Numer. Math. Theor. Meth. Appl. Vol. **xx**, No. x, pp. 1-23
doi: 10.4208/nmtma.OA-2025-0068 x 2026

A Resampling-Free Stochastic Projection Contraction Algorithm for Solving Stochastic Variational Inequalities

Ting Li[1], Xingju Cai[2] and Deren Han[1,*]

[1] *LMIB of the Ministry of Education, School of Mathematical Sciences, Beihang University, Beijing 100191, P.R. China*
[2] *Ministry of Education Key Laboratory of NSLSCS, School of Mathematical Sciences, Nanjing Normal University, Nanjing 210023, P.R. China*

Received 17 July 2025; Accepted (in revised version) 18 December 2025

Abstract. Sampling is a major computational bottleneck in stochastic algorithms. This paper proposes a stochastic projection contraction algorithm for stochastic variational inequality problems, significantly reducing runtime by eliminating resampling in the correction step. We introduce an adjustable offset weight to optimize search direction, along with different adaptive step size strategies in prediction and correction steps. We further present discrete differential equation interpretations for specific offset weight values. To address bias due to the absence of resampling in the correction step, we develop an error control scheme and provide convergence guarantees. Numerical experiments demonstrate the algorithm's efficiency.

AMS subject classifications: 90C15, 49J40, 49K45, 93E35

Key words: Stochastic projection contraction algorithm, stochastic variational inequality, resampling-free, adaptive stepsize, search direction selection.

1. Introduction

Define a probability space $(X, \mathbb{F}, \mathbb{P})$ and a support set Ξ of ξ. $\xi : X \to \Xi$ is a random variable with distribution $\mathbf{P}$, where $\mathbf{P} := \mathbb{P} \circ \xi^{-1}$ is the probability measure on Ξ induced by ξ. We use $\mathbb{E}$ to represent the expectation with respect to the probability measure $\mathbf{P}$. The stochastic variational inequality (SVI) problem considered in this paper is constructed as

$$\langle x - x^*, F(x^*)\rangle \geq 0, \quad \forall x \in \Omega, \tag{1.1}$$

where $F(x^*) = \mathbb{E}_\xi[T(x^*, \xi)]$, $\Omega \subset \mathbb{R}^n$ is a non-empty closed convex set, and $x^* \in \Omega$

*Corresponding author. *Email addresses:* handr@buaa.edu.cn (D. Han), liting_math@buaa.edu.cn (T. Li), caixingju@njnu.edu.cn (X. Cai)

is a solution of this SVI. Numerous stochastic problems, including stochastic differentiable optimization problems, stochastic complementarity problems, stochastic saddle point problems, stochastic equilibrium problems, and stochastic system of equations problems can all be formulated as SVI problems [4, 9, 10, 13, 22–24, 30]. Therefore, it is of great significance to study efficient numerical algorithms for SVI problems.

Stochastic approximation (SA) methods [14, 15, 25] are highly effective for solving the SVI problems. Unlike the sample average approximation (SAA) methods [28, 31], which approximate the expectation operator $F(x)$ using large batches of external offline samples $(\sum_{i=1}^{N} T(x, \xi_i))/N$, SA methods employ internal online stochastic operators $\widehat{T}(x^k, \xi^k)$ at each iteration to approximate $F(x^k)$. This approach often significantly reduces computational costs compared to SAA. The earliest SA method is stochastic gradient descent algorithm, proposed by Robbins and Monro [27] for solving stochastic equation systems. Subsequently, efficient SA methods for the SVI problems, such as the stochastic projected gradient (SPG) algorithm [17], stochastic Tikhonov regularization algorithm [18], stochastic extra-gradient (SEG) algorithm [14, 15, 19], and stochastic extra-point (SEP) algorithm [13], just to name a few, have been studied.

SEG algorithm has attracted widespread attention due to its great numerical performance and relatively weak assumptions on the problem. However, each iteration of SEG requires computing two projections, which significantly increases the computational cost for problems where projection operations are expensive. To overcome this drawback, Bot *et al.* [3] proposed the stochastic forward-backward-forward (SFBF) algorithm, whose iteration format can be written as

$$
\begin{aligned}
&\text{prediction}: \\
&z^k = \Pi\left(x^k - h_k \frac{1}{N_k}\sum_{i=1}^{N_k} T\big(x^k, \xi_i^k\big)\right), \\
&\text{correction}: \\
&x^{k+1} = x^k - \left(x^k - z^k - h_k\left(\frac{1}{N_k}\sum_{i=1}^{N_k} T\big(x^k, \xi_i^k\big) - \frac{1}{N_k}\sum_{i=1}^{N_k} T\big(z^k, \eta_i^k\big)\right)\right),
\end{aligned}
\tag{1.2}
$$

where $\Pi(x)$ denotes the projection of x on the constraint set Ω, i.e.,

$$\Pi(x) := \operatorname{argmin}_{y\in\Omega} \|y - x\|^2.$$

However, both SEG algorithm and SFBF algorithm (1.2) require two independent samples per iteration. While this two-sample approach ensures that all the stochastic operators in the iteration remain unbiased estimators of the expected operator—a property that greatly simplifies theoretical analysis—the sampling process in stochastic algorithms is computationally expensive, making this approach particularly time-consuming. To reduce the time cost, we propose using only a single sample per iteration, which introduces substantial challenges in the theoretical analysis of the algorithm.

Huang and Zhang [13] demonstrate that incorporating search directions beyond simple gradient information, such as using optimism gradient [7] or extra-gradient, is indeed important in practical numerical experiments. Their proposed SEP algorithm outperforms many stochastic first-order methods by employing a broader spectrum of search directions. In (1.2), at the current iteration point x^k the search direction is $x^k - z^k - h_k((\sum_{i=1}^{N_k} T(x^k, \xi_i^k))/N_k - (\sum_{i=1}^{N_k} T(z^k, \eta_i^k))/N_k)$. This direction uses $-h_k((\sum_{i=1}^{N_k} T(x^k, \xi_i^k)/N_k) - (\sum_{i=1}^{N_k} T(z^k, \eta_i^k))/N_k)$ as an offset of direction $x^k - z^k$. Assigning an appropriate weight to this offset can yield a favorable search direction, and this paper will investigate this weight. Recent research has revealed deep connections between optimization problem and ordinary differential equation (ODE) [1,6,8,26,29, 32]. Through analysis of ODE discretization schemes, we explain that offset weight near $1/2$ achieve superior approximation accuracy. In practical computations, the optimal value of the offset weight depend on different function classes, and here we will also provide a random selection method for the weight to allow the use of different search directions.

Securing an appropriate step size is equally crucial after obtaining a high-quality search direction [5,11]. For example, when using the same SEG algorithm, the adaptive step size in [15] demonstrates significantly better numerical performance compared to the constant step size in [14]. What is more, when the correction step uses an adaptive step size different from that of the prediction step, the numerical performance is further significantly improved [19]. Thus, we consider use different adaptive step sizes in prediction and correction step.

The main contributions of this paper are as follows:

1. For the SVI problems, we propose a stochastic projection contraction algorithm that requires only one projection and one sample per single-step iteration. To address the biased estimation of the expectation operator caused by omitting resampling in the correction step, we develop a comprehensive error control scheme with rigorous convergence guarantees.

2. The algorithm determines search ranges and optimal search directions by tuning the offset weight, with several special cases of this weight receiving interpretation through differential equation discretization. Through specific weight assignments, we naturally derived randomized versions of two deterministic algorithms. The prediction and correction steps use different adaptive step size strategies to further enhance algorithmic efficiency.

3. We demonstrate the superiority of our algorithm through two numerical experiments, and the results indicate that the numerical performance is generally better when the offset weight is around $1/2$. This also supports our explanation from the ODE perspective.

The rest of this paper is arranged as follows. Section 2 consists of two subsections. Subsection 2.1 is some basic notations and properties. Subsection 2.2 provides a theoretical explanation of some special offset weights. Section 3 presents the stochastic improved projection contraction algorithm to solve the SVI problem. The proof of its convergence is also given. Section 4 gives a numerical experiment to show the superiority of the proposed algorithm. Section 5 is the conclusion.

2. Preliminaries

2.1. Basic notions and properties

We give some basic notions, definitions, and properties in this subsection that are useful in the subsequent paper.

For vectors $x, y \in \mathbb{R}^n$, $\langle x, y\rangle$ is the standard inner product. $\|x\| = \sqrt{\langle x, x\rangle}$ is the Euclidean norm. Given a random variable ξ and a σ-algebra $\mathcal{F}$, we denote $\mathbb{E}[\xi]$ as the expectation of ξ, $\mathbb{E}[\xi|\mathcal{F}]$ as expectation of ξ conditional to $\mathcal{F}$. For $q \geq 1$, $|\xi|_q := \sqrt[q]{\mathbb{E}[|\xi|^q]}$ is the $\mathcal{L}_q$ norm of ξ. We denote $\sigma(\xi_1, \cdots, \xi_k)$ as the σ-algebra generated by the random variables $\{\xi_i\}_{i=1}^k$. We denote "$\xi \in \mathcal{F}$" if ξ is $\mathcal{F}$-measurable, "a.s." for "almost surely". Given $m \in \mathbb{R}$, $\lceil m \rceil$ denotes the smallest integer greater than or equal to m.

Let $H : \mathbb{R}^n \to \mathbb{R}^n$ be a single-valued operator. H is called monotone, if

$$\langle x - y, H(x) - H(y)\rangle \geq 0, \quad \forall x, y \in \mathbb{R}^n.$$

H is called L-Lipschitz continuous, if

$$\|H(x) - H(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

H is called σ-cocoercive with $\sigma > 0$, if

$$\langle x - y, H(x) - H(y)\rangle \geq \sigma\|H(x) - H(y)\|^2, \quad \forall x, y \in \mathbb{R}^n.$$

From these definitions, it implies that if H is σ-cocoercive, then H is monotone and $1/\sigma$-Lipschitz continuous.

