ, δ_n, ξ_n, and ρ_n be the same as in *Case 1*. The numerical behavior of the proposed algorithms with different parameters β is shown in Figure 2.

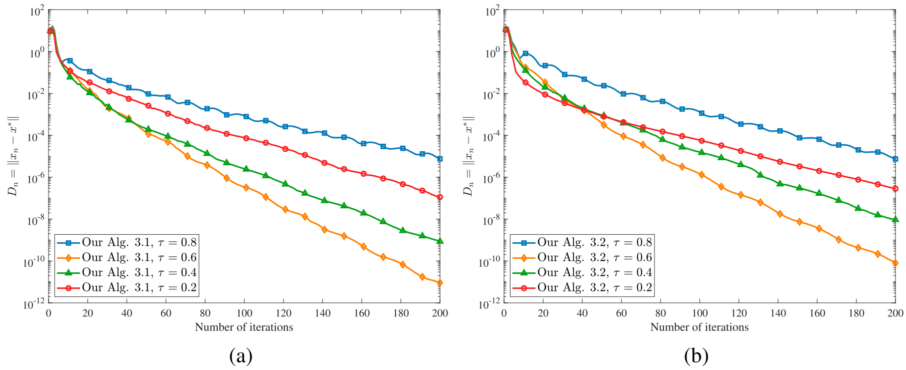

Figure 1. The behavior of our algorithms with different τ in Example 4.1 ($m = 20$). (a) Our Algorithm 3.1 and (b) Our Algorithm 3.2.

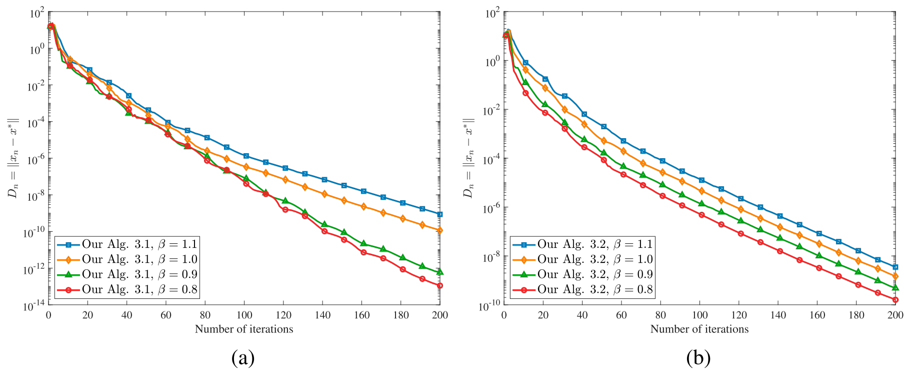

Figure 2. The behavior of our algorithms with different β in Example 4.1 ($m = 20$). (a) Our Algorithm 3.1 and (b) Our Algorithm 3.2.

Case 3*: Compare step size χ_n.* Take $\tau = 0.6$ and let the values of parameters σ, θ_n, ϵ_n, β, χ_1, and μ be the same as in *Case 1*. The numerical behavior of the proposed algorithms with different step size χ_n is expressed in Figure 3.

To end this example, we compare the proposed algorithms with the Algorithms 3.1, 3.3, 3.4, and 3.6 presented by Tan, Li, and Cho [29] (shortly, TLC Algorithm 3.1, TLC Algorithm 3.3, TLC Algorithm 3.4, and TLC Algorithm 3.6), and the Algorithm 3.11 introduced by Jolaoso [19], and the Algorithm 3.1 suggested by Thong et al. [34]. The parameters of all algorithms are set as follows.

- Choose $\tau = 0.6$, $\epsilon_n = 100/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 1/(n+1)$, $\beta = 0.8$, $\chi_1 = 0.6$, $\mu = 0.6$, $\delta_n = 1 + 1/n$, $\xi_n = 1 + 1/(n+1)^{1.1}$, and $\rho_n = 1/(n+1)^{1.1}$ for the proposed Algorithms 3.1 and 3.2.
- Take $\tau = 0.6$, $\epsilon_n = 100/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 1/(n+1)$, $\alpha_n = 0.8(1 - \theta_n)$, $f(x) = 0.1x$, $\delta = 2$, $\zeta = 0.5$, and $\mu = 0.6$ for TLC Algorithm 3.1, TLC Algorithm 3.3, TLC Algorithm 3.4, and TLC Algorithm 3.6 [29].

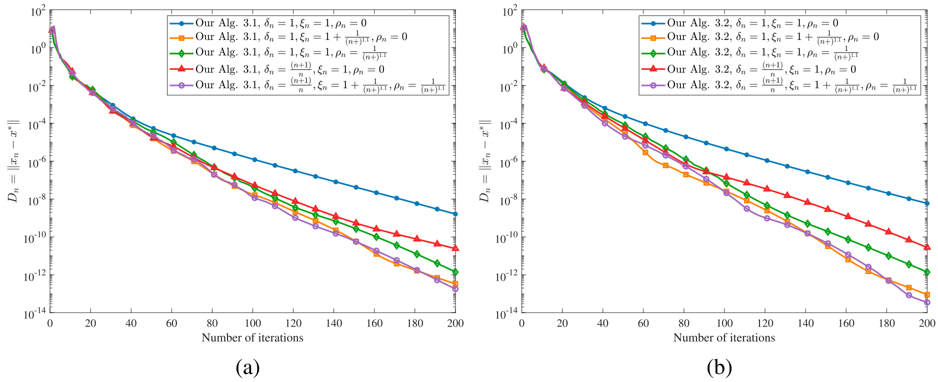

Figure 3. The behavior of our algorithms with different χ_n in Example 4.1 ($m = 20$). (a) Our Algorithm 3.1 and (b) Our Algorithm 3.2.

Table 1. Numerical results of all algorithms with different dimensions for Example 4.1.

	$m = 20$		$m = 50$		$m = 100$		$m = 200$	
Algorithms	D_n	CPU (s)	D_n	CPU (s)	D_n	CPU (s)	D_n	CPU (s)
Our Algorithm 3.1	3.21E−52	0.0775	1.85E−25	0.0791	1.56E−16	0.1058	9.27E−13	0.1644
Our Algorithm 3.2	1.98E−45	0.0725	2.01E−21	0.0769	1.46E−14	0.1092	3.13E−11	0.1335
TLC Algorithm 3.1	2.14E−22	0.1058	6.46E−15	0.1236	1.13E−11	0.3062	5.31E−10	0.4587
TLC Algorithm 3.3	1.96E−26	0.0910	4.80E−16	0.1131	6.48E−12	0.2305	1.22E−09	0.4502
TLC Algorithm 3.4	2.14E−22	0.0897	6.46E−15	0.1253	1.13E−11	0.2421	5.31E−10	0.4154
TLC Algorithm 3.6	1.96E−26	0.1141	4.80E−16	0.1157	6.48E−12	0.2475	1.22E−09	0.4147
Jolaoso Algorithm 3.11	1.97E−23	0.0981	2.57E−12	0.1211	1.83E−12	0.2361	3.49E−11	0.4168
Thong et al. Algorithm 3.1	8.95E−10	0.0743	1.71E−03	0.0794	2.85E−02	0.1065	7.63E−01	0.1246

- Set $\alpha = 100$, $\epsilon_n = 100/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 1/(n+1)$, $f(x) = 0.1x$, $\delta = 2$, $\zeta = 0.5$, and $\mu = 0.6$ for Jolaoso's Algorithm 3.11 [19].
- Select $\chi_1 = 0.6$, $\tau = 0.1(\sqrt{5} - 2)$, $\mu = 0.2(\frac{1-4\tau-\tau^2}{(1-\tau)^2})$, and $\rho_n = 1/(n+1)^{1.1}$ for the Algorithm 3.1 presented by Thong et al. [34].

The initial values $x_0 = x_1$ are chosen randomly in $\mathbb{R}^m$ and the maximum number of iterations 1000 is used as a stopping criterion common to all algorithms. The numerical results of all algorithms with four dimensions are shown in Table 1. To demonstrate the overall computational efficiency of the proposed algorithms and the comparison algorithms in 30 different dimensions ($m = \{5, 15, 25, 35, \ldots, 295\}$) of Example 4.1, we measured their performance profiles using computation time and termination error as metrics, and the results are shown in Figures 4 and 5, respectively.

Next, we consider a variational inequality problem in infinite-dimensional real Hilbert spaces, where the operator involved is pseudomonotone rather than monotone. In this case, Algorithm 3.11 introduced by Jolaoso [19] for solving monotone variational inequality problems will not be available. Therefore, Jolaoso's algorithm will not be implemented in this example.