Let $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ be a set-valued operator from $\mathbb{R}^n$ to $\mathbb{R}^n$. Its graph is defined as $\mathrm{gra}A = \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^n : u \in Ax\}$. A is called monotone if $\langle x - y, u - v\rangle \geq 0$, $\forall (x, u), (y, v) \in \mathrm{gra}A$. A is called maximal monotone if there exists no monotone operator B such that $\mathrm{gra}B$ properly contains $\mathrm{gra}A$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. The subdifferential of f is the set-valued operator

$$\partial f(x) = \{u \in \mathbb{R}^n : f(y) \geq f(x) + \langle y - x, u\rangle, \forall y \in \mathbb{R}^n\}, \quad \forall x \in \mathbb{R}^n.$$

Let $\Omega \subset \mathbb{R}^n$ be a closed convex set. The indicator function on the set Ω is defined as

$$I_\Omega(x) := \begin{cases} 0, & x \in \Omega, \\ \infty, & x \notin \Omega. \end{cases}$$

Then, $I_\Omega(x)$ is a closed convex function. The set

$$\mathcal{N}_\Omega(x) := \begin{cases} \left\{d \,|\, (y - x)^T d \leq 0, \forall y \in \Omega\right\}, & x \in \Omega, \\ \emptyset, & x \notin \Omega \end{cases}$$

is called the normal cone to Ω at x. As is well known, $\mathcal{N}_\Omega$ is the subdifferential operator of the indicator function I_Ω, and $\mathcal{N}_\Omega$ is maximal monotone. For any constant $c > 0$, $\mathrm{dom}(I+c\mathcal{N}_\Omega)^{-1} = \mathbb{R}^n$, and $(I+c\mathcal{N}_\Omega)^{-1}$ is singled-valued. Furthermore, $(I+c\mathcal{N}_\Omega)^{-1}(x) = \Pi(x)$, $\forall x \in \mathbb{R}^n$.

For the SVI problem (1.1), we define the error function

$$e(x,h) := \|x - \Pi(x - hF(x))\|,$$

where $h > 0$. The following properties are well known.

Lemma 2.1 ([14, Lemma 2.1]). *Let $\Omega \subset \mathbb{R}^n$ be a non-empty closed convex set. The following conclusions hold:*

(i) For all $x, y \in \mathbb{R}^n$, $\|\Pi(x) - \Pi(y)\| \le \|x - y\|$.

(ii) Denote $SOL(\Omega, F)$ as the solution set of SVI(Ω, F). Then, for all $h > 0, SOL(\Omega, F) = \{x \in \mathbb{R}^n : e(x,h) = 0\}$.

(iii) Given $x \in \mathbb{R}^n$, $\Pi(x)$ is the unique point that satisfies $\langle x - \Pi(x), y - \Pi(x)\rangle \le 0$, $\forall y \in \Omega$.

Lemma 2.2 ([9, Proposition 10.3.6]). *For fixed $x \in \mathbb{R}^n$, $e(x,h)$ is non-decreasing on $h \in (0,\infty)$, while the function $e(x,h)/h$ is non-increasing on $h \in (0,\infty)$.*

The following supermartingale convergence theorem is significant for the convergence analysis of our algorithm, and can be found in [2, 14, 15, 23].

Lemma 2.3 ([2, Proposition 8.2.10]). *Let $\{y^k\}, \{u^k\}, \{a^k\}, \{b^k\}$ be sequences of nonnegative integrable random variables, and let $\mathcal{F}^k, k = 0, 1, 2, \dots$ be sets of random variables such that $\mathcal{F}^k \subset \mathcal{F}^{k+1}$ for all k. Assume that:*

(i) y^k, u^k, a^k, b^k are random variables in $\mathcal{F}^k$.

(ii) For each k, $\mathbb{E}[y^{k+1}|\mathcal{F}^k] \le (1 + a^k)y^k - u^k + b^k$.

(iii) $\sum a^k < \infty$, $\sum b^k < \infty$.

Then, a.s. $\{y^k\}$ converges and $\sum u^k < \infty$.

2.2. Explanation of the offset weight from ODE perspective

Building upon the ODE-based analysis for deterministic algorithms solving unconstrained optimization in [20], and the theoretical concepts in Section 2.1, we provide an interpretation of the offset weight in our algorithm for constrained stochastic optimization problems (a special case of SVI problem).

Consider the constrained smooth stochastic convex optimization problem

$$\begin{aligned} \min_x \quad & f(x) = \mathbb{E}_\xi[g(x,\xi)] \\ \text{s.t.} \quad & x \in \Omega. \end{aligned} \tag{2.1}$$

It is equivalent to problem $\min_x \mathbb{E}_\xi[g(x,\xi)] + I_\Omega(x)$. The corresponding first order ODE is

$$\begin{cases} \dot{x}(t) \in -\nabla \mathbb{E}_\xi[g(x(t),\xi)] - \mathcal{N}_\Omega(x(t)), \quad t \geq 0, \\ x(0) = x^0. \end{cases} \tag{2.2}$$

The equilibrium points $x^* = \lim_{t\to\infty} x(t)$ of (2.2) are the optimal points of optimization problem (2.1), where $x(t)$ is the solution trajectory of (2.2) [6, 26]. We now discretize (2.2) using the approximate trapezoidal formula, i.e.,

$$\frac{x^{k+1} - x^k}{h} \in -\frac{1}{2}\left(\nabla \mathbb{E}_\xi[g(x^k,\xi)] + \nabla \mathbb{E}_\xi[g(x^{k+1},\xi)] + \mathcal{N}_\Omega(x^{k+1})\right).$$

Using stochastic gradient to approximate expectation gradient while employing the same random sample to reduce computational costs, we obtain

$$x^{k+1} \in x^k - h\nabla g(x^k,\xi^k) - h\mathcal{N}_\Omega(x^{k+1}) - \frac{h}{2}\nabla g(x^{k+1},\xi^k) + \frac{h}{2}\nabla g(x^k,\xi^k). \tag{2.3}$$

However, (2.3) is an implicit update. If we take the point obtained by the SPG algorithm (a stochastic approximation of the Euler discretization scheme for the ODE (2.2)): $z^k = \Pi(x^k - h\nabla g(x^k,\xi^k))$ as the approximation of the point x^{k+1}, combining with $\Pi = (I + h\mathcal{N}_\Omega)^{-1}$, i.e. $z^k \in x^k - h\nabla g(x^k,\xi^k) - h\mathcal{N}_\Omega(z^k)$. Then (2.3) becomes

$$x^{k+1} = z^k + \frac{h}{2}\left(\nabla g(x^k,\xi^k) - \nabla g(z^k,\xi^k)\right),$$

which is algorithm

$$x^{k+1} = x^k - \left(x^k - z^k - \beta h\left(\nabla g(x^k,\xi^k) - \nabla g(z^k,\xi^k)\right)\right) \tag{2.4}$$

with offset weight $\beta = 1/2$. Note that when $\beta = 1$, it corresponds to the SFBF algorithm using the same sample, and when $\beta = 0$, it corresponds to the SPG algorithm. Since the trapezoidal formula has higher order of global discretization error than Euler scheme [21], the search direction in algorithm (2.4) should have better performance compared with $\beta = 0$. To further enhance the algorithm's numerical performance, we employ different line search step sizes in the prediction and correction steps. This leads to the following generalized algorithm:

$$\begin{aligned} z^k &= \Pi\left(x^k - h_k\nabla g(x^k,\xi^k)\right), \\ x^{k+1} &= x^k - \eta\alpha_k\left(x^k - z^k - h_k\beta\left(\nabla g(x^k,\xi^k) - \nabla g(z^k,\xi^k)\right)\right). \end{aligned}$$

3. Stochastic improved projection contraction algorithms

In this section, we give the stochastic improved projection contraction (S-IPC) algorithm to solve SVI problem (1.1). The related convergence analysis is also given.

Algorithm 3.1 (S-IPC)

1: Initialize $x^0 \in \mathbb{R}^n$, $0 < \mu < \nu < 1$, $0 < \underline{h} < 1 \leq \gamma_0^0 \leq \overline{h} < 4\sigma$, $0 \leq \beta \leq 1$, $\theta \in (0,1)$, $\tau > 1$, $\eta \in (0,2)$, $\epsilon > 0$, sample rate $\{N_k\} \subset \mathbb{N}$ satisfying $\sum_{k=0}^{\infty} 1/\sqrt{N_k} < \infty$, $k = 0$.
2: **if** $e(x^k, 1) < \epsilon$ **then** stop
3: **else** go to step 5.
4: **end if**
5: Generate samples $\xi^k := \{\xi_j^k\}_{j=1}^{N_k}$ from $\mathbf{P}$.
$\widehat{T}(x^k, \xi^k) = (\sum_{j=1}^{N_k} T(x^k, \xi_j^k))/N_k$.
Line search for h_k.
$l = 0$.
6: **while** $r_k(\gamma_l^k) := \frac{\gamma_l^k \|\widehat{T}(z^k(\gamma_l^k),\xi^k) - \widehat{T}(x^k,\xi^k)\|}{\|z^k(\gamma_l^k) - x^k\|} > \nu$,
where $z^k(\gamma_l^k) = \Pi(x^k - \gamma_l^k \widehat{T}(x^k, \xi^k))$ **do**
$\gamma_{l+1}^k = \gamma_l^k \theta * \min\{1, 1/r_k(\gamma_l^k)\}$, $l = l+1$.
7: **end while**
$h_k = \gamma_l^k$.
8: Generate new iteration point.

$$\text{prediction}: \quad z^k = \Pi(x^k - h_k \widehat{T}(x^k, \xi^k)), \tag{3.1}$$

$$\text{correction}: \quad x^{k+1} = x^k - \eta \alpha_k d^k, \tag{3.2}$$

where

$$\begin{aligned} d^k &= x^k - z^k - h_k \beta \left(\widehat{T}(x^k, \xi^k) - \widehat{T}(z^k, \xi^k) \right), \\ \alpha_k &= \frac{\varphi_k}{\|d^k\|^2}, \\ \varphi_k &= (1-\beta) \left(1 - \frac{h_k}{4\sigma} \right) \|x^k - z^k\|^2 \\ &\quad + \beta \left\langle x^k - z^k, x^k - z^k - h_k \left(\widehat{T}(x^k, \xi^k) - \widehat{T}(z^k, \xi^k) \right) \right\rangle. \end{aligned}$$

9: Adaptively select γ_0^{k+1}, the initial line search step size for iteration $k+1$.
10: **if** $r_k(h_k) \leq \mu$ **then**
$\gamma_0^{k+1} = \tau h_k$
11: **else**
$\gamma_0^{k+1} = h_k$.
12: **end if**
$\gamma_0^{k+1} = \mathbf{P}_{[\underline{h}, \overline{h}]}(\gamma_0^{k+1})$.
$k = k+1$. Go to step 5.

Remark 3.1. (i) In step 5, if $\|x^k - z^k(\gamma_l^k)\| = 0$, we also accept γ_l^k and set $h_k = \gamma_l^k$.

(ii) Algorithm 3.1 naturally produces stochastic versions of two deterministic algorithms as byproducts. Specifically, when $\beta = 0$, Algorithm 3.1 is the stochastic version of the relaxed projection gradient (RPG) algorithm in [16]. When $\beta = 1$, Algorithm 3.1 is the stochastic version of the projection contraction (PC) algorithm in [12].

Remark 3.2. The proposed algorithm has the following characteristics:

(i) Algorithm 3.1 needs only one projection and one sampling per iteration. This significantly reduces computational costs, particularly beneficial for stochastic algorithms and problems where projection operations are expensive.

(ii) The correction step omits projection, meaning the generated sequence may not always maintain feasibility. The correction step reuses the same sampling, making its stochastic gradient a biased estimator of the expected gradient. These characteristics introduce additional challenges in proving the algorithm's convergence.

We next give the assumptions required by Algorithm 3.1.

Assumption 3.1. (i) The solution set $SOL(\Omega, F)$ of problem (1.1) is non-empty.

(ii) The operator F is σ-cocoercive.

(iii) For almost every $\xi \in \Xi$, the operator $T(\cdot, \xi)$ is $L(\xi)$-Lipschitz continuous and $\mathbb{E}[L^4(\xi)] < \infty$.

(iv) $\{\xi_j^k\}_{j=1}^{N_k}$ are independent identically distributed sampling.

(v) The sample rate $\{N_k\}$ increases with the number of iteration k, and satisfies $\sum_{k=0}^{\infty}(1/\sqrt{N_k}) < \infty$.

According to [14], the Lipschitz continuity of certain operators related to $T(\cdot, \xi)$ can be derived from the Lipschitz continuity of $T(\cdot, \xi)$ itself.

Remark 3.3. For almost every $\xi \in \Xi$, if $T(\cdot, \xi)$ is $L(\xi)$-Lipschitz continuous and $\mathbb{E}_\xi[T(\cdot, \xi)] = F(\cdot)$, then

(i) The operator $F(\cdot)$ is L-Lipschitz continuous, where $L = \mathbb{E}[L(\xi)]$.

(ii) $|\|\epsilon(\cdot, \xi)\||_q$ is L_q-Lipschitz continuous, where $\epsilon(\cdot, \xi) = T(\cdot, \xi) - F(\cdot)$ and $L_q = |L(\xi)|_q + L$.

(iii) $\widehat{T}(\cdot, \xi) := (\sum_{j=1}^{N} T(\cdot, \xi_j))/N$ is $\widehat{L}(\xi)$-Lipschitz continuous, where

$$\widehat{L}(\xi) = \frac{1}{N}\left(\sum_{j=1}^{N} L(\xi_j)\right).$$

Let $\mathcal{F}^k = \sigma(x^0, \xi^0, \cdots, \xi^{k-1})$ be the σ-algebra which relates to the generation of x^k, z^k and h_k. It can be seen that $x^k \in \mathcal{F}^k$, $h_k, z^k \in \mathcal{F}^{k+1}$. For brevity, we give the following notations:

$$\epsilon_1^k := \widehat{T}(x^k, \xi^k) - F(x^k), \quad \epsilon_2^k := \widehat{T}(z^k, \xi^k) - F(z^k). \tag{3.3}$$

Similar to [19], the following results of the finite terminability of line search and the lower bound of the step size h_k can be obtained.

Remark 3.4. Suppose that Assumption 3.1(iii) holds, then the line search in Algorithm 3.1 terminates in finite steps.

Remark 3.5. Suppose that Assumptions 3.1(iii)-3.1(iv) hold, then

(i) $h_k \geq \min\{\underline{h}, \nu\theta/(\widehat{L}^2(\xi^k)\overline{h})\}$ a.s., where $\widehat{L}(\xi^k) = (\sum_{j=1}^{N_k} L(\xi_j^k))/N_k$.

(ii) $\mathbb{E}[h_k^2|\mathcal{F}^k] \geq \min\{\underline{h}^2, (\nu^2\theta^2)/\overline{h}^2\} \cdot (1/\mathbb{E}[L^4(\xi)]) := h_{\min}^2$.

We next demonstrate the property of the direction d^k in (3.2) though a lemma.

Lemma 3.1. *Suppose that Assumptions* 3.1*(i)*-3.1*(ii) holds, then the sequences* $\{x^k\}, \{z^k\}$ *generated by Algorithm* 3.1 *satisfy*

$$\begin{aligned} \langle x^k - x^*, d^k\rangle &\geq \varphi_k + h_k(1-\beta)\langle z^k - x^*, \epsilon_1^k\rangle + \beta h_k\langle z^k - x^*, \epsilon_2^k\rangle & (3.4)\\ &\geq \kappa_1\|x^k - z^k\|^2 + h_k(1-\beta)\langle z^k - x^*, \epsilon_1^k\rangle + \beta h_k\langle z^k - x^*, \epsilon_2^k\rangle, & (3.5)\end{aligned}$$

where

$$\kappa_1 = (1-\beta)\left(1 - \frac{h_k}{4\sigma}\right) + \beta(1-\nu) > 0,$$

φ_k *can be recalled from step 8 in Algorithm* 3.1.

Proof. According to the definition of d^k, we have

$$\langle x^k - x^*, d^k\rangle = \left\langle x^k - x^*, x^k - z^k - h_k\beta\left(\widehat{T}(x^k, \xi^k) - \widehat{T}(z^k, \xi^k)\right)\right\rangle. \tag{3.6}$$

On the one hand, due to the iteration format of (3.1) and Lemma 2.1(iii), we have

$$\left\langle x - z^k, z^k - x^k + h_k\widehat{T}(x^k, \xi^k)\right\rangle \geq 0, \quad \forall x \in \Omega. \tag{3.7}$$

Setting $x = x^* \in \Omega$ in (3.7), $x = z^k \in \Omega$ in (1.1) and adding these two inequalities, we obtain

$$\left\langle z^k - x^*, x^k - z^k - h_k\widehat{T}(x^k, \xi^k) + h_kF(x^*)\right\rangle \geq 0.$$

Using the notation in (3.3), we can rearrange the above inequality as

$$\begin{aligned} \langle x^k - x^*, x^k - z^k\rangle &\geq \left\langle x^k - z^k, x^k - z^k - h_k\widehat{T}(x^k, \xi^k) + h_kF(x^*)\right\rangle \\ &\quad + \left\langle x^k - x^*, h_k\widehat{T}(x^k, \xi^k) - h_kF(x^*)\right\rangle \\ &= \|x^k - z^k\|^2 + \left\langle x^k - z^k, h_kF(x^*) - h_kF(x^k)\right\rangle - \langle x^k - z^k, h_k\epsilon_1^k\rangle \\ &\quad + \langle x^k - x^*, h_k\epsilon_1^k\rangle + \left\langle x^k - x^*, h_kF(x^k) - h_kF(x^*)\right\rangle. \end{aligned} \tag{3.8}$$

According to the property that F is σ-cocoercive and the inequality $2ab \geq -a^2 - b^2$ with

$$a = \sqrt{\frac{h_k}{4\sigma}}(x^k - z^k), \quad b = \sqrt{\sigma h_k}\big(F(x^k) - F(x^*)\big),$$

the inequality (3.8) can be scaled as

$$\begin{aligned}\langle x^k - x^*, x^k - z^k\rangle &\geq \|x^k - z^k\|^2 - \frac{h_k}{4\sigma}\|x^k - z^k\|^2 - h_k\sigma\|F(x^k) - F(x^*)\|^2 \\ &\quad + \big\langle z^k - x^*, h_k\epsilon_1^k\big\rangle + h_k\sigma\|F(x^k) - F(x^*)\|^2 \\ &= \left(1 - \frac{h_k}{4\sigma}\right)\|x^k - z^k\|^2 + \big\langle z^k - x^*, h_k\epsilon_1^k\big\rangle. \end{aligned} \tag{3.9}$$

On the other hand, setting $x = x^* \in \Omega$ in (3.7), and making an identity transformation, we obtain

$$\big\langle x^* - z^k, z^k - x^k + h_k\widehat{T}(x^k, \xi^k) - h_k\widehat{T}(z^k, \xi^k) + h_k\widehat{T}(z^k, \xi^k)\big\rangle \geq 0.$$

This means

$$\begin{aligned}&\big\langle x^k - x^*, x^k - z^k - h_k\widehat{T}(x^k, \xi^k) + h_k\widehat{T}(z^k, \xi^k)\big\rangle \\ &\geq \big\langle z^k - x^*, h_k\widehat{T}(z^k, \xi^k)\big\rangle + \big\langle x^k - z^k, x^k - z^k - h_k\widehat{T}(x^k, \xi^k) + h_k\widehat{T}(z^k, \xi^k)\big\rangle. \end{aligned} \tag{3.10}$$