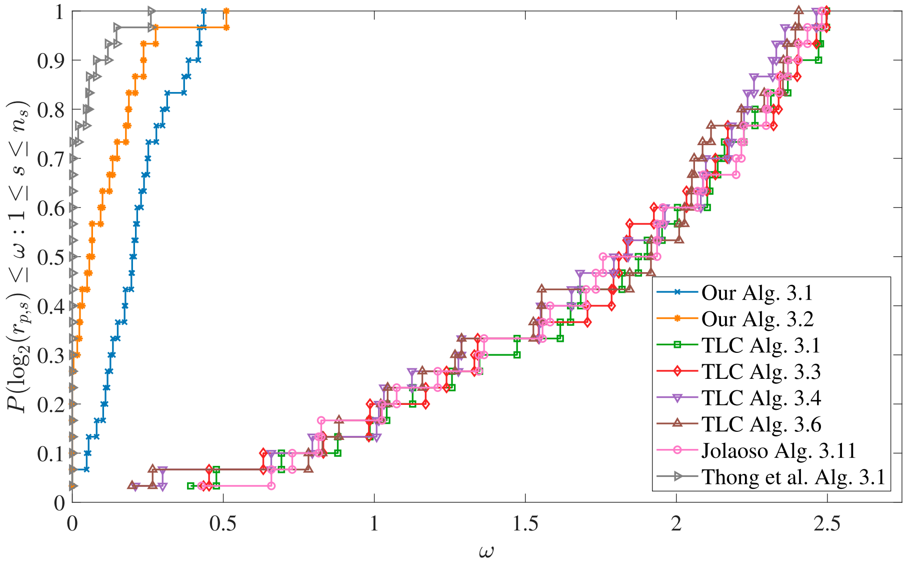

Figure 4. Performance profiles of all algorithms based on computation time for Example 4.1.

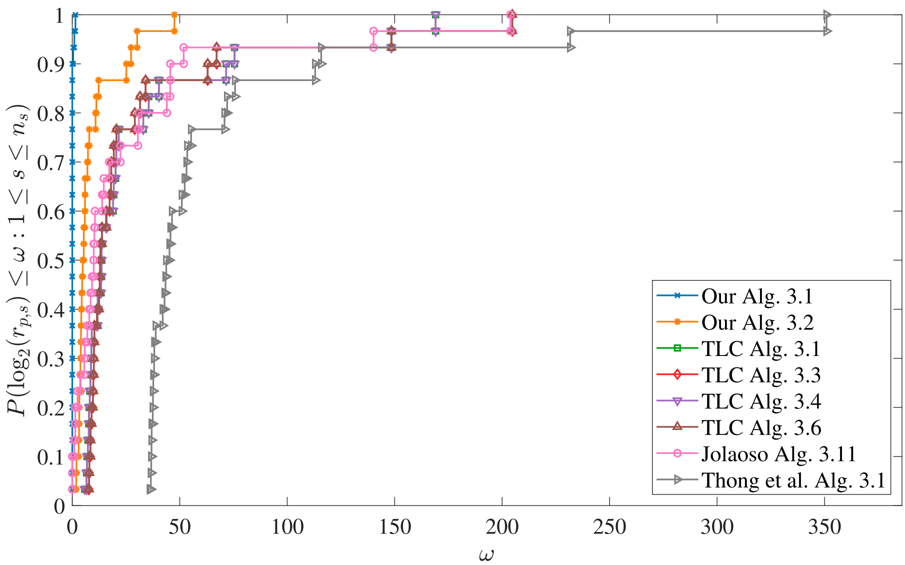

Figure 5. Performance profiles of all algorithms based on termination error for Example 4.1.

Example 4.2: This example is considered in [43] where the variational inequality operator involved is pseudomonotone rather than monotone. Let $\mathcal{H} = L^2([0,1])$ be an infinite-dimensional Hilbert space with inner product

$$\langle x, y\rangle = \int_0^1 x(t)y(t)\,\mathrm{d}t, \quad \forall x, y \in \mathcal{H}$$

Table 2. Numerical results of all algorithms with different initial values for Example 4.2.

	$x_1(t) = 10t^2$		$x_1(t) = 2e^t$		$x_1(t) = 3\cos(t)$		$x_1(t) = \log(2t)$	
Algorithms	D_n	CPU (s)	D_n	CPU (s)	D_n	CPU (s)	D_n	CPU (s)
Our Algorithm 3.1	8.57E−32	22.8165	4.16E−30	22.4390	8.85E−30	22.2947	8.42E−31	22.3925
Our Algorithm 3.2	8.87E−30	22.1634	1.66E−30	20.9524	4.70E−30	20.9188	8.42E−31	21.1196
TLC Algorithm 3.1	2.18E−26	54.7587	2.49E−26	47.7078	2.67E−26	47.5906	9.81E−27	47.5743
TLC Algorithm 3.3	2.21E−27	53.1460	1.91E−27	47.3066	1.40E−27	47.3911	2.47E−27	47.1040
TLC Algorithm 3.4	1.35E−26	47.7945	8.63E−26	46.4754	2.97E−26	46.3224	9.81E−27	46.4572
TLC Algorithm 3.6	3.13E−26	46.7180	4.61E−27	46.3843	1.24E−26	46.4638	2.47E−27	45.8154
Thong et al. Algorithm 3.1	3.02E−07	17.0902	6.11E−07	17.4131	1.04E−06	17.4149	2.23E−06	17.5943

and induced norm

$$\|x\| = \left(\int_0^1 |x(t)|^2 \,\mathrm{d}t\right)^{1/2}, \quad \forall x \in \mathcal{H}.$$

Assume that r and R are two positive real numbers such that $R/(k+1) < r/k < r < R$ for some $k > 1$. Let the feasible set be defined by

$$C = \{x \in \mathcal{H} : \|x\| \leq r\}$$

and the operator $A : \mathcal{H} \to \mathcal{H}$ be given by

$$Ax = (R - \|x\|)x, \quad \forall x \in \mathcal{H}.$$

It is not hard to check that operator A is pseudomonotone rather than monotone (see [43, Section 4]). For the experiment, we choose $R = 1.5$, $r = 1$, and $k = 1.1$. The solution of the variational inequality problem (VIP) with A and C given above is $x^*(t) = 0$. The parameters of all algorithms are set as follows.

- Adopt $\tau = 0.2, \epsilon_n = 1/(n+1)^2, \sigma = 1.5, \theta_n = 1/(n+1), \beta = 1.0, \chi_1 = 0.1$, $\mu = 0.4$, $\delta_n = 1 + 1/n$, $\xi_n = 1 + 1/(n+1)^{1.1}$, and $\rho_n = 1/(n+1)^{1.1}$ for the proposed Algorithms 3.1 and 3.2.
- Set $\tau = 0.2$, $\epsilon_n = 1/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 1/(n+1)$, $\alpha_n = 0.9(1-\theta_n)$, $f(x) = 0.1x$, $\delta = 2$, $\zeta = 0.5$, and $\mu = 0.4$ for TLC Algorithm 3.1, TLC Algorithm 3.3, TLC Algorithm 3.4, and TLC Algorithm 3.6 [29].
- Select $\chi_1 = 0.1$, $\tau = 0.1(\sqrt{5} - 2)$, $\mu = 0.2(\frac{1-4\tau-\tau^2}{(1-\tau)^2})$, and $\rho_n = 1/(n+1)^{1.1}$ for the Algorithm 3.1 presented by Thong et al. [34].

The function $D_n = \|x_n(t) - x^*(t)\|$ is used to measure the error of the nth iteration step and the maximum number of iterations 50 is used as a common stopping criterion for all algorithms. The numerical results of all algorithms with four different initial values $x_0(t) = x_1(t)$ are stated in Table 2.

Notice that the constraint sets in Examples 4.1 and 4.2 are bounded, we next provide an unconstrained variational inequality problem with an unbounded feasible set in a finite-dimensional space.

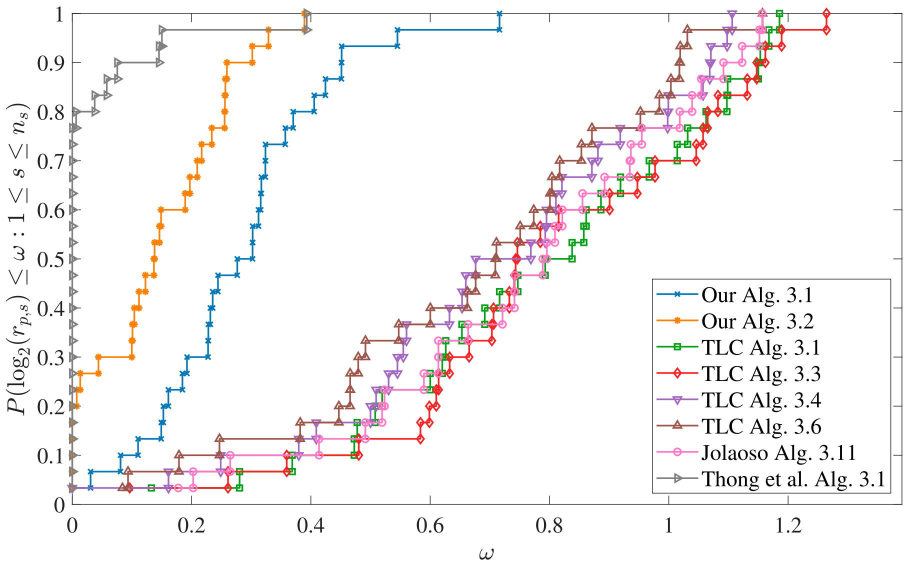

Figure 6. Performance profiles of all algorithms based on computation time for Example 4.3.