According to (1.1) and $z^k \in \Omega$, we have $\langle z^k - x^*, h_kF(x^*)\rangle \geq 0$. Under Assumption 3.1(ii), the operator F is monotone, thus

$$\big\langle z^k - x^*, h_kF(z^k)\big\rangle \geq 0.$$

Using the notation in (3.3), we obtain

$$\begin{aligned}&\big\langle z^k - x^*, h_k\widehat{T}(z^k, \xi^k)\big\rangle \\ &= \langle z^k - x^*, h_kF(z^k)\rangle + \langle z^k - x^*, h_k\epsilon_2^k\rangle \\ &\geq \langle z^k - x^*, h_k\epsilon_2^k\rangle. \end{aligned} \tag{3.11}$$

According to Cauchy-Schwarz inequality and the line search condition in step 5 of Algorithm 3.1, we have

$$\begin{aligned}&\big\langle x^k - z^k, x^k - z^k - h_k\widehat{T}(x^k, \xi^k) + h_k\widehat{T}(z^k, \xi^k)\big\rangle \\ &\geq \|x^k - z^k\|^2 - h_k\|x^k - z^k\|\|\widehat{T}(x^k, \xi^k) - \widehat{T}(z^k, \xi^k)\| \\ &\geq (1 - \nu)\|x^k - z^k\|^2. \end{aligned} \tag{3.12}$$

Combining (3.10)-(3.12), we obtain

$$\begin{aligned}&\big\langle x^k - x^*, x^k - z^k - h_k\widehat{T}(x^k, \xi^k) + h_k\widehat{T}(z^k, \xi^k)\big\rangle \\ &\geq \big\langle z^k - x^*, h_k\epsilon_2^k\big\rangle + (1 - \nu)\|x^k - z^k\|^2. \end{aligned} \tag{3.13}$$

Combining (3.9)-(3.11), we obtain (3.4). The inequality (3.5) can be obtained from the inequalities (3.9), (3.13) and $0 \leq \beta \leq 1$. ☐

We next show that the stochastic step size α_k has a deterministic lower bound.

Lemma 3.2. *Under Assumption* 3.1*(iii), the step size* α_k *in Algorithm* 3.1 *satisfies* $\alpha_k \geq \alpha_{\min}$*, where*

$$\alpha_{\min} = \frac{((1-\beta)(1-\overline{h}/4\sigma)+\beta(1-\nu))}{(2+2\beta^2\nu^2)}.$$

Proof. According to the definition of α_k, we have

$$\begin{aligned}\alpha_k &\geq \frac{((1-\beta)(1-h_k/(4\sigma))+\beta(1-\nu))\|x^k-z^k\|^2}{2\|x^k-z^k\|^2+2\|\beta h_k(\widehat{T}(x^k,\xi^k)-\widehat{T}(z^k,\xi^k))\|^2}\\ &\geq \frac{((1-\beta)(1-h_k/(4\sigma))+\beta(1-\nu))\|x^k-z^k\|^2}{(2+2\beta^2\nu^2)\|x^k-z^k\|^2}\\ &= \frac{((1-\beta)(1-h_k/(4\sigma))+\beta(1-\nu))}{(2+2\beta^2\nu^2)}\\ &\geq \frac{((1-\beta)(1-\overline{h}/(4\sigma))+\beta(1-\nu))}{(2+2\beta^2\nu^2)} := \alpha_{\min},\end{aligned}$$

where the first and second inequalities use the line search strategy in step 5 in Algorithm 3.1. □

In order to give the convergence analysis, an upper bound for $h_k\alpha_k$ also needs to be provided, which we will give in Lemma 3.3.

Lemma 3.3. *Under Assumption* 3.1*(iii), the step size in Algorithm* 3.1 *satisfies*

$$h_k\alpha_k \leq t := \begin{cases}\dfrac{\sigma}{1-\beta}, & 0\leq\beta<1,\\ 2\overline{h}, & \beta=1.\end{cases}$$

Proof. We declare this lemma in two cases.

(i) If $0\leq\beta<1$, according to the definition of α_k in step 8 in Algorithm 3.1, and after simplification, we have

$$\begin{aligned}\alpha_k &= \frac{((h_k(\beta-1))/4\sigma+1)\|x^k-z^k\|^2-\beta h_k\langle x^k-z^k,\widehat{T}(x^k,\xi^k)-\widehat{T}(z^k,\xi^k)\rangle}{\|d^k\|^2}\\ &= \frac{\sigma}{h_k(1-\beta)}+\frac{((h_k(\beta-1))/4\sigma+1-\sigma/(h_k(1-\beta)))\|x^k-z^k\|^2}{\|d^k\|^2}\\ &\quad-\frac{\|\sqrt{\sigma/(h_k(1-\beta))}\beta h_k(\widehat{T}(x^k,\xi^k)-\widehat{T}(z^k,\xi^k))\|^2}{\|d^k\|^2}\\ &\quad+\frac{(2\sigma/(h_k(1-\beta))-1)\beta h_k\langle x^k-z^k,\widehat{T}(x^k,\xi^k)-\widehat{T}(z^k,\xi^k)\rangle}{\|d^k\|^2}\\ &= \frac{\sigma}{h_k(1-\beta)}-\frac{\|t_1(x^k-z^k)-t_2(\widehat{T}(x^k,\xi^k)-\widehat{T}(z^k,\xi^k))\|^2}{\|d^k\|^2}\\ &\leq \frac{\sigma}{h_k(1-\beta)},\end{aligned}$$

where

$$t_1 = \frac{1}{2}\left(\frac{2\sigma}{h_k(1-\beta)} - 1\right)\sqrt{\frac{h_k(1-\beta)}{\sigma}}, \quad t_2 = \sqrt{\frac{\sigma}{h_k(1-\beta)}}\beta h_k.$$

Thus, $h_k\alpha_k \le \sigma/(1-\beta)$.

(ii) If $\beta = 1$, then

$$\alpha_k = \frac{\|x^k - z^k\|^2 - h_k\langle x^k - z^k, \widehat{T}(x^k,\xi^k) - \widehat{T}(z^k,\xi^k)\rangle}{\|d^k\|^2}. \tag{3.14}$$

According to step 5 in Algorithm 3.1, we have

$$\begin{aligned}
\|d^k\|^2 &= \left\|x^k - z^k - h_k\left(\widehat{T}(x^k,\xi^k) - \widehat{T}(z^k,\xi^k)\right)\right\|^2 \\
&= \|x^k - z^k\|^2 + h_k^2\left\|\widehat{T}(x^k,\xi^k) - \widehat{T}(z^k,\xi^k)\right\|^2 \\
&\quad - 2h_k\left\langle x^k - z^k, \widehat{T}(x^k,\xi^k) - \widehat{T}(z^k,\xi^k)\right\rangle \\
&\ge \|x^k - z^k\|^2 + h_k^2\left\|\widehat{T}(x^k,\xi^k) - \widehat{T}(z^k,\xi^k)\right\|^2 - 2\nu\|x^k - z^k\|^2 \\
&\ge (1-2\nu)\|x^k - z^k\|^2,
\end{aligned} \tag{3.15}$$

and

$$-h_k\left\langle x^k - z^k, \widehat{T}(x^k,\xi^k) - \widehat{T}(z^k,\xi^k)\right\rangle \le \nu\|x^k - z^k\|^2. \tag{3.16}$$

Combining (3.14)-(3.16) and $\nu \in (0,1)$, we have

$$\alpha_k \le \frac{(1+\nu)\|x^k - z^k\|^2}{(1-2\nu)\|x^k - z^k\|^2} \le 2.$$

Thus, $h_k\alpha_k \le 2\overline{h}$. Integrating these two cases, the desired conclusion is proved. □

In each iteration of Algorithm 3.1, the stochastic approximation of the expectation operator generates stochastic errors ϵ_1^k, ϵ_2^k, which can be controlled as the same way in [15, 19]. Here, we present the conclusions.

Lemma 3.4 ([15, Lemma 3.9,Theorem 3.11], [19, Lemma 8,9]). *Suppose that Assumptions 3.1(i),3.1(iii),3.1(iv) holds, then the sequence $\{x^k\}$ generated by Algorithm 3.1 satisfies*

$$\begin{aligned}
\big|\|\epsilon_1^k\|\big|_p &\le d_p\frac{|\|\epsilon(x^*,\xi)\||_p + L_p\|x^k - x^*\|}{\sqrt{N_k}}, \\
\big|\|\epsilon_2^k\|\big|_p &\le \frac{c_1|\|\epsilon(x^*,\xi)\||_{2p} + \overline{L}_{2p}\|x^k - x^*\|}{\sqrt{N_k}},
\end{aligned}$$

where $\overline{L}_{2p} = c_2L_2 + c_3L_p + c_4L_{2p}$; d_p, c_1, c_2, c_3, c_4 are positive constants (when $p = 2$, $d_p = 1$). The definitions of L_2, L_p, L_{2p} and $\epsilon(x^*,\xi)$ can be recalled from Lemma 3.3.

Next, we present the quasi-Ferjer monotonicity result of the sequence $\{x^k\}$ generated by Algorithm 3.1 with respect to the solution set $SOL(\Omega, F)$.

Proposition 3.1. *Under Assumptions* 3.1*(i)*-3.1*(iv), the sequence* $\{x^k\}$ *generated by Algorithm* 3.1 *satisfies*

$$\begin{aligned}&\mathbb{E}\left[\|x^{k+1}-x^*\|^2|\mathcal{F}^k\right]\\ &\le\left(1+\frac{D_1}{\sqrt{N_k}}\right)\|x^k-x^*\|^2-\frac{\kappa_2 h_{\min}^2}{4\overline{h}^2}e^2(x^k,\overline{h})+\frac{D_2}{\sqrt{N_k}}\|x^k-x^*\|+\frac{D_3}{\sqrt{N_k}},\end{aligned}$$

where

$$\begin{aligned}D_1&=2L_2^2\left(\frac{4\eta^2t^2}{\kappa_2}+\frac{\kappa_2\overline{h}^2}{2}\right)+\frac{8\eta^2t^2\overline{L}_4^2}{\kappa_2}+2\eta t(d_1L_1+\overline{L}_2),\\ D_2&=2\eta t\big(d_1|||\epsilon(x^*,\xi)|||_1+c_1|||\epsilon(x^*,\xi)|||_2\big),\\ D_3&=2\left(\frac{4\eta^2t^2}{\kappa_2}+\frac{\kappa_2\overline{h}^2}{2}\right)|||\epsilon(x^*,\xi)|||_2^2+\frac{8\eta^2t^2c_1^2|||\epsilon(x^*,\xi)|||_4^2}{\kappa_2}.\end{aligned}$$

Proof. From the iteration format (3.2), we have

$$\|x^{k+1}-x^*\|^2=\|x^k-x^*\|^2-2\eta\alpha_k\langle x^k-x^*,d^k\rangle+\eta^2\alpha_k^2\|d^k\|^2.$$

Combining (3.4) with the above inequality, we obtain

$$\begin{aligned}\|x^{k+1}-x^*\|^2&\le\|x^k-x^*\|^2+\eta^2\alpha_k^2\|d^k\|^2-2\eta\alpha_k\varphi_k\\ &\quad-2\eta\alpha_k\left(h_k(1-\beta)\langle z^k-x^*,\epsilon_1^k\rangle+\beta h_k\langle z^k-x^*,\epsilon_2^k\rangle\right)\\ &=\|x^k-x^*\|^2-\eta(2-\eta)\alpha_k\varphi_k\\ &\quad-2\eta\alpha_k\left(h_k(1-\beta)\langle z^k-x^*,\epsilon_1^k\rangle+\beta h_k\langle z^k-x^*,\epsilon_2^k\rangle\right)\\ &\le\|x^k-x^*\|^2-\kappa_2\|x^k-z^k\|^2\\ &\quad-2\eta\alpha_k\left(h_k(1-\beta)\langle z^k-x^*,\epsilon_1^k\rangle+\beta h_k\langle z^k-x^*,\epsilon_2^k\rangle\right),\end{aligned}\tag{3.17}$$

where

$$\kappa_2=\eta(2-\eta)\alpha_{\min}\left((1-\beta)\left(1-\frac{\overline{h}}{4\sigma}\right)+\beta(1-\nu)\right)>0,$$

the equality is according to the setting of α_k in step 8 in Algorithm 3.1, and the last inequality is due to (3.5), $\alpha_k\ge\alpha_{\min}$ and $h_k\le\overline{h}$. According to the Young inequality, we have

$$\begin{aligned}-2\langle z^k-x^*,\epsilon_1^k\rangle&=-2\langle z^k-x^k,\epsilon_1^k\rangle-2\langle x^k-x^*,\epsilon_1^k\rangle\\ &\le\sigma_1^2\|z^k-x^k\|^2+\frac{1}{\sigma_1^2}\|\epsilon_1^k\|^2+2\|x^k-x^*\|\|\epsilon_1^k\|,\end{aligned}\tag{3.18}$$

$$
\begin{aligned}
-2\langle z^k - x^*, \epsilon_2^k\rangle &= -2\langle z^k - x^k, \epsilon_2^k\rangle - 2\langle x^k - x^*, \epsilon_2^k\rangle \\
&\le \sigma_2^2\|z^k - x^k\|^2 + \frac{1}{\sigma_2^2}\|\epsilon_2^k\|^2 + 2\|x^k - x^*\|\|\epsilon_2^k\|, \qquad (3.19)
\end{aligned}
$$

where

$$
\sigma_1^2 = \frac{\kappa_2}{4\eta t(1-\beta)}, \quad \sigma_2^2 = \frac{\kappa_2}{4\eta t\beta}
$$

are two constants. Combining (3.17)-(3.19) and Lemma 3.3, we obtain

$$
\begin{aligned}
\|x^{k+1} - x^*\|^2 \le \|x^k - x^*\|^2 - \frac{\kappa_2}{2}\|x^k - z^k\|^2 + \frac{4\eta^2 t^2}{\kappa_2}\left(\|\epsilon_1^k\|^2 + \|\epsilon_2^k\|^2\right) \\
+ 2\eta t\left(\|x^k - x^*\|\|\epsilon_1^k\| + \|x^k - x^*\|\|\epsilon_2^k\|\right). \qquad (3.20)
\end{aligned}
$$

We then deal with the term $\|x^k - z^k\|^2$ in (3.20). Using Lemma 2.2 with $h_k \le \overline{h}$, we have

$$
\begin{aligned}
\frac{h_k^2 e^2(x^k, \overline{h})}{\overline{h}^2} &\le e^2(x^k, h_k) \\
&= \left\|x^k - \Pi(x^k - h_k F(x^k))\right\|^2 \\
&\le 2\|x^k - z^k\|^2 + 2\left\|\Pi(x^k - h_k\widehat{T}(x^k, \xi^k)) - \Pi(x^k - h_k F(x^k))\right\|^2 \\
&= 2\|x^k - z^k\|^2 + 2\left\|\Pi(x^k - h_k(F(x^k) + \epsilon_1^k)) - \Pi(x^k - h_k F(x^k))\right\|^2 \\
&\le 2\|x^k - z^k\|^2 + 2h_k^2\|\epsilon_1^k\|^2, \qquad (3.21)
\end{aligned}
$$

where the last inequality dues to Lemma 2.1(i). From (3.20) and (3.21), we obtain

$$
\begin{aligned}
\|x^{k+1} - x^*\|^2 \le \|x^k - x^*\|^2 - \frac{\kappa_2 h_k^2}{4\overline{h}^2} e^2(x^k, \overline{h}) + \left(\frac{4\eta^2 t^2}{\kappa_2} + \frac{\kappa_2\overline{h}^2}{2}\right)\|\epsilon_1^k\|^2 \\
+ \frac{4\eta^2 t^2}{\kappa_2}\|\epsilon_2^k\|^2 + 2\eta t\left(\|x^k - x^*\|\|\epsilon_1^k\| + \|x^k - x^*\|\|\epsilon_2^k\|\right). \qquad (3.22)
\end{aligned}
$$

Combining (3.22), Remark 3.5(ii) and Lemma 3.4, we obtain

$$
\begin{aligned}
&\mathbb{E}\left[\|x^{k+1} - x^*\|^2 | \mathcal{F}^k\right] \\
&\le \|x^k - x^*\|^2 - \frac{\kappa_2 h_{\min}^2}{4\overline{h}^2} e^2(x^k, \overline{h}) \\
&\quad + \left(\frac{2L_2^2(4\eta^2 t^2/\kappa_2 + \kappa_2\overline{h}^2/2)}{N_k} + \frac{8\eta^2 t^2 \overline{L}_4^2}{\kappa_2 N_k} + \frac{2\eta t(d_1 L_1 + \overline{L}_2)}{\sqrt{N_k}}\right)\|x^k - x^*\|^2 \\
&\quad + \frac{2\eta t(d_1|||\epsilon(x^*, \xi)|||_1 + c_1|||\epsilon(x^*, \xi)|||_2)}{\sqrt{N_k}}\|x^k - x^*\| \\
&\quad + \frac{2(4\eta^2 t^2/\kappa_2 + \kappa_2\overline{h}^2/2)|||\epsilon(x^*, \xi)|||_2^2}{N_k} + \frac{8\eta^2 t^2 c_1^2|||\epsilon(x^*, \xi)|||_4^2}{\kappa_2 N_k}. \qquad (3.23)
\end{aligned}
$$

According to the definitions of D_1, D_2, D_3 and the fact $1/N_k \leq 1/\sqrt{N_k}$, the desired conclusion is proved. $\square$

In Algorithm 3.1, the correction step does not involve resampling. Consequently, the stochastic operator in this step loses its unbiasedness as an estimator of the expectation operator, i.e.,

$$\mathbb{E}\big[\epsilon_2^k|\mathcal{F}^k\big] = \mathbb{E}\big[\widehat{T}(z^k,\xi^k) - F(z^k)|\mathcal{F}^k\big] \neq 0.$$

Moreover, since the random variables α_k, h_k, and z_k are not in $\mathcal{F}^k$ and are interdependent, even though

$$\mathbb{E}\big[\epsilon_1^k|\mathcal{F}^k\big] = \mathbb{E}\big[\widehat{T}(x^k,\xi^k) - F(x^k)|\mathcal{F}^k\big] = 0,$$

the conditional expectation $\mathbb{E}[\alpha_k h_k\langle z^k - x^*, \epsilon_1^k\rangle|\mathcal{F}^k]$ is generally non-zero. Hence, in Proposition 3.1, it becomes necessary to control both $\langle z^k - x^*, \epsilon_1^k\rangle$ and $\langle z^k - x^*, \epsilon_2^k\rangle$. Applying Young's inequality introduces an additional term $(D_2/\sqrt{N_k})\|x^k - x^*\|$, which obstructs a direct application of the supermartingale convergence theorem. Accordingly, we develop a case-by-case analysis to complete the proof.

Theorem 3.1. *Suppose that Assumption* 3.1 *holds, then the sequence* $\{x^k\}$ *generated by Algorithm* 3.1 *converges to a solution point* $x^* \in SOL(\Omega, F)$ *a.s.*

Proof. Under Assumption 3.1(i)-3.1(iv), Proposition 3.1 holds. We consider two cases.