Example 4.3: This a classical numerical example in finite-dimensional Euclidean spaces where the usual gradient method does not converge, and it is used by many authors to verify the convergence performance of their algorithms; see, e.g.[12,18,19]. Let the feasible set be given by $C = \mathbb{R}^m$ for some positive even integer m. Let variational inequality operator $A : \mathbb{R}^m \to \mathbb{R}^m$ be the square $m \times m$ matrix defined by $A = (a_{ij})_{1\leq i,j\leq m}$, where the elements are generated by the following form

$$a_{ij} = \begin{cases} -1, & \text{if } j = m+1-i \text{ and } j > i; \\ 1, & \text{if } j = m+1-i \text{ and } j < i; \\ 0, & \text{otherwise.} \end{cases}$$

The solution to the (VIP) with A and C given above is $x^* = (0, 0, \ldots, 0)^\top \in \mathbb{R}^m$. In this example keep the parameters of our algorithms and the comparison methods as in Example 4.1, but adjust $\tau = 0.02$ and $\beta = 1.1$ for our Algorithms 3.1 and 3.2, and adjust $\tau = 0.02$ for the methods introduced by Tan, Li, and Cho [29], and adjust $\alpha = 10000$ for the Algorithm 3.1 proposed by Jolaoso [19]. The initial values $x_0 = x_1$ are chosen randomly in $\mathbb{R}^m$. We use $D_n = \|x_n - x^*\|$ to capture the error of the algoritms at the n-th iteration. The maximum number of iterations 500 is taken as a general stopping criterion for all algorithms. Table 3 shows their computational results in four different dimensions. Then, we consider $m = \{10, 30, 50, 70, \ldots, 590\}$ for a total of 30 different dimensions, and the performance profiles of all algorithms regarding execution time and termination error at different initial values are shown in Figures 6 and 7, respectively.

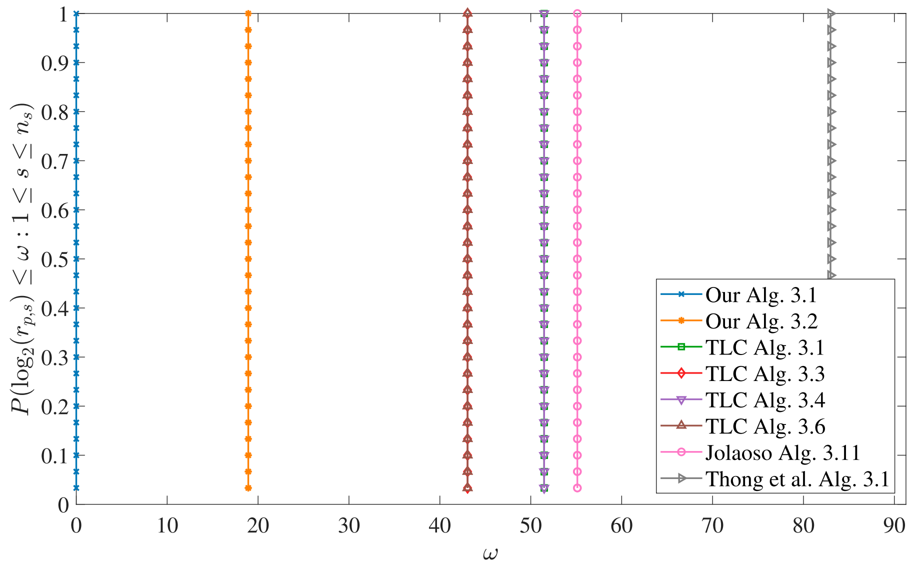

Figure 7. Performance profiles of all algorithms based on termination error for Example 4.3.

Table 3. Numerical results of all algorithms with different dimensions for Example 4.3.

	$m = 200$		$m = 500$		$m = 1000$		$m = 2000$	
Algorithms	D_n	CPU (s)	D_n	CPU (s)	D_n	CPU (s)	D_n	CPU (s)
Our Algorithm 3.1	1.98E−35	0.0816	2.92E−35	0.1982	4.22E−35	2.6434	5.99E−35	10.5476
Our Algorithm 3.2	9.82E−30	0.0587	1.45E−29	0.1603	2.10E−29	2.2536	2.98E−29	8.8933
TLC Algorithm 3.1	6.17E−20	0.0990	9.12E−20	0.2340	1.32E−19	3.6865	1.87E−19	19.7716
TLC Algorithm 3.3	1.80E−22	0.0905	2.65E−22	0.2183	3.83E−22	5.0999	5.44E−22	19.7334
TLC Algorithm 3.4	6.17E−20	0.0912	9.12E−20	0.2246	1.32E−19	4.6607	1.87E−19	18.3037
TLC Algorithm 3.6	1.80E−22	0.0891	2.65E−22	0.2568	3.83E−22	4.6407	5.44E−22	18.2501
Jolaoso Algorithm 3.11	7.76E−19	0.0805	1.15E−18	0.2448	1.66E−18	4.7288	2.35E−18	18.2412
Thong et al. Algorithm 3.1	1.88E−10	0.0595	2.78E−10	0.1249	4.01E−10	1.8215	5.70E−10	7.1698

Remark 4.1: We make the following comments on Examples 4.1–4.3.

- The information in Figures 1–3 demonstrates that the appropriate parameters τ, β, and χ_n play an active role in the convergence speed and accuracy of the proposed algorithms, respectively.
- Our two algorithms work well in different dimensions and with different initial values; see Tables 1–3. Figures 4 and 6 show the performance profiles of the algorithms based on computation time in different dimensions, and they demonstrate that Thong et al.'s Algorithm 3.1 [34] and our two algorithms proposed in this paper outperform the other compared algorithms in terms of computation time, i.e. Thong et al.'s Algorithm 3.1 and our two algorithms require less computation time than Tan et al.'s algorithms [29] and Jolaoso's Algorithm 3.11 [19] when performing the same number of iterations. This phenomenon is easily explained by the fact that Armijo-type algorithms take

more time to find the step size in each iteration. On the other hand, Figures 5 and 7 show that our two algorithms outperform the compared algorithms in terms of termination error, which means that our algorithms can obtain higher accuracy than the compared algorithms when performing the same number of iterations. Therefore, the proposed methods perform better in terms of accuracy and convergence speed than the algorithms embedded with Armijo-type step size presented in [19,29], and the proposed algorithms can obtain higher accuracy than Algorithm 3.1 in [34]. These observations are not significantly related to the choice of initial values and the size of dimensions. Thus, our methods are efficient and robust.

- Tables 1–3 and Figures 4–7 illustrate that our Algorithm 3.2 outperforms our Algorithm 3.1 in terms of computation time, while our Algorithm 3.1 performs better than our Algorithm 3.2 in terms of termination error.
- Notice that the accuracy of TLC Algorithm 3.1 and TLC Algorithm 3.4 in Table 3 is the same, as is TLC Algorithm 3.3 and TLC Algorithm 3.6. Indeed, when $C = \mathbb{R}^m$, one has $y_n = P_C(w_n - \chi_n A w_n) = w_n - \chi_n A w_n$, in which case $z_n = P_{T_n}(w_n - \sigma \chi_n d_n A y_n) = w_n - \sigma \chi_n d_n A y_n$ by the definition of T_n, and $\eta_n = \chi_n A y_n$. Thus, one obtains $z_n = w_n - \sigma d_n \eta_n$, at which point TLC Algorithm 3.1 becomes TLC Algorithm 3.4 and TLC Algorithm 3.3 turns into TLC Algorithm 3.6.

5. Two real-world applications

In this section we apply the proposed algorithms to optimal control problems and signal processing problems, and demonstrate their computational efficiency and stability with the help of the performance profiles introduced in [35].