(1) If $\|x^k - x^*\| \leq 1$. According to Proposition 3.1, we get

$$\mathbb{E}\big[\|x^{k+1} - x^*\|^2|\mathcal{F}^k\big] \leq \left(1 + \frac{D_1}{\sqrt{N_k}}\right)\|x^k - x^*\|^2 - \frac{\kappa_2 h_{\min}^2}{4\overline{h}^2}e^2(x^k,\overline{h}) + \frac{D_2 + D_3}{\sqrt{N_k}}.$$

(2) If $\|x^k - x^*\| > 1$. According to Proposition 3.1, we obtain

$$\mathbb{E}\big[\|x^{k+1} - x^*\|^2|\mathcal{F}^k\big] \leq \left(1 + \frac{D_1 + D_2}{\sqrt{N_k}}\right)\|x^k - x^*\|^2 - \frac{\kappa_2 h_{\min}^2}{4\overline{h}^2}e^2(x^k,\overline{h}) + \frac{D_3}{\sqrt{N_k}}.$$

Thus, for all k, we have

$$\mathbb{E}\big[\|x^{k+1} - x^*\|^2|\mathcal{F}^k\big] \leq \left(1 + \frac{D_1 + D_2}{\sqrt{N_k}}\right)\|x^k - x^*\|^2 - \frac{\kappa_2 h_{\min}^2}{4\overline{h}^2}e^2(x^k,\overline{h}) + \frac{D_2 + D_3}{\sqrt{N_k}}.$$

Setting

$$y^k = \|x^k - x^*\|^2, \quad a^k = \frac{D_1 + D_2}{\sqrt{N_k}}, \quad b^k = \frac{D_2 + D_3}{\sqrt{N_k}}, \quad u^k = \frac{\kappa_2 h_{\min}^2}{4\overline{h}^2}e^2(x^k,\overline{h})$$

in Lemma 2.3, combining with Assumption 3.1(v), we obtain a.s. $\{\|x^k - x^*\|^2\}$ converges and $\sum_{k=0}^{\infty} e^2(x^k,\overline{h}) < \infty$. Thus, a.s. $\{x^k\}$ is bounded and $e(x^k,\overline{h}) \to 0$. Letting the subsequence $x^{k_j} \to x^\infty$, by the continuity of $e^2(\cdot,\overline{h})$, we have $e(x_j^k,\overline{h}) \to e(x^\infty,\overline{h})$. Since $e(x^k,\overline{h}) \to 0$, we have $e(x^\infty,\overline{h}) = 0$. According to Lemma 2.1(ii), $x^{k_j} \to x^\infty \in SOL(\Omega, F)$. Letting $x^* = x^\infty$, we have $\{\|x^k - x^\infty\|^2\}$ converges a.s. On the other hand, $\|x^{k_j} - x^\infty\|^2 \to \|x^\infty - x^\infty\|^2 = 0$. Thus, $\|x^k - x^\infty\| \to 0$ a.s. i.e., $x^k \to x^\infty$ a.s. $\square$

4. Numerical experiment

In this section, we give two numerical experiments to illustrate the superiority of proposed algorithm. The experiments are operated with Matlab R2024b on a Windows 11 with a 3.30 Ghz processor and 32 GB of memory.

Experiment 1: Regularized two-player zero-sum matrix game—special equation constraints

This experiment relates to a regularized two-player zero-sum matrix game with an uncertain payoff matrix [13]. In particular, the payoff matrix A_ξ is randomly distributed and can only be sampled for each (mixed) strategy. The problem can be formulated as follows:

$$\begin{aligned} \min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} \quad & f(x,y)=\mathbb{E}\left[\frac{\lambda}{2}\|x\|^2+x^TA_\xi y-\frac{\lambda}{2}\|y\|^2\right] \\ \text{s.t.} \quad & \sum_{i=1}^n x_i=1, \quad \sum_{i=1}^m y_i=1, \quad x,y\geq 0. \end{aligned} \tag{4.1}$$

According to [2, Proposition 2.6.1], problem (4.1) is equivalent to the SVI problem, where

$$F(u)=\begin{pmatrix}\lambda x+A_0y\\ -A_0^Tx+\lambda y\end{pmatrix}=\mathbb{E}_\xi[T(u,\xi)], \quad T(u,\xi)=\begin{pmatrix}\lambda x+A_\xi y\\ -A_\xi^Tx+\lambda y\end{pmatrix},$$

and the constraint set is $\Omega=\{u\geq 0: Bu=c\}$ with

$$u=\begin{pmatrix}x\\ y\end{pmatrix}, \quad B=\begin{pmatrix}e_{n\times 1}^T & 0\\ 0 & e_{m\times 1}^T\end{pmatrix}, \quad c=\begin{pmatrix}1\\ 1\end{pmatrix}.$$

The vectors $e_{n\times1}, e_{m\times1}$ are column vectors, each with components equal to 1.

We generate the random matrix A_ξ by sampling its elements as independent and identically distributed Gaussian variables, where $A_\xi\sim\mathcal{N}(A_0,\varrho^2 I_{(n+m)})$ with A_0 being randomly generated. Each element $a_{ij}\in A_0$ follows a distribution $a_{ij}\sim\mathcal{N}(0,1)$. The parameters are set as $\varrho^2=1$ and $\lambda=0.01$. We consider two dimensionality settings: (i) $n=10, m=20$, the same setting as [13]; and (ii) the higher-dimensional case $n=100, m=200$.

The parameter settings in our Algorithm 3.1 are as follows: $\mu=0.4$, $\nu=0.9$, $\sigma=\lambda/(\lambda^2+\|A_0\|^2)$, $\underline{h}=10^{-6}$, $\overline{h}=3.9\sigma$, $\gamma_0^0=\overline{h}$, $\theta=0.9$, $\tau=1.5$, $\eta=1.9$, $N_k=\lceil(k+1)^{2.1}/(m+n)\rceil$. All methods start at the same initial point x_0 whose entries are randomly generated in the interval $(0,1)$. The residual is set as $e(x^k,1)$. The stopping criterion is set as $e(x^k,1)<10^{-1}$ or maximum number of iteration which equals to 2000.

First, we test the impact of different β on Algorithm 3.1. Due to the randomness of the algorithm, we conduct 5 trials for each value of β and take the average to ensure robustness. The results under different dimensions are shown in Tables 1 and 2. Fig. 1 presents the intuitive outcome from one of the random trials (We have taken the logarithm of the values on the vertical axis). From Tables 1 and 2, we can see that the best numerical performance is achieved when β is set to 0.5. This aligns with our interpretation from the perspective of discrete ODE in Section 2.2. Therefore, in the following numerical experiments, we set the offset weight $\beta = 0.5$ in our Algorithm 3.1.

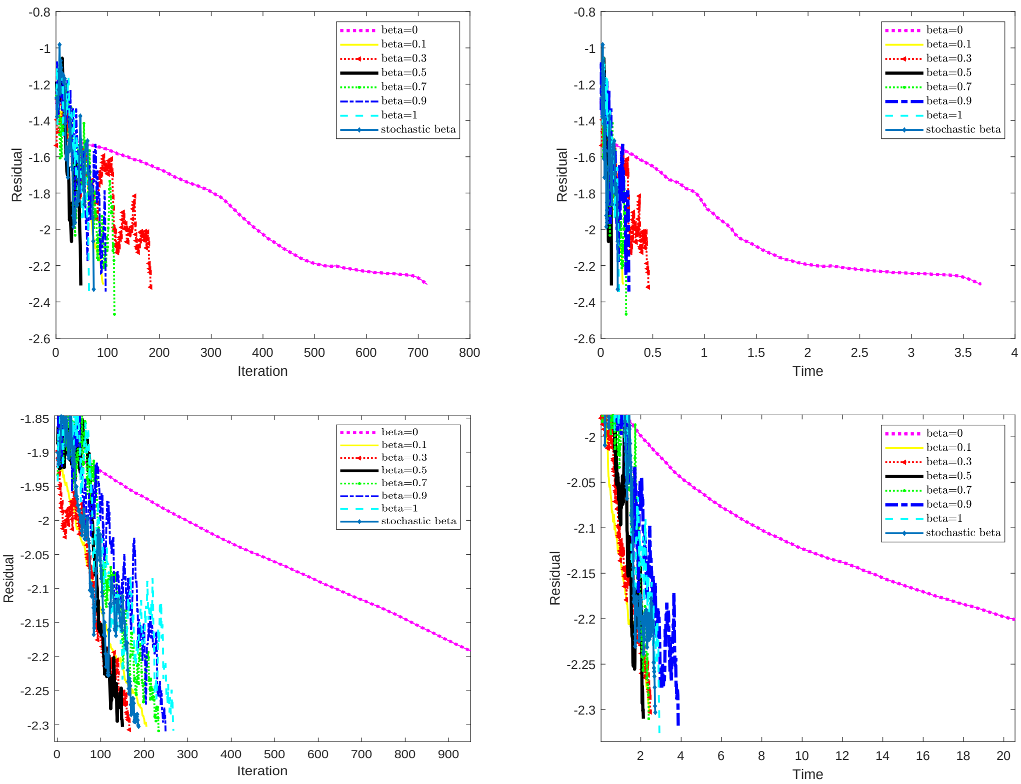

Figure 1: Comparison of Algorithm 3.1 under different β in Experiment 1: Results for dimensions $(n, m) = (10, 20)$ (top row) and $(100, 200)$ (bottom row).

Table 1: Comparison of Algorithm 3.1 under different β for Experiment 1 ($n = 10, m = 20$).

β	0	0.1	0.3	0.5	0.7	0.9	1	stochastic
Iter.	717	148.8	88.8	**64.4**	111	122.4	123.8	83.2
CPU	5.8757	0.4365	0.2179	**0.1462**	0.2561	0.3052	0.2955	0.1938

Table 2: Comparison of Algorithm 3.1 under different β for Experiment 1 ($n = 100, m = 200$).

β	0	0.1	0.3	0.5	0.7	0.9	1	stochastic
Iter.	1290.2	205.5	166.8	**150.2**	233	249	266.8	187
CPU	48.3844	3.6871	3.1570	**2.2271**	4.1163	3.4018	5.4923	2.7177

Next, we compare the numerical performance of algorithm SEP proposed in [13], algorithm I-L-SEG proposed in [19], algorithm S-RPG (when β is set to 0). In this algorithm, each iteration step requires only one gradient estimation $\widehat{T}(x^k, \xi^k)$ to be computed. Therefore, we have additionally written separate code for this algorithm, algorithm S-PC (when β is set to 1), and our algorithm S-IPC. The parameter settings of algorithms SEP and I-L-SEG are consistent with those in their respective original papers. We also conduct 5 trials for each algorithm and take the average to ensure robustness. The results under different dimensions are shown in Tables 3 and 4. Fig. 2 presents the intuitive outcome from one of the random trials.