5.1. Applications to optimal control problems

In this subsection, we apply the proposed algorithms for solving the variational inequality problem (VIP) that appears in optimal control problems and compare them with the algorithms in [19,29,34]. Let $L_2([0,T],\mathbb{R}^m)$ be the square-integrable Hilbert space with inner product $\langle p,q\rangle = \int_0^T \langle p(t), q(t)\rangle \,\mathrm{d}t$ and norm $\|p\| = \sqrt{\langle p,p\rangle}$. Let $g(p)$ be the terminal objective function and Φ be a convex and differentiable function defined on the attainability set. Let $p(t)$ represent the control function, V stand for a set of feasible controls consisting of m piecewise continuous functions, and $x(t)$ refer to the trajectory. Recall that the optimal control problem is stated as follows:

$$\begin{cases} \text{find } p^*(t) \in \operatorname{Argmin}\{g(p) \mid p \in V\}, \\ g(p) = \Phi(x(T)), \\ V = \left\{p(t) \in L_2\left([0,T],\mathbb{R}^m\right) : p_i(t) \in \left[p_i^-, p_i^+\right], \, i = 1,2,\ldots,m\right\}, \\ \text{such that } \dot{x}(t) = Q(t)x(t) + W(t)p(t), \quad 0 \le t \le T, \quad x(0) = x_0, \\ \text{where } Q(t) \in \mathbb{R}^{n\times n} \text{ and } W(t) \in \mathbb{R}^{n\times m} \text{ for } t \in [0,T]. \end{cases} \tag{42}$$

By the solution of problem (42), we mean a optimal control $p^*(t)$ and a corresponding trajectory $x^*(t)$ such that its terminal value $x^*(T)$ minimizes objective function $g(p)$. It is known that the optimal control problem (42) can be transformed into a variational inequality problem; e.g. see [1] for more details. As with the method described in [1], we begin with the decomposition of the optimal control problem (42) by using the classical Euler discretization method and then solve the variational inequality problem corresponding to the discretized version of the problem by applying the proposed algorithms as well as the compared methods.

Select the parameter N associated with the mesh size to be 100. The initial controls $p_0(t) = p_1(t)$ are randomly generated in $[-0.5, 0.5]$ and the common stopping criterion between our algorithms and the comparison ones is $D_n = \|w_n - y_n\| \le 10^{-4}$. In the next two numerical examples all the parameters of the algorithms are chosen as follows.

- Choose $\tau = (n-1)/(n-1+3)$, $\epsilon_n = 10^{-4}/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 10^{-4}/(n+1)$, $\beta = 0.8$, $\chi_1 = 1.5$, $\mu = 0.4$, $\delta_n = 1 + 1/(n+1)^{1.1}$, $\xi_n = 1 + 10^{-1}/(n+1)^{1.1}$, and $\rho_n = 10^{-1}/(n+1)^{1.1}$ for the proposed Algorithms 3.1 and 3.2.
- Take $\tau = 0.01$, $\epsilon_n = 10^{-4}/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 10^{-4}/(n+1)$, $\alpha_n = 0.8(1-\theta_n)$, $f(x) = 0.1x$, $\delta = 1$, $\zeta = 0.5$, and $\mu = 0.4$ for TLC Algorithm 3.1, TLC Algorithm 3.3, TLC Algorithm 3.4, and TLC Algorithm 3.6 [29].
- Set $\alpha = 3$, $\epsilon_n = 10^{-4}/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 10^{-4}/(n+1)$, $f(x) = 0.1x$, $\delta = 1$, $\zeta = 0.5$, and $\mu = 0.4$ for Jolaoso's Algorithm 3.11 [19].
- Select $\chi_1 = 1.5$, $\tau = 0.1(\sqrt{5}-2)$, $\mu = 0.2(\frac{1-4\tau-\tau^2}{(1-\tau)^2})$, and $\rho_n = 10^{-1}/(n+1)^{1.1}$ for the Algorithm 3.1 presented by Thong et al. [34].

Example 5.1 (Control of a harmonic oscillator, see [44]):

$$
\begin{aligned}
\text{minimize} \quad & x_2(3\pi) \\
\text{subject to} \quad & \dot{x}_1(t) = x_2(t), \\
& \dot{x}_2(t) = -x_1(t) + p(t), \quad \forall t \in [0, 3\pi], \\
& x(0) = 0, \\
& p(t) \in [-1, 1].
\end{aligned}
$$

The exact optimal control of Example 5.1 is known:

$$
p^*(t) = \begin{cases} 1, & \text{if } t \in [0, \pi/2) \cup (3\pi/2, 5\pi/2); \\ -1, & \text{if } t \in (\pi/2, 3\pi/2) \cup (5\pi/2, 3\pi. \end{cases}
$$

As shown in [41, Example 5.1], Example 5.1 can be converted to a monotone and Lipschitz type variational inequality problem and therefore the algorithms presented in this paper can be used. Figure 8 presents the numerical results of the proposed Algorithm 3.1 in Example 5.1.

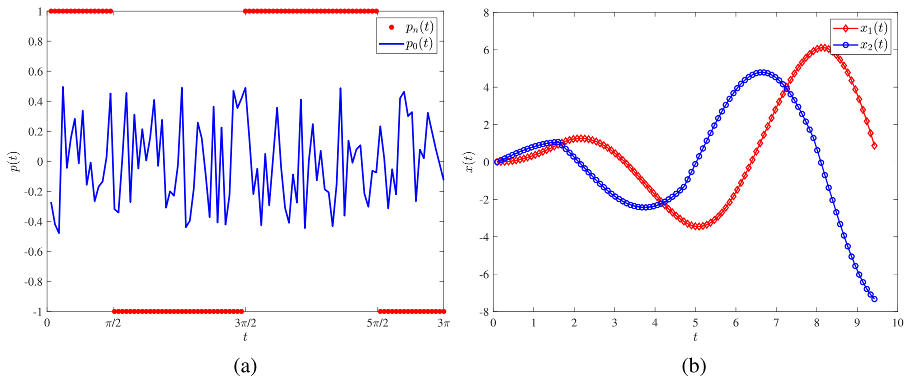

Figure 8. The numerical behavior of our Algorithm 3.1 for Example 5.1. (a) Initial and optimal controls and (b) Optimal trajectories.

Next we consider the case where the terminal function is nonlinear.

Example 5.2 (See [45]):

$$\begin{aligned} \text{minimize} \quad & -x_1(2) + (x_2(2))^2 \\ \text{subject to} \quad & \dot{x}_1(t) = x_2(t), \\ & \dot{x}_2(t) = p(t), \quad \forall t \in [0, 2], \\ & x_1(0) = 0, \;\; x_2(0) = 0, \\ & p(t) \in [-1, 1]. \end{aligned}$$

The exact optimal control of Example 5.2 is

$$p^*(t) = \begin{cases} 1, & \text{if } t \in [0, 1.2); \\ -1, & \text{if } t \in (1.2, 2]. \end{cases}$$

Example 5.2 can similarly be converted to a monotone variational inequality problem. The numerical results of the proposed Algorithm 3.2 in Example 5.2 are shown in Figure 9.

In order to demonstrate the computational efficiency and stability of the proposed algorithms in solving optimal control problems, we randomly selected 100 different initial values to test for Examples 5.1 and 5.2. The performance profiles of all algorithms based on computation time and number of iterations for Examples 5.1 and 5.2 are shown in Figures 10 and 11, and Figures 12 and 13, respectively. Furthermore, we present in Table 4 the average computation time and the average number of iterations for all algorithms in solving Examples 5.1 and 5.2 for these 100 tests.

Remark 5.1: We have the following observations for Examples 5.1 and 5.2 in optimal control problems.

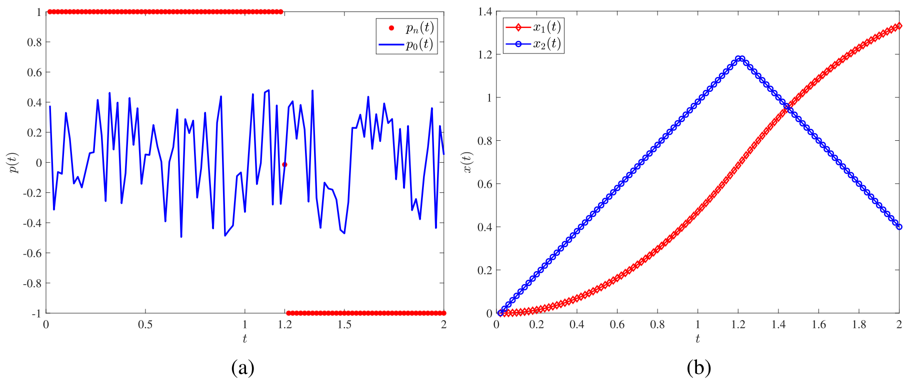

Figure 9. The numerical behavior of our Algorithm 3.2 for Example 5.2. (a) Initial and optimal controls and (b) Optimal trajectories.