From Tables 3-4 and Fig. 2, it can be seen that the proposed algorithm S-IPC demonstrates superior performance in both iteration steps and CPU time. In terms of iteration steps, it achieves a 20%-30% improvement than algorithm I-L-SEG. In terms of CPU time, it outperforms algorithm S-PC by 50%-60% and algorithm I-L-SEG by 60%-70%. The primary reason for S-IPC's greater time efficiency over I-L-SEG lies in its single-sample and one-projection design in per iteration, which significantly reduces computational CPU time in stochastic iterations.

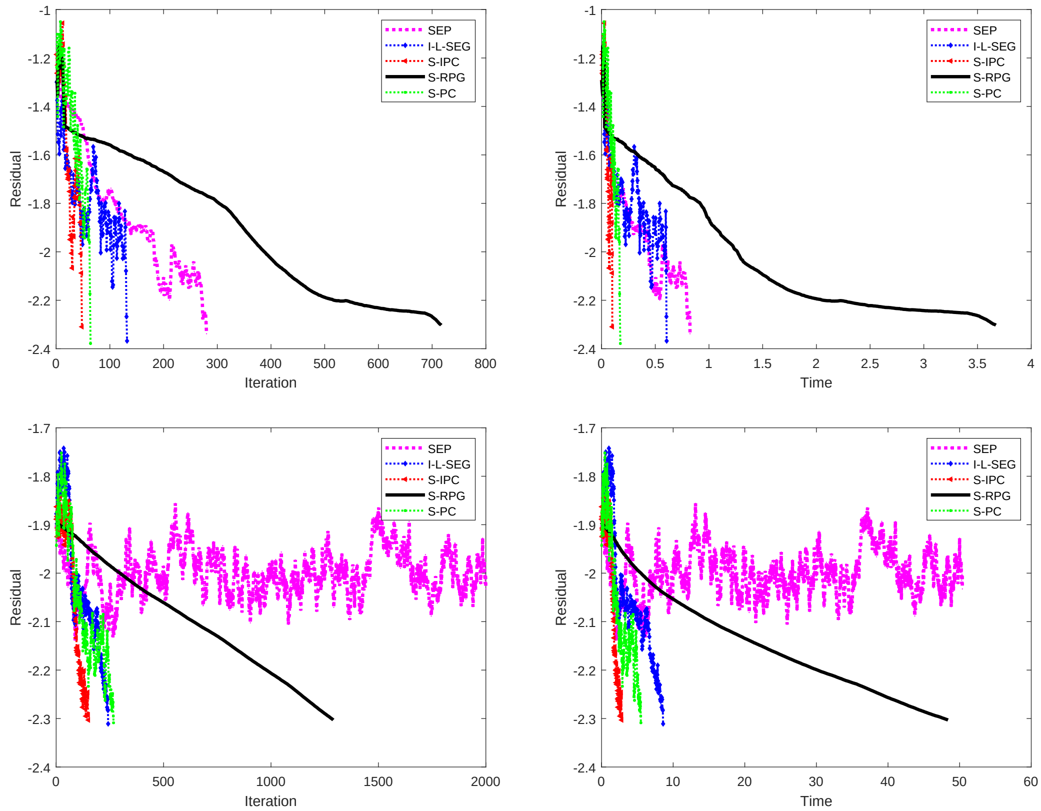

Figure 2: Comparison of five algorithms' efficiency in Experiment 1: results for dimensions $(n, m) = (10, 20)$ (top row) and $(100, 200)$ (bottom row).

Table 3: Comparison of five algorithms' efficiency for Experiment 1 ($n = 10, m = 20$).

Algorithm	SEP	I-L-SEG	S-IPC	S-RPG	S-PC
Iter.	111.6	81.2	**64.4**	717	123.8
CPU	0.4146	0.4011	**0.1462**	5.8757	0.2955

Table 4: Comparison of five algorithms' efficiency for Experiment 1 ($n = 100, m = 200$).

Algorithm	SEP	I-L-SEG	S-IPC	S-RPG	S-PC
Iter.	2000	242	**150.2**	1290.2	266.8
CPU	50.3848	8.5967	**2.2271**	48.3844	5.4923

Experiment 2: Regularized two-player zero-sum matrix game—general polyhedral constraints

Experiment 2 is designed to test on a strongly convex strongly concave stochastic saddle-point problem with more general polyhedral constraints

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^m} \quad & f(x,y) = \mathbb{E}\left[\frac{\lambda}{2}\|x\|^2 + x^T A_\xi y - \frac{\lambda}{2}\|y\|^2\right] \\ \text{s.t.} \quad & A_1 x \le b_1, \quad A_2 y \le b_2. \end{aligned}$$

The stochastic matrix A_ξ is sampled in the same way as in Experiment 1. The constraint coefficients $A_1 \in \mathbb{R}^{(2n+2)\times n}$, $A_2 \in \mathbb{R}^{(2m+2)\times m}$, and each element of A_1 and A_2 is randomly generated from a uniform distribution over the interval $(-5, 5)$. $b_1 \in \mathbb{R}^{2n+2}$, $b_2 \in \mathbb{R}^{2n+2}$ are generated randomly such that the initial all-ones vector x^0, y^0 is strictly feasible.

When the constraint set is a general polyhedron, the cost of projection increases substantially; even a modest rise in problem dimension leads to significantly longer computation times. To illustrate this effect, we compare two small-scale cases: (i) $n = 10$, $m = 20$ (as in [13]), and (ii) $n = 15, m = 30$. The maximum iteration counts are set to 2000 and 5000, respectively. Other problem and algorithm parameters are set the same as Experiment 1.

Tables 5 and 6 respectively presents the average iteration counts and CPU times of Algorithm 3.1 across 5 randomized trials with different β values under different dimensions. Fig. 3 visually demonstrates the results from one typical trial. From these tables and figure, we can see that the best numerical performance is achieved when β is near 0.5. Therefore, for unity, in the following numerical experiments, we set the offset weight $\beta = 0.5$ in our Algorithm 3.1.

Tables 7 and 8 present the average iteration counts and CPU times of Algorithm 3.1 with four other different algorithms over 5 randomized trials for Experiment 2. Fig. 4 visually illustrates the results from one representative trial. As shown in Tables 7 and 8, our proposed S-IPC algorithm demonstrates superior performance in both iteration counts and CPU time. Although our S-IPC algorithm achieves only about 6%

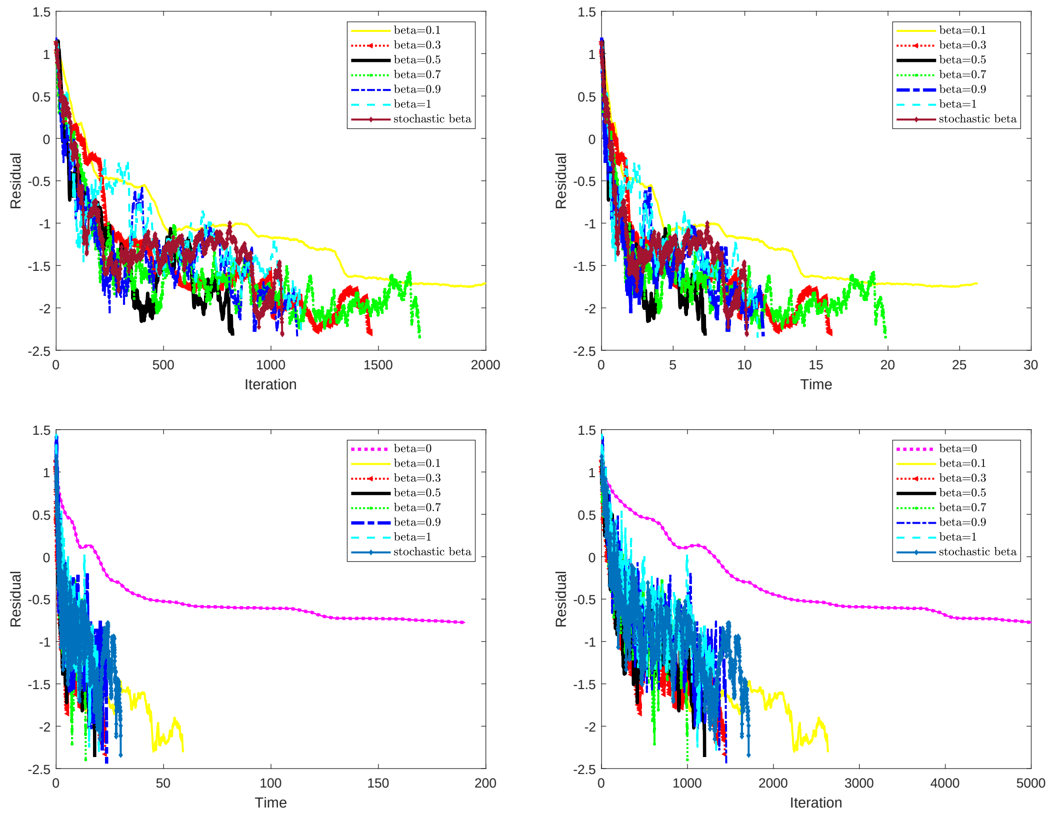

Figure 3: Comparison of Algorithm 3.1 under different β in Experiment 2: Results for dimensions $(n, m) = (10, 20)$ (top row) and $(100, 200)$ (bottom row).

Table 5: Comparison of Algorithm 3.1 under different β for Experiment 2 ($n = 10, m = 20$).

β	0	0.1	0.3	0.5	0.7	0.9	1	stochastic
Iter.	2000	1279.4	1145.6	**1121.4**	1552.2	1547.4	1768.6	1217.8
CPU	26.6552	14.5877	12.1150	**12.0838**	23.6031	21.6251	22.4715	13.1776

Table 6: Comparison of Algorithm 3.1 under different β for Experiment 2 ($n = 15, m = 30$).