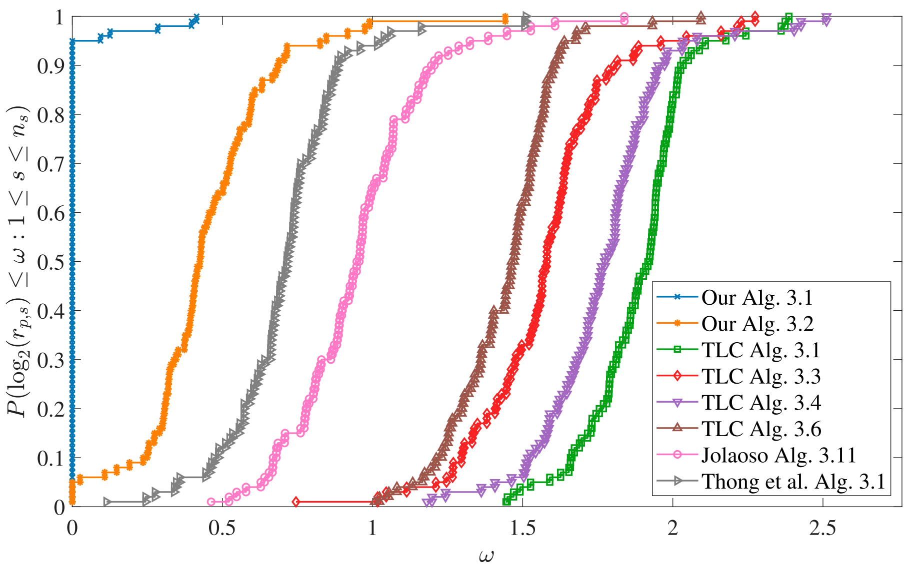

Figure 10. Performance profiles of all algorithms based on computation time for Example 5.1.

- The proposed algorithms provide a good solution to optimal control problems when the terminal function is linear or nonlinear, as shown in Figures 8 and 9.
- Figure 10 shows that our Algorithm 3.1 can win 95% compared to other algorithms in terms of computation time when solving Example 5.1. In other words, the proposed Algorithm 3.1 would be preferred in solving Example 5.1 if we were interested in computational time. Furthermore, the information in Figure 11 indicates that our Algorithms 3.1 and 3.2 are also preferable to other algorithms in terms of the number of iterations.
- In solving Example 5.2 where the terminal function is nonlinear, the information in Figure 12 shows that our Algorithm 3.1 can solve 80% of the problems

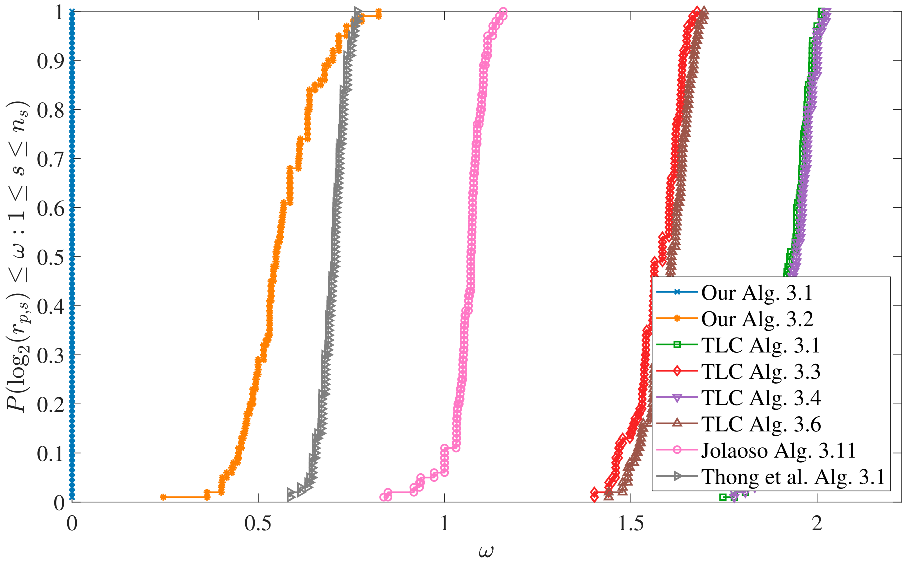

Figure 11. Performance profiles of all algorithms based on number of iterations for Example 5.1.

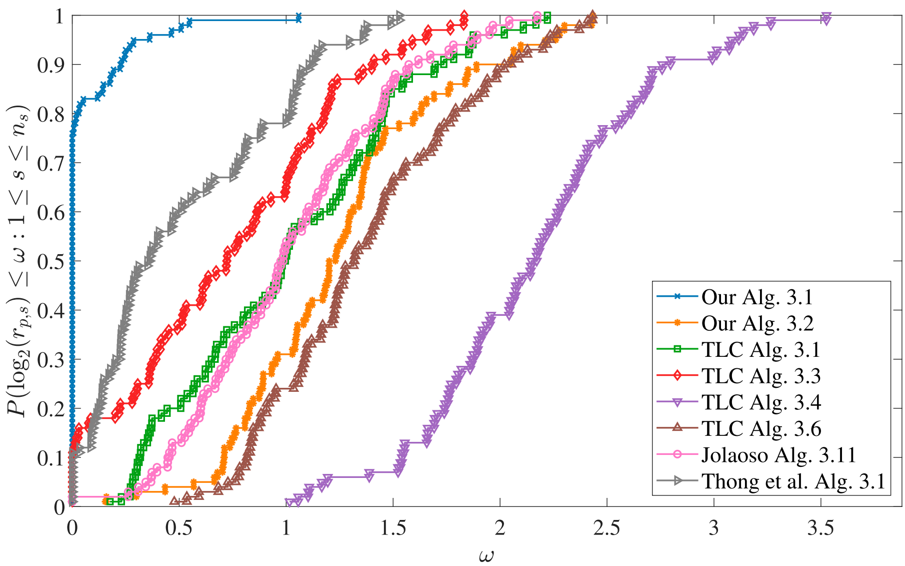

Figure 12. Performance profiles of all algorithms based on computation time for Example 5.2.

with maximum efficiency in terms of computation time, while Figure 13 indicates that our Algorithm 3.1 can win 60% in terms of the number of iterations.

- It is worth noting that the proposed Algorithm 3.2 does not perform as well in Example 5.2 as it does in Example 5.1; see Figures. 10–13.
- In solving Example 5.1 where the terminal function is linear, algorithms using the simple step size update criterion (our Algorithms 3.1 and 3.2 and Thong

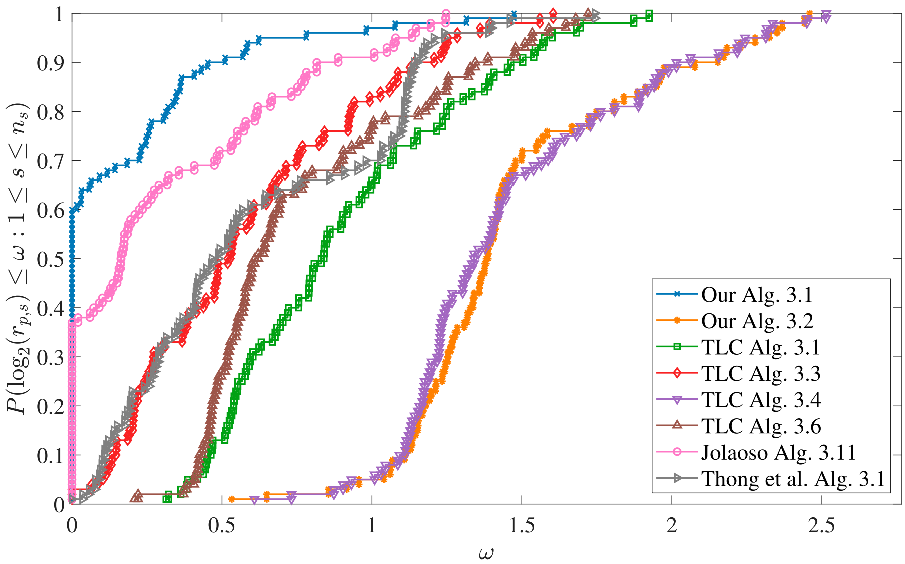

Figure 13. Performance profiles of all algorithms based on number of iterations for Example 5.2.

Table 4. Numerical results for all algorithms in Examples 5.1 and 5.2.

	Example 5.1		Example 5.2	
Algorithms	Avg. CPU (*s*)	Avg. num. of iter.	Avg. CPU (*s*)	Avg. num. of iter.
Our Algorithm 3.1	0.0092	22.67	0.0590	163.2
Our Algorithm 3.2	0.0122	33.12	0.1343	399.8
TLQ Algorithm 3.1	0.0333	86.62	0.1160	270.69
TLQ Algorithm 3.3	0.0268	68	0.0950	217.56
TLQ Algorithm 3.4	0.0305	87.35	0.2560	401.28
TLQ Algorithm 3.6	0.0244	69.33	0.1465	247.14
Jolaoso Algorithm 3.11	0.0175	47.31	0.1143	179.55
Thong et al. Algorithm 3.1	0.0148	36.92	0.0780	214.6

et al.'s Algorithm 3.1 [34]) require fewer iterations and less computation time than algorithms employing the Armijo type step size criterion. However, this conclusion does not necessarily hold when solving Example 5.2 where the terminal function is nonlinear because algorithms that use a simple step size criterion may require a higher number of iterations to reach the error criterion, which leads to more computation time required. In terms of these 100 tests for Examples 5.1 and 5.2, our Algorithm 3.1 is generally preferred, see Table 4.

5.2. Applications to signal processing problems

In this subsection, we apply the proposed algorithms to deal with signal processing problems arising in real-world and measure the computational efficiency of the algorithms using performance profiles [35].

Example 5.3: The signal recovery problem is a practical one as signals in the real-world can be subject to interference during transmission and thus the original clean signals need to be recovered from noisy signals. Recall that the model for signal processing problems is as follows

$$\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{g}, \tag{43}$$

where $\mathbf{x} \in \mathbb{R}^n$ with k non-zero elements is the original signal, $\mathbf{y} \in \mathbb{R}^m$ is the observed noisy signal, $\mathbf{B} : \mathbb{R}^{m\times n}$ is a bounded linear operator, and $\mathbf{g} \in \mathbb{R}^m$ is the noisy observation. An important consideration in this problem is that the signal $\mathbf{x}$ is sparse, i.e. the number of non-zero elements in the signal $\mathbf{x}$ is much smaller than the dimension of $\mathbf{x}$ (that is $k \ll n$). A well-known model for solving problem (43) is the Least Absolute Shrinkage and Selection Operator (LASSO), which has the following expression

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) := \frac{1}{2}\|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{subject to } \|\mathbf{x}\|_1 \le t, \quad t > 0, \tag{44}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ represent the 2-norm and 1-norm defined in Euclidean spaces, respectively. Notice that the problem (44) can be converted to a variational inequality problem (VIP). Indeed, according to the optimal condition for convex optimization, it is known that the optimal solution $\mathbf{x}^*$ of the problem (44) is equivalent to the fact that all feasible directions at that point are not gradient descent directions. That is, finding solutions to the problem (44) is equivalent to solving the following problem (45):

$$\text{find } \mathbf{x}^* \in C \text{ such that } \nabla f\left(\mathbf{x}^*\right)^\top \left(\mathbf{x} - \mathbf{x}^*\right) \ge 0, \quad \forall \mathbf{x}^* \in C. \tag{45}$$

Notice that the problem (45) described above is in fact a variational inequality problem about ∇f on the feasible set C. The gradient of f is known to be $\nabla f(\mathbf{x}) = \mathbf{B}^\top(\mathbf{B}\mathbf{x} - \mathbf{y})$, so we set $A(\mathbf{x}) = \mathbf{B}^\top(\mathbf{B}\mathbf{x} - \mathbf{y})$ and $C = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_1 \le t\}$ in the proposed algorithms. It can be checked that A is monotone and L-Lipschitz continuous with $L = \|\mathbf{B}^\top\mathbf{B}\|$. We can now use the suggested algorithms to solve the problem (43). The parameters of all algorithms are set as follows.

- Choose $\tau = (n-1)/n$, $\epsilon_n = 100/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 0.01/(n+1)$, $\beta = 0.8$, $\chi_1 = 0.006$, $\mu = 0.6$, $\delta_n = 1 + 1/n$, $\xi_n = 1 + 1/(n+1)^{1.1}$, and $\rho_n = 0$ for the proposed Algorithms 3.1 and 3.2.
- Take $\tau = (n-1)/n$, $\epsilon_n = 100/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 0.01/(n+1)$, $\alpha_n = 0.8(1-\theta_n)$, $f(x) = 0.1x$, $\delta = 2$, $\zeta = 0.5$, and $\mu = 0.6$ for TLC Algorithm 3.1, TLC Algorithm 3.3, TLC Algorithm 3.4, and TLC Algorithm 3.6 [29].
- Set $\alpha = 1$, $\epsilon_n = 100/(n+1)^2$, $\sigma = 1.5$, $\theta_n = 0.01/(n+1)$, $f(x) = 0.1x$, $\delta = 2$, $\zeta = 0.5$, and $\mu = 0.6$ for Jolaoso's Algorithm 3.11 [19].
- Select $\chi_1 = 0.006$, $\tau = 0.1(\sqrt{5} - 2)$, $\mu = 0.2(\frac{1-4\tau-\tau^2}{(1-\tau)^2})$, and $\rho_n = 1/(n+1)^{1.1}$ for the Algorithm 3.1 presented by Thong et al. [34].

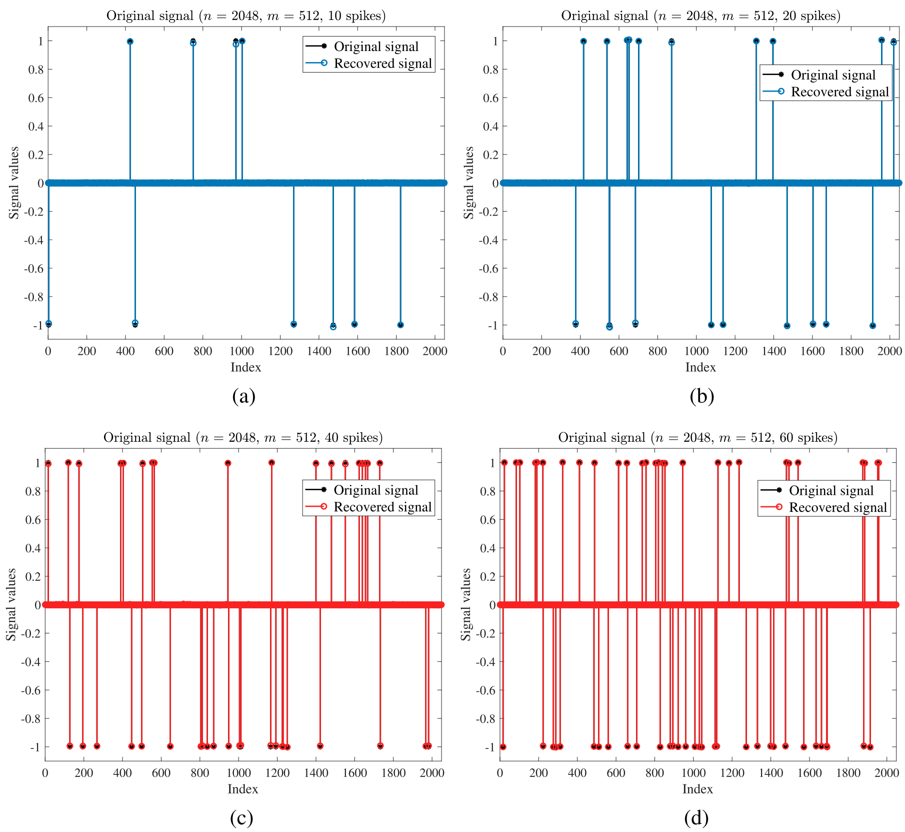

Figure 14. Signals with different sparsity recovered by our algorithms in Example 5.3. (a) Recovered by our Algorithm 3.1. (b) Recovered by our Algorithm 3.1. (c) Recovered by our Algorithm 3.2 and (d) Recovered by our Algorithm 3.2.

In our numerical experiments, the original signal $\mathbf{x} \in \mathbb{R}^n$ contains k ($k \ll n$) randomly generated ± 1 spikes, the matrix $\mathbf{B} : \mathbb{R}^{m\times n}$ is randomly produced by the MATLAB function `randn(m,n)` with a standard normal distribution, and $\mathbf{g} = 10^{-3}$`randn(m,1)`. Hence the observation $\mathbf{y}$ is created by $\mathbf{y} = \mathbf{Bx} + \mathbf{g}$. We use the mean squared error defined as $\text{MSE} = (1/n)\|\mathbf{x}^* - \mathbf{x}\|^2$ to measure the error accuracy between the signal $\mathbf{x}^*$ recovered by the algorithms and the original signal $\mathbf{x}$. The recovery procedure for all algorithms starts from the initial signals $\mathbf{x}_0 = \mathbf{x}_1 = \mathbf{0}$ and stops the iteration when $\text{MSE} < 10^{-6}$ is satisfied. In our first test, we set $n = 2048$, $m = 512$, and $k = \{10, 20, 40, 60\}$, and choose $t = k$ in (44) for all algorithms. The computation time in seconds and the number of iterations required by the proposed algorithms and the comparison methods at different sparsity k are shown in Table 5. The results recovered by our algorithms for different sparsity signals are displayed in Figure 14. Furthermore, we also draw in Figure 15 the MSE curves with the number of iterations for all algorithms under different cases.