β	0	0.1	0.3	0.5	0.7	0.9	1	stochastic
Iter.	5000	2635.2	1433.8	1199.5	**999.2**	1451	1334	1712.8
CPU	190.1154	59.0670	22.8295	17.9238	**13.7526**	23.4922	19.8950	30.0595

improvement in iteration counts compared to the I-L-SEG algorithm, it demonstrates a significantly greater 45%-50% reduction in CPU time. What is more, comparing the numerical results across different dimensions, it can be seen that our algorithm is significantly less sensitive to changes in dimension.

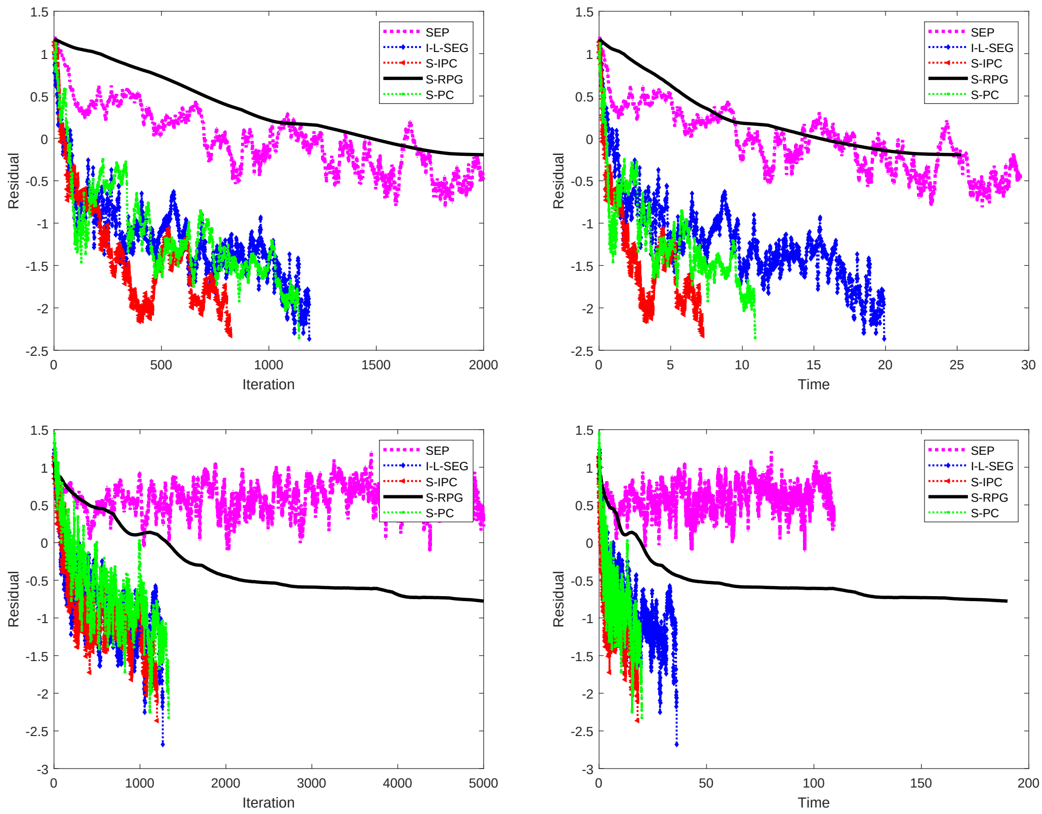

Figure 4: Comparison of five algorithms' efficiency in Experiment 2: Results for dimensions $(n, m) = (10, 20)$ (top row) and $(100, 200)$ (bottom row).

Table 7: Comparison of five algorithms' efficiency for Experiment 2 $(n = 10, m = 20)$.

Algorithm	SEP	I-L-SEG	S-IPC	S-RPG	S-PC
Iter.	2000	1197.4	**1121.4**	2000	1768.6
CPU	32.6663	22.0812	**12.0838**	26.6552	22.4715

Table 8: Comparison of five algorithms' efficiency for Experiment 2 $(n = 15, m = 30)$.

Algorithm	SEP	I-L-SEG	S-IPC	S-RPG	S-PC
Iter.	5000	1266.2	**1199.5**	5000	1334
CPU	109.3994	36.1289	**17.9238**	190.1154	19.8950

5. Conclusion

This paper presents an improved stochastic projection contraction algorithm for solving SVI problems. The proposed method features three key innovations:

(i) Elimination of resampling in the correction step, substantially improving computational efficiency.

(ii) Introduction of a offset weight β to optimize the search direction, with differential-equation-based interpretations provided for special cases.

(iii) Implementation of adaptive step sizes in both prediction and correction steps.

Numerical experiments demonstrate the algorithm's effectiveness, with results for optimal β values corroborating our theoretical analysis from the differential equation perspective. Future research directions include developing adaptive strategies for dynamic weight selection (β_k) to automatically determine optimal search directions for varying problem structures and sample characteristics.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant Nos. 12131004, 12471290, 12471282), by the National Key Research and Development Program of China (Grant No. 2024YFA1012901), and by the Scientific and Technological Innovation Project of China Academy of Chinese Medical Sciences (Grant No. CI2023C064YLL).

References

[1] H. Attouch and L. Csaba, *Newton-like inertial dynamics and proximal algorithms governed by maximally monotone operators*, SIAM J. Optim. 30(4) (2020), 3252–3283.

[2] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, (2003).

[3] R. I. Bot, P. Mertikopoulos, M. Staudigl, and P. T. Vuong, *Forward-backward-forward methods with variance reduction for stochastic variational inequalities*, arXiv:1902.03355v1, (2019).

[4] L. Bottou, F. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev. 60(2) (2018), 223–311.

[5] X. Cai, D. Han, and L. Xu, *An improved first-order primal-dual algorithm with a new correction step*, J. Global Optim. 57(4) (2013), 1419–1428.

[6] L. Chen and H. Luo, *A unified convergence analysis of first order convex optimization methods via strong Lyapunov functions*, arXiv:2108.00132v1, (2021).

[7] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, *Training GANs with Optimism*, ICLR 2018 Conference, (2018).

[8] J. Diakonikolas and L. Orecchia, *The approximate duality gap technique: A unified theory of first-order methods*, SIAM J. Optim. 29(1) (2019), 660–689.

[9] F. Facchinei and J. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems. Vol. I*, Springer, (2003).

[10] B. He and L. Liao, *Improvements of some projection methods for monotone nonlinear variational inequalities*, J. Optim. Theory Appl. 112(1) (2002), 111–128.

[11] B. He and M. Xu, *A general framework of contraction methods for monotone nonlinear inequalities*, Pac. J. Optim. 4(2) (2008), 195–212.

[12] B. He, X. Yuan, and J. Zhang, *Comparison of two kinds of prediction-correction methods for monotone variational inequalities*, Comput. Optim. Appl. 27(3) (2004), 247–267.

[13] K. Huang and S. Zhang, *New first-order algorithms for stochastic variational inequalities*, SIAM J. Optim. 32(4) (2022), 2745–2772.

[14] A. Iusem, A. Jofré, R. Oliveira, and P. Thompson, *Extragradient method with variance reduction for stochastic variational inequalities*, SIAM J. Optim. 27(2) (2017), 686–724.

[15] A. Iusem, A. Jofré, R. Oliveira, and P. Thompson, *Variance-based extragradient methods with line search for stochastic variational inequalities*, SIAM J. Optim. 29(1) (2019), 175–206.

[16] Z. Jia and X. Cai, *A relaxation of the parameter in the forward-backward splitting method*, Pac. J. Optim. 13 (2017), 665–681.

[17] H. Jiang and H. Xu, *Stochastic approximation approaches to the stochastic variational inequality problem*, IEEE Trans. Automat. Control 53(6) (2008), 1462–1475.

[18] J. Koshal, A. Nedic, and U. V. Shanbhag, *Regularized iterative stochastic approximation methods for stochastic variational inequality problems*, IEEE Trans. Autom. Control 58(3) (2012), 594–609.

[19] T. Li, X. Cai, Y. Song, and Y. Ma, *Improved variance reduction extragradient method with line search for stochastic variational inequalities*, J. Global Optim. 87 (2023), 423–446.

[20] T. Li, D. Han, T. Wang, and X. Cai, *Research on the descent direction of prediction correction algorithms for pseudo-convex/convex optimization problems*, arXiv:2512.04575, (2025).

[21] C. Lin, *Numerical Computation Methods*, (in Chinese), Science Press, (2005).

[22] Y. Malitsky and M. Tam, *A forward-backward splitting method for monotone inclusions without cocoercivity*, SIAM J. Optim. 30(2) (2020), 1451–1472.

[23] K. Mishchenko, D. Kovalev, E. Shulgin, P. Richtárik, and Y. Malitsky, *Revisiting stochastic extragradient*, in: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, S. Chiappa and R. Calandra (Eds), PMLR 108 (2020), 4573–4582.

[24] A. Nemirovski, *Prox-method with rate of convergence O(1/t) for variational inequality with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, SIAM J. Optim. 15(1) (2005), 229–251.

[25] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim. 19(4) (2009), 1574–1609.

[26] N. Parikh, S. Boyd, *Proximal algorithms*, Found. Trends Optim. 1(3) (2013), 123–231.

[27] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statist. 22(3) (1951), 400–407.

[28] A. Shapiro, D. Dentcheva, and A. Ruszczyňski, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, (2009).

[29] W. Su, S. Boyd, and E. Candès, *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*, J. Mach. Learn. Res. 17(153) (2016), 1–43.

[30] P. Tseng, *A modified forward-backward splitting method for maximal monotone mappings*, SIAM J. Control Optim. 38(2) (2000), 431–446.

[31] H. Xu, *Sample average approximation methods for a class of stochastic variational inequality problems*, Asia-Pac. J. Oper. Res. 27(1) (2010), 103–119.

[32] G. Zhang et al., *A unified analysis of first-order methods for smooth games via integral quadratic constraints*, arXiv:2009.11359v4, (2021).