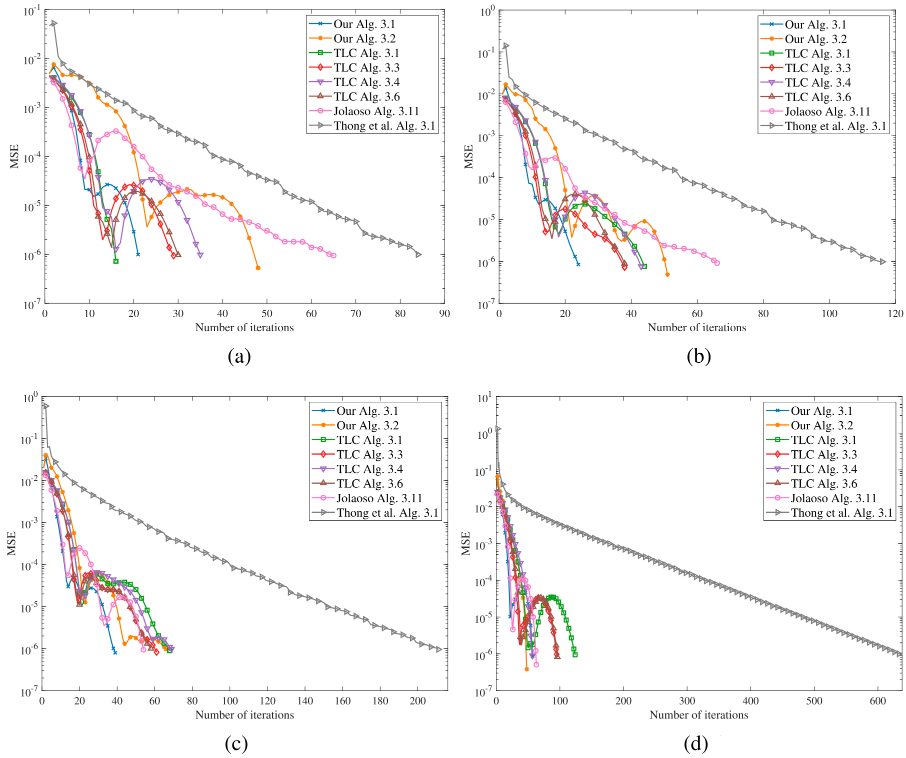

Figure 15. Numerical performance of MSE for all algorithms under different caces in Example 5.3. (a) $n = 2048$, $m = 512$, $k = 10$. (b) $n = 2048$, $m = 512$, $k = 20$. (c) $n = 2048$, $m = 512$, $k = 40$ and (d) $n = 2048$, $m = 512$, $k = 60$.

Table 5. Numerical results for all algorithms at different sparsity k in Example 5.3 ($n = 2048$, $m = 512$).

	$k = 10$		$k = 20$		$k = 40$		$k = 60$	
Algorithms	CPU (s)	Iter.	CPU (s)	Iter.	CPU (s)	Iter.	CPU (s)	Iter.
Our Algorithm 3.1	0.2705	21	0.1549	24	0.4714	39	0.6858	57
Our Algorithm 3.2	0.4875	48	0.2756	51	0.6288	66	0.4857	48
TLC Algorithm 3.1	0.8906	16	1.8140	44	4.1145	68	7.5798	124
TLC Algorithm 3.3	1.8974	29	1.6319	38	3.5870	61	5.8499	95
TLC Algorithm 3.4	2.1421	35	1.7778	43	3.9610	69	3.4818	57
TLC Algorithm 3.6	1.8118	30	1.5107	38	3.3098	58	5.8488	96
Jolaoso Algorithm 3.11	3.8757	65	2.5165	66	3.3553	54	3.9726	63
Thong et al. Algorithm 3.1	0.7137	84	0.5632	116	1.8454	211	5.3031	633

Finally, we use different dimensions n to test the computational efficiency of these algorithms in the case of keeping $m = 512$ and $k = 20$. We chose $n = \{1024, 1074, 1124, 1174, \ldots, 4074\}$ to generate 62 groups of execution times and number of iterations required for all algorithms to reach the stopping error MSE $< 10^{-6}$. The performance profiles based on execution time and number of iterations are presented in Figures 16 and 17, respectively.

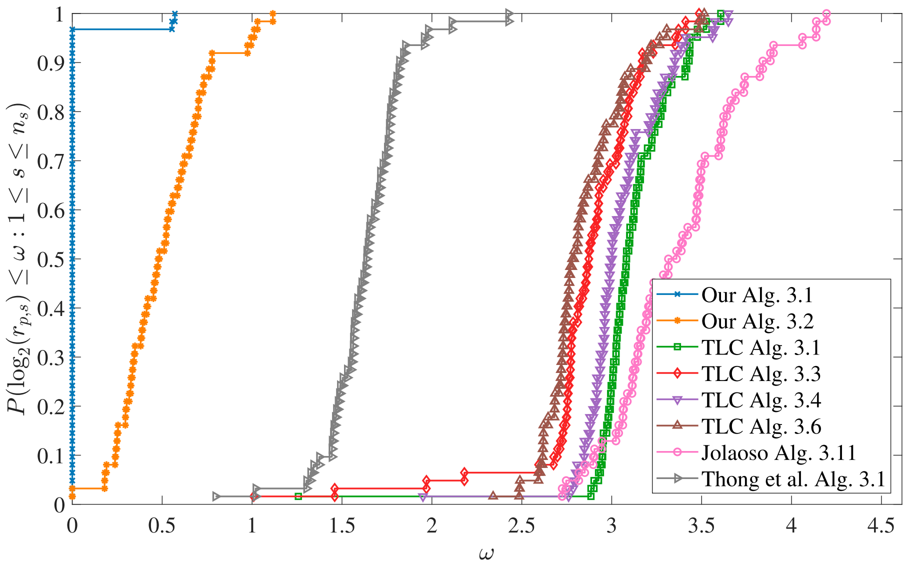

Figure 16. Performance profiles of all algorithms based on execution time for Example 5.3.

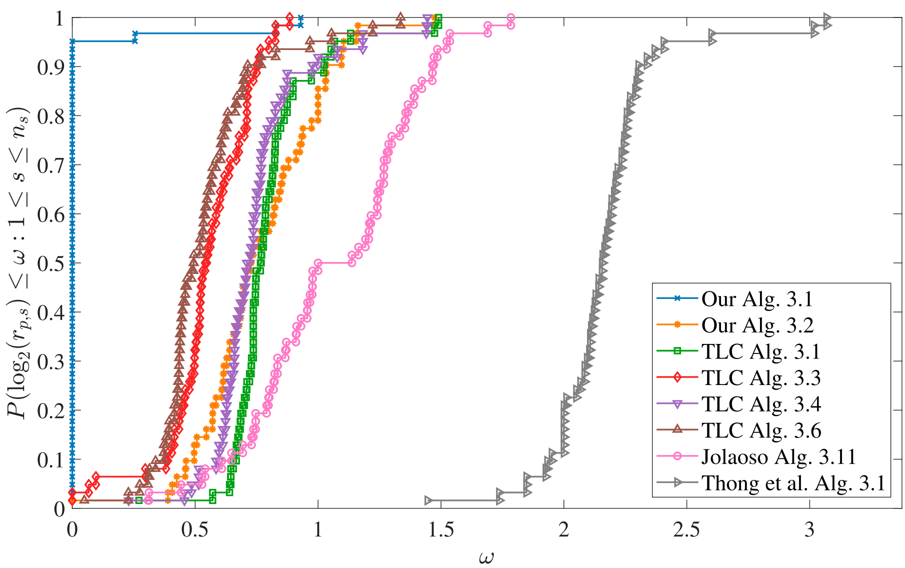

Figure 17. Performance profiles of all algorithms based on number of iterations for Example 5.3.

Remark 5.2: We have the following remarks about Example 5.3 in signal processing problems.

- The algorithms introduced in this paper work well in signal processing problems; see Table 5, Figures 14, and 15.

- The numerical results of the algorithms for different sparsity in the same dimension are shown in Table 5, from which it can be seen that, in general, the number of iterations and execution time required by the algorithms increases with increasing sparsity. This means that solvers generally require less computation time and fewer iterations when the original signal is sparser.
- As shown in Figures 16 and 17, our Algorithm 3.1 stands out if we are interested in the algorithm that can solve 95% problems with maximum efficiency. Figure 16 illustrates that if we choose $\omega = 2.5$ as the range of interest for all algorithms, then our Algorithms 3.1 and 3.2 and Thong et al.'s Algorithm 3.1 [34] perform better than the other methods, as shown by the height of their performance profile for $\omega = 2.5$ in Figure 16. It is worth noting that Thong et al.'s Algorithm 3.1 takes more iterations than Armijo-type algorithms, but it requires less computation time due to the use of a simple adaptive step size; see Figures 16 and 17.
- As can be seen in Figure 15 the convergence of the algorithms with inertial terms is oscillatory, i.e. the convergence is not monotonic. How to reduce this oscillation is a future work that can be considered.

6. Conclusions

In this paper, two new iterative schemes with inertial effects are proposed for solving pseudomonotone variational inequalities in infinite-dimensional Hilbert spaces. The proposed methods employ a new non-monotone step size criterion permitting them to work without the prior information about the Lipschitz constant of the mapping. Strong convergence theorems of the proposed algorithms are established under some suitable conditions. Finally, some numerical examples occurring in finite- and infinite-dimensional spaces verify the computational efficiency and robustness of the offered algorithms compared to some known schemes by means of performance profiles. Furthermore, numerical results of the proposed algorithms in optimal control problems and signal processing problems support this conclusion. The results obtained in this paper improved and extended many known ones in the field. We consider extending the work of this paper to Banach spaces and equilibrium problems in the future.

Acknowledgments

The authors are deeply grateful to the Editor and the anonymous referee for their careful reading, excellent insights and comments, which helped us to improve the quality of the original manuscript considerably.

Disclosure statement

The authors declare that they have no conflict of interest.

Funding

Bing Tan thanks the China Scholarship Council for the financial support (CSC No. 202106070094).

ORCID

Bing Tan http://orcid.org/0000-0003-1509-1809
Songxiao Li http://orcid.org/0000-0001-7911-2758

References

[1] Vuong PT, Shehu Y. Convergence of an extragradient-type method for variational inequality with applications to optimal control problems. Numer Algorithms. 2019;81(1):269–291.
[2] Chen J, Köbis E, Yao JC. Optimality conditions and duality for robust nonsmooth multiobjective optimization problems with constraints. J Optim Theory Appl. 2019;181(2):411–436.
[3] Chen J, Ju X, Köbis E, et al. Tikhonov type regularization methods for inverse mixed variational inequalities. Optimization. 2020;69(2):401–413.
[4] Ansari QH, Islam M, Yao JC. Nonsmooth variational inequalities on Hadamard manifolds. Appl Anal. 2020;99(2):340–358.
[5] Zhao X, Köbis MA, Yao Y, et al. A projected subgradient method for nondifferentiable quasiconvex multiobjective optimization problems. J Optim Theory Appl. 2021;190(1):82–107.
[6] Korpelevich GM. The extragradient method for finding saddle points and other problems. Èkonom i Mat Metody. 1976;12(4):747–756.
[7] Tseng P. A modified forward-backward splitting method for maximal monotone mappings. SIAM J Control Optim. 2000;38(2):431–446.
[8] He BS. A class of projection and contraction methods for monotone variational inequalities. Appl Math Optim. 1997;35(1):69–76.
[9] Censor Y, Gibali A, Reich S. The subgradient extragradient method for solving variational inequalities in Hilbert space. J Optim Theory Appl. 2011;148(2):318–335.
[10] Censor Y, Gibali A, Reich S. Extensions of Korpelevich's extragradient method for the variational inequality problem in Euclidean space. Optimization. 2012;61(9):1119–1132.
[11] Censor Y, Gibali A, Reich S. Strong convergence of subgradient extragradient methods for the variational inequality problem in Hilbert space. Optim Methods Softw. 2011;26(4–5):827–845.
[12] Malitsky Y. Projected reflected gradient methods for variational inequalities. SIAM J Optim. 2015;25(1):502–520.
[13] Dong QL, Jiang D, Gibali A. A modified subgradient extragradient method for solving the variational inequality problem. Numer Algorithms. 2018;79(3):927–940.
[14] Bauschke HH, Combettes PL. A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert spaces. Math Oper Res. 2001;26(2):248–264.
[15] Shehu Y, Dong QL, Jiang D. Single projection method for pseudo-monotone variational inequality in Hilbert spaces. Optimization. 2019;68(1):385–409.
[16] Gibali A, Thong DV, Tuan PA. Two simple projection-type methods for solving variational inequalities. Anal Math Phys. 2019;9(4):2203–2225.

[17] Thong DV, Gibali A. Two strong convergence subgradient extragradient methods for solving variational inequalities in Hilbert spaces. Jpn J Ind Appl Math. 2019;36(1):299–321.
[18] Cholamjiak P, Thong DV, Cho YJ. A novel inertial projection and contraction method for solving pseudomonotone variational inequality problems. Acta Appl Math. 2020;169(1):217–245.
[19] Jolaoso LO. An inertial projection and contraction method with a line search technique for variational inequality and fixed point problems. Optimizaiton. 2021. DOI:10.1080/02331934.2021.1901289.
[20] Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci. 2009;2(1):183–202.
[21] Chambolle A, Dossal C. On the convergence of the iterates of the "Fast Iterative Shrinkage/Thresholding Algorithm". J Optim Theory Appl. 2015;166(3):968–982.
[22] Attouch H, Cabot A. Convergence rates of inertial forward-backward algorithms. SIAM J Optim. 2018;28(1):849–874.
[23] Dong QL, Cho YJ, Zhong LL, et al. Inertial projection and contraction algorithms for variational inequalities. J Global Optim. 2018;70(3):687–704.
[24] Hieu DV, Strodiot JJ, Muu LD. An explicit extragradient algorithm for solving variational inequalities. J Optim Theory Appl. 2020;185(2):476–503.
[25] Shehu Y, Iyiola OS. Projection methods with alternating inertial steps for variational inequalities: weak and linear convergence. Appl Numer Math. 2020;157:315–337.
[26] Shehu Y, Gibali A. New inertial relaxed method for solving split feasibilities. Optim Lett. 2021;15(6):2109–2126.
[27] Gibali A, Jolaoso LO, Mewomo OT, et al. Fast and simple Bregman projection methods for solving variational inequalities and related problems in Banach spaces. Results Math. 2020;75(4):179.
[28] Jolaoso LO, Alakoya TO, Taiwo A, et al. Inertial extragradient method via viscosity approximation approach for solving equilibrium problem in Hilbert space. Optimization. 2021;70(2):387–412.
[29] Tan B, Li S, Cho SY. Inertial projection and contraction methods for pseudomonotone variational inequalities with non-Lipschitz operators and applications. Appl Anal. 2021. DOI:10.1080/00036811.2021.1979219.
[30] Yang J, Liu H. Strong convergence result for solving monotone variational inequalities in Hilbert space. Numer Algorithms. 2019;80(3):741–752.
[31] Liu H, Yang J. Weak convergence of iterative methods for solving quasimonotone variational inequalities. Comput Optim Appl. 2020;77(2):491–508.
[32] Hieu DV, Anh PK, Muu LD. Modified extragradient-like algorithms with new stepsizes for variational inequalities. Comput Optim Appl. 2019;73(3):913–932.
[33] Jolaoso LO, Shehu Y. Single Bregman projection method for solving variational inequalities in reflexive Banach spaces. Appl Anal. 2022;101(14):4807–4828. DOI:10.1080/00036811.2020.1869947.
[34] Thong DV, Long LV, Li XH, et al. A new self-adaptive algorithm for solving pseudomonotone variational inequality problems in Hilbert spaces. Optimization. 2021. DOI:10.1080/02331934.2021.1909584.
[35] Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. Math Program. 2002;91(2):201–213.
[36] Bauschke HH, Combettes PL. Convex analysis and monotone operator theory in Hilbert spaces. 2nd ed. New York: Springer; 2017.
[37] Beck A, Guttmann-Beck N. FOM–a MATLAB toolbox of first-order methods for solving convex optimization problems. Optim Methods Softw. 2019;34(1):172–193.

[38] Osilike MO, Aniagbosor SC. Weak and strong convergence theorems for fixed points of asymptotically nonexpansive mappings. Math Comput Model. 2000;32(10):1181–1191.
[39] Saejung S, Yotkaew P. Approximation of zeros of inverse strongly monotone operators in Banach spaces. Nonlinear Anal. 2012;75(2):742–750.
[40] Denisov SV, Semenov VV, Chabak LM. Convergence of the modified extragradient method for variational inequalities with non-Lipschitz operators. Cybernet Syst Anal. 2015;51(5):757–765.
[41] Tan B, Qin X, Yao JC. Strong convergence of inertial projection and contraction methods for pseudomonotone variational inequalities with applications to optimal control problems. J Global Optim. 2022;82(3):523–557.
[42] Jolaoso LO, Taiwo A, Alakoya TO, et al. A strong convergence theorem for solving pseudo-monotone variational inequalities using projection methods. J Optim Theory Appl. 2020;185(3):744–766.
[43] Hieu DV, Cho YJ, Xiao YB, et al. Modified extragradient method for pseudomonotone variational inequalities in infinite dimensional Hilbert spaces. Vietnam J Math. 2021;49(4):1165–1183.
[44] Pietrus A, Scarinci T, Veliov VM. High order discrete approximations to Mayer's problems for linear systems. SIAM J Control Optim. 2018;56(1):102–119.
[45] Bressan B, Piccoli B. Introduction to the mathematical theory of control. Springfield: American Institute of Mathematical Sciences (AIMS); 2007